Hochschule Neubrandenburg
University of Applied Sciences

Fachbereich Landschaftswissenschaften und Geomatik

Master-Studiengang Geodäsie und Geoinformatik

Masterarbeit

Entwicklung eines MEMS-basierten Low-Cost-Sensors für geodätische Überwachungsmessungen

Zur Erlangung des akademischen Grades

Master of Engineering

Vorgelegt von: ***B.Eng. Uwe Köster***

Erstprüfer: **Prof. Dr. –Ing. Karl Foppe**

Zweitprüfer: **Prof. Dr. rer. nat. habil. Gerd Teschke**

URN: **urn:nbn:de:gbv:519-thesis2018-0012-7**

Neubrandenburg, den 17. September 2018

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die hier vorliegende Masterarbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäß entlehnte Stellen als solche kenntlich gemacht habe. Die Arbeit hat in gleicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

_____

Ort, Datum Unterschrift

Kurzfassung

Die Masterarbeit befasst sich mit der Entwicklung, der Konstruktion, der Kalibrierung und der Anwendung eines MEMS-basierten Sensors für geodätische Überwachungsmessungen. Ebenfalls soll ein in der Ingenieurgeodäsie neuer Auswertealgorithmus vorgestellt werden. Geodätische Überwachungsmessungen werden in Zukunft an Bedeutung gewinnen. Hierzu sind insbesondere preiswerte Sensoren und neue Auswertealgorithmen wichtig. In dieser Masterarbeit sollen die Grundsteine für neue Anwendungsgebiete gelegt werden, da hier neben einem neuentwickelten Sensor auch eine neue Auswertestrategie zur Anwendung kommt. Geeignete MEMS-basierte Sensoren werden ausgewählt. Ein Sensorblock wird konstruiert, kalibriert und getestet. Eine Validierung des Sensors wird dann an einem Referenzobjekt vorgenommen. Danach werden ausgewählte Bauwerke im Rahmen eines Structural-Health-Monitoring überwacht und die Messdaten ausgewertet. Die Auswertung erfolgt im Kontext der Zeitreihenanalyse. Hier wird vor allem auf die Short-Time-Fourier-Transformation eingegangen, welche im Bereich des geodätischen Monitorings bisher noch wenig Anwendung findet. Es soll insbesondere auf die großen Vorteile der Auswertung von Sensordaten mittels Short-Time-Fourier-Transformation im Gegensatz zu herkömmlichen Auswertealgorithmen eingegangen werden.

Abstract

This thesis deals with the development, the design, the calibration and the application of a MEMS-based sensor for geodetic surveillance measurements. Also, a new evaluation algorithm in engineering geodesy will be presented. Geodetic monitoring measurements will become more important in the future. In particular, inexpensive sensors and new evaluation algorithms are important for this purpose. Within this master thesis, the foundations for new fields of application should be laid, as a new evaluation strategy is used in addition to a newly developed sensor. Suitable MEMS-based sensors are selected. A sensor block is constructed, and this sensor block is calibrated and tested. The sensor is then validated on a reference object. Thereafter, selected structures are monitored as part of a structural health monitoring and the measured data are evaluated. The evaluation takes place in the context of the time series analysis. Here, the focus is on the short-time Fourier analysis, which is still

rarely used in geodetic monitoring. In particular, the great advantages of the evaluation of sensor data by means of short-time Fourier analysis in contrast to conventional evaluation algorithms will be discussed.

Schlagwörter:

MEMS, Beschleunigungssensor, Monitoring, Zeitreihenanalyse, Structural-Health-Monitoring, Raspberry Pi, Arduino, Mikrocontroller, Einplatinencomputer

Danksagung

Ich möchte mich bei Prof. Dr. –Ing. Karl Foppe und Prof. Dr. rer. nat. habil. Gerd Teschke für die fachliche Unterstützung bedanken. Herr Prof. Dr. rer. nat. habil. Teschke präsentierte in seiner Vorlesung „Digitale Bildverarbeitung“ die notwendigen Algorithmen, welche bisher noch kaum im geodätischen Bereich angewendet werden. Herr Prof. Dr. –Ing. Foppe ließ sich von meiner Idee eines selbst entwickelten niedrigpreisigen Sensors für Überwachungsmessungen überzeugen und zeigte großen Enthusiasmus an dieser Arbeit.

Weiterhin bedanke ich mich bei Prof. Dr. –Ing. Wilhelm Heger, der mir mit dem ganzen Team des Aninstituts „Messen-Forschen-Prüfen“ zur Seite stand. Hier sind besonders Sebastian Geschwind und Edgar Hoffeins zu erwähnen, die für die immer sehr zügige Realisierung meiner Ideen in Bezug auf Fertigung und Anbringung von Bauteilen sorgten.

Ebenfalls möchte ich mich bei M. Eng. Philipp Engel für die Integration meines Sensors in das bestehende Datenbankmonitoringsystem DABAMOS bedanken.

Bei meinen Kommilitonen möchte ich mich auch bedanken, da sie während des gesamten Studiums mit Rat und Tat zur Seite standen. Außerdem bedanke ich mich bei meinem Kommilitonen Marten Jeschky benennen, welcher sich mit mir intensiv über dieses Masterthema auseinandergesetzt hat. Ebenfalls bedanke ich mich bei Juana Stieber, die mit mir zusammen in einem Informatikprojekt die ersten Kontakte mit diesem Thema hatte.

Ein ganz besonderer Dank gilt meiner Familie, die mich während des ganzen Studiums in allen Lebenslagen tatkräftig unterstützt hat.

Inhalt

1 Einleitung 1
1.1 Motivation für dieses Thema 1
1.2 Überwachungsmessungen als Teilgebiet der Ingenieurgeodäsie 3
1.3 Deformationsmessungen 4
1.4 Geodätisches Monitoring 6
1.5 Structural health monitoring 8
1.6 Multi-Sensor-Systeme 9
1.7 MEMS-Sensoren 10
2 Sensoren und Hardware 12
2.1 Notwendige Sensortypen 12
2.1.1 Temperatursensor *Linker Temperature* 13
2.1.2 Beschleunigungssensor *Linker 3-Axis-Accelerometer (ADXL335)* 14
2.1.3 Beschleunigungssensor *ADXL354* 14
2.1.4 Beschleunigungssensor *ADXL355* 15
2.2 Möglichkeiten der Datenübertragung 17
2.2.1 *Linker Kit* Erweiterungsboard für den Raspberry Pi 17
2.2.2 Mikrocontroller *Arduino* 18
2.2.3 Andere Mikrocontroller 20
2.3 Der *Raspberry Pi* 22
2.4 Kostenfaktor 22
3 Kommunikation mit den Sensoren 24
3.1 Schnittstellen 24
3.1.1 RS232 25
3.1.2 USB 27

3.1.3 I²C ... 28
3.2 Auslesen der Daten ... 29
3.2.1 Temperatursensor *Linker Temperature* ... 31
3.2.2 Beschleunigungssensor *Linker 3-Axis-Accelerometer* ... 31
3.2.3 Beschleunigungssensor *ADXL354* ... 32
3.2.4 Beschleunigungssensor *ADXL355* ... 33
3.3 Probleme bei der Auflösung der AD-Wandlung ... 33
3.3.1 Mikrocontroller Arduino *Due* und AD-Wandler *ADS1115* ... 33
3.3.2 Änderung des Spannungsbereiches des AD-Wandlers ... 34
3.3.3 AD-Wandlung des Beschleunigungssensors *ADXL355* ... 34
4 Programmierung und Bau der Sensoren ... 35
4.1 Programmierung ... 35
4.1.1 Wahl der Programmiersprachen ... 35
4.1.2 Nutzung der Programmiersprache Python am Beispiel des Raspberry Pi ... 36
4.1.3 Nutzung der Programmiersprachen C und C++ am Beispiel der Arduinoplattform ... 37
4.1.4 Anforderungen an die Auslese- und Verarbeitungsprogramme ... 38
4.1.5 Entwicklung des Programmschemas ... 39
4.1.6 Umsetzung des Programmschemas ... 40
4.1.7 Fehlerbehebung und Optimierung ... 42
4.2 Entwurf und Konstruktion der Sensorblöcke ... 45
4.2.1 Anforderungen an die Hardware ... 45
4.2.2 Schnittstellen ... 46
4.2.3 Stromversorgung ... 47
4.2.4 Konstruktion eines Prototyps ... 48
5 Zeitreihenanalyse ... 49
5.1 Die Zeitreihe im Zeitbereich ... 50
5.1.1 Kovarianzfunktionen ... 50

5.1.2 Datenaufbereitung 51
5.2 Die Zeitreihe im Frequenzbereich 54
5.2.1 Fouriertransformation 55
5.3 Die Zeitreihe im Zeitbereich und Frequenzbereich 58
5.3.1 Kurzzeit Fouriertransformation 59
5.3.2 Wavelet Transformation 61
6 Testmessungen der Sensoren 64
6.1 Erste Tests und Plausibilitätsprüfung 64
6.2 Prüfung der Sensorparameter 65
6.2.1 Datenraten 65
6.2.2 Genauigkeiten 66
6.3 Driftverhalten 68
6.4 Zuverlässigkeit 69
6.5 Test auf Detektion einer simulierten Bauwerksdeformation 69
6.6 Bestimmung der Amplitude der Auslenkung 71
7 Kalibrierung 73
7.1 Kalibrierung des Temperatursensors 73
7.2 Kalibrierung des Temperaturabhängigkeit 74
7.3 Kalibrierung der Abhängigkeit von der Eingangsspannung 74
7.4 Kalibrierung der Ausgangswerte auf Normalschwere 75
7.5 Vergleich mit einem Referenzgerät 76
8 Messung an der Marienkirche als Referenzobjekt 77
8.1 Das Messobjekt 77
8.2 Messaufbau 78
8.3 Auswertung und Vergleich mit den anderen Messsystemen 80
9 Treppenüberwachung in der Hochschule Neubrandenburg 84
9.1 Das Messobjekt 84

9.2 Messaufbau 85
9.3 Auswertung 86
10 Bauwerksmonitoring in Waren/Müritz 89
10.1 Das Messobjekt 89
10.2 Messaufbau 90
10.3 Auswertung 92
11 Monitoring des Turmes der St. Peter Kirche in Altentreptow 96
11.1 Das Messobjekt 97
11.2 Messaufbau 98
11.3 Auswertung 99
12 Zusammenfassung und Ausblick 103
12.1 Sensorkonstruktion und Anwendung 103
12.2 Anwendung der Kurzzeitfouriertransformation 107
12.3 Ausblick 110
13 Literaturverzeichnis 112
14 Tabellenverzeichnis 116
15 Abbildungsverzeichnis 117

Anhang A Arduino Sketch 121
Anhang B Python Programm 123
Anhang C Matlab Programm 125

1 Einleitung

1.1 Motivation für dieses Thema

Diese Masterarbeit befasst sich mit dem ingenieurgeodätischen Monitoring. Bei diesen permanenten Überwachungsmessungen werden entweder bestehende Bauwerke oder Neubauten überwacht. Dieser Bereich hat ein hohes Zukunftspotential. Der Markt für das ingenieurgeodätische Monitoring ist in einem ständigen Wachstum begriffen. Neue Ingenieursbauwerke können schon ab Baubeginn mit Monitoring-Systemen ausgestattet werden.

Abbildung 1.1: Rügenbrücke [Knopf, 2013]

Neben der Überwachung von neuen Bauwerken stellt das Monitoring vieler bestehender historischer Bauwerke einen wichtigen Zweig der Ingenieurgeodäsie dar. So haben viele der Straßen- und Autobahnbrücken in Deutschland ihre ausgelegte Zeit überschritten. Um dennoch einen sicheren Betrieb dieser Bauwerke zu gewährleisten, muss in Zukunft vor allem auf Überwachung gesetzt werden. Ebenfalls ist der Einsatzbereich bei Bahnbauwerken hervorzuheben. Auch hier sind etliche Brückenbauwerke nicht mehr in einem zuverlässigen

Zustand. Zusätzlich zu den Brücken stellen viele andere Bauwerke, wie Kirchen und Sportanlagen ein wichtiges Einsatzgebiet dar. Mit dem Einsturz des Daches der Eislaufhalle in Bad Reichenhall am 2. Januar 2006 ist die Problematik des Bauwerks-Monitoring auch in die breite Öffentlichkeit gerückt.

Abbildung 1.2: Eislaufhalle in Bad Reichenhall [n-tv, 2011]

Um den Einsatz des ingenieurgeodätischen Monitorings für viele Anwendungsbereiche attraktiver zu machen, ist eine Kosteneinsparung der zu verwendenden Sensoren erforderlich. Die klassischen geodätischen Sensoren, wie z.B. Tachymeter oder GNSS sind in der Anschaffung sehr teuer. Hier ist immer mit Kosten im fünfstelligen Bereich zu rechen. In dieser Masterarbeit soll der Grundstein für die Entwicklung preiswerterer Sensorblöcke gelegt werden. Hier sollen die Kosten für einen Sensorblock nur im zwei- bis dreistelligen Bereich liegen. Die Tauglichkeit bestimmter Sensoren ist zu untersuchen. Dafür müssen diese zuerst ausgelesen und die Daten ausgewertet werden, um die Ergebnisse mit hochpreisigen Sensoren vergleichen zu können. Daher ist als erster Schritt die Kommunikation zu diesen Sensoren herzustellen und die Daten auszulesen. Probleme der Datenübertragung und Datenwandlung sollen hier in besonderem Maß betrachtet werden.

Aus dem Bereich der Ingenieurgeodäsie sind folgende Forderungen zu beachten:

- Erfassung von Strukturen mit kontinuierlichen Beobachtungen über lange Zeiträume.
- Sofortige Verfügbarkeit der Ergebnisse einschließlich Analyse.
- Allgemein verständliche graphische Präsentation der Ergebnisse.

1.2 Überwachungsmessungen als Teilgebiet der Ingenieurgeodäsie

Die Helmertsche Definition der Geodäsie als „Wissenschaft von der Ausmessung und Abbildung der Erdoberfläche“ und die Unterteilung in höhere und niedere Geodäsie lässt noch keinen Schluss auf das Tätigkeitsfeld der Ingenieurvermessung zu. Die einfachen Aufgaben des Feldmessens und Nivellierens wurden im späten 19. Jahrhundert zur niederen Geodäsie gezählt. Erst im 20. Jahrhundert und dort speziell nach dem 2. Weltkrieg entwickelte sich die Ingenieurvermessung als eigenständiges Fachgebiet. Durch den Aufschwung des Bauwesens und den daraus entstehenden wachsenden Anforderungen im Qualitätsbereich formierte sich dieser spezielle Bereich. „Zur Ingenieurvermessung gehören alle Vermessungsaufgaben, die in Verbindung mit der Planung, Absteckung, Aufnahme und Überwachung von technischen Objekten (oder deren Umgebung) durchzuführen sind“. [Rinner, 1978] Seit den 1980er Jahren spaltet sich die Industrievermessung davon ab. Hier geht es vor allem um die Vermessung im Maschinen-, Automobil-, Flugzeug- und Schienenfahrzeugbau, sowie dem Erfassen von Oberflächen und Formen. Die „klassische“ Ingenieurvermessung befasst sich hiervon abgrenzend mit technischen Objekten, die bodenverbunden sind.

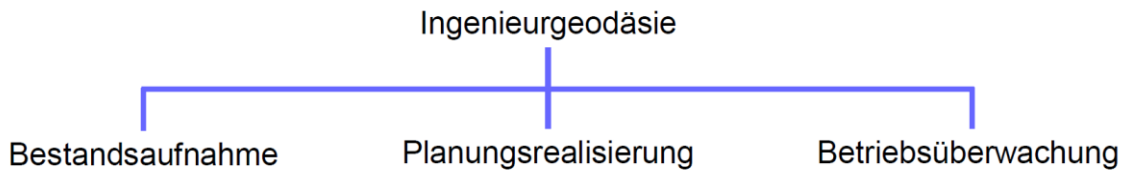


Abbildung 1.3: Teilgebiete der Ingenieurgeodäsie

Aus Abbildung 1.3 wird ersichtlich, dass sich die Ingenieurvermessung mit dem kompletten Zyklus eines technischen Bauwerkes von der Bestandsaufnahme als Grundlage der Planung,

die Planung selbst und deren Realisierung bis hin zur Überwachung des Bauwerkes befasst. Die Aufgabenbewältigung ist gekennzeichnet durch die Koordination des Vermessungsingenieurs mit anderen Ingenieursdisziplinen, vor allem dem Bauingenieurwesen. Ausgehend von den Genauigkeitsanforderungen muss die technische und wirtschaftliche Realisierung eines Projektes durch Auswahl von Messverfahren und Arbeitseinsatz festgelegt werden. Der Produktionsablauf oder Bauablauf darf durch die Arbeit nur möglichst wenig beeinträchtigt werden. [Möser, et al., 2012]

1.3 Deformationsmessungen

Deformationen sind geometrische Veränderungen des Messobjekts. Diese Messobjekte sind zumeist Bauwerke oder Baumwerksteile, dazu können aber auch im weiteren Sinne maschinenbauliche Anlagen gehören. Die Messung und Auswertung von geometrischen Veränderungen dienen der Bauwerkssicherung, der Verlängerung der Nutzbarkeit und nicht zuletzt zum Schutz des Menschen. Zu den Deformationen zählen neben den Formveränderungen auch die Lageveränderungen des Messobjekts. [Witte, 2000]

Tabelle 1.1: Arten von Deformationen

Art der Deformation	Unterteilungen
Verschiebungen	- vertikal - horizontal
Verdrehungen	
Neigungen	
Verformungen	- Längenänderung - Biegung - Torsion

Ein wichtiges Kriterium einer Deformationsmessung ist die Auswahl der Messpunkte. Hierbei muss darauf geachtet werden, dass der Sensor direkten Kontakt mit dem Messobjekt aufweist und die zu erwartende Deformation in vollem Umfang mit ausführt.

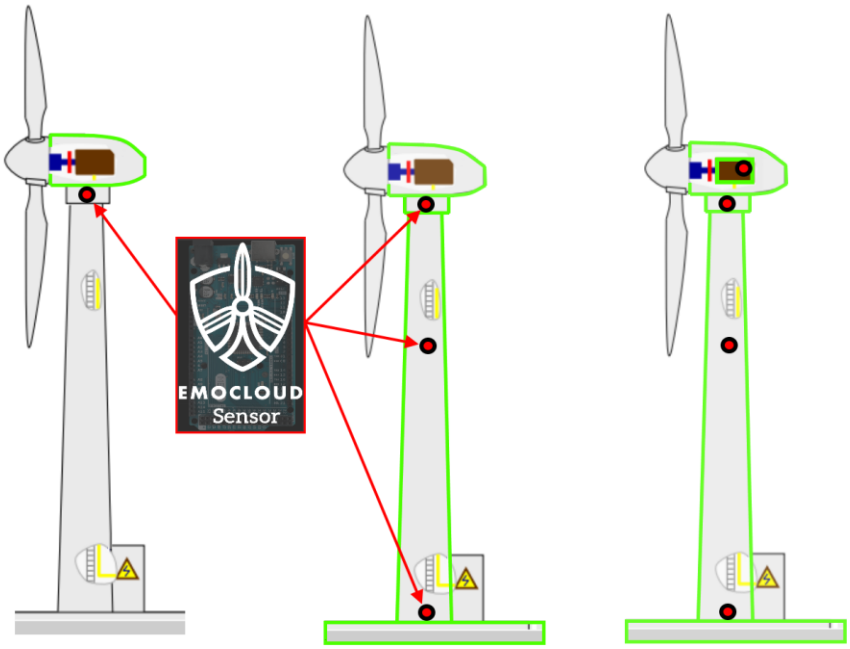


Abbildung 1.4: Messpunkte an einer Windenergieanlage

Wie in Abbildung 1.4 dargestellt ist, erfordert insbesondere die Überwachung komplexer Bauwerke eine große Anzahl an Messpunkten, um ein möglichst komplexes Deformationsverhalten nachweisen zu können. Die Deformation einer Windenergieanlage wird sich vor allem in einer Biegung des Turmes darstellen. Da die Biegelinie des Turmes keine einfache Gerade ist, werden für die Bestimmung der Koeffizienten der Modelldarstellung der Biegelinie mindestens drei Sensoren benötigt.

Bei einer Deformationsmessung wird zwischen kontinuierlichen Messungen und diskontinuierlichen Messungen unterschieden. Diskontinuierliche Messungen, auch epochenweise Messungen genannt, lassen sich dann realisieren, wenn die auftretenden Kräfte und das dazugehörige Objektverhalten näherungsweise bekannt sind. Beispiele hierfür sind das Füllen eines Staudammes oder die Auswirkungen des Tidenhubs im Hafenbereich. Kontinuierliche Messungen werden dann angewendet, wenn ein nicht vorhersehbarer Deformationsverlauf zu befürchten ist. Dies kann unkontrollierbare Belastungsänderungen, wie Windeinfluss oder Verkehrseinfluss, zur Ursache haben. Die Erfassung dieser so verursachten Deformationen setzt die Messung in kontinuierlichen Abständen voraus. [Witte, 2000]

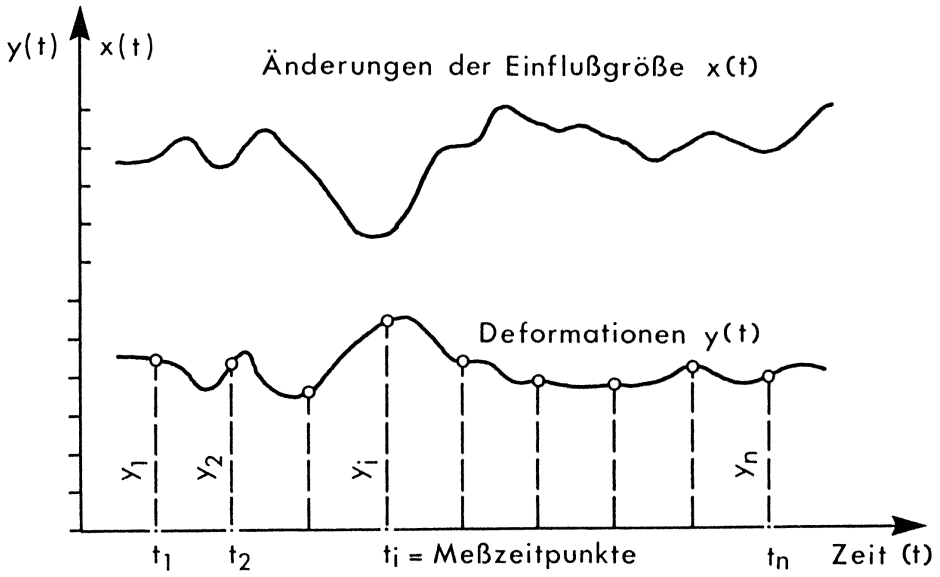


Abbildung 1.5: kontinuierliche Deformationsmessung [Witte, 2000]

Eine zeitkontinuierliche Messung ist am einfachsten mit einem automatisiertem Messablauf und fest verbauten Messsystemen zu realisieren. Wird das Messobjekt permanent überwacht, spricht man von einem geodätischen Monitoring.

1.4 Geodätisches Monitoring

Der dritten Säule der Abbildung 1.3, der Ergebnisüberwachung soll im Zusammenhang mit dieser Masterarbeit besondere Aufmerksamkeit gewidmet werden. Die geodätischen Überwachungsmessungen befassen sich mit der Planung, Durchführung, Auswertung und Dokumentation von Messvorhaben die Objektveränderungen beinhalten. Hierbei werden geometrische Objektveränderungen als Bewegungen und Verformungen erfasst. Das Ziel der Überwachungsmessungen ist der Nachweis des Verhaltens ausgewählter Objektpunkte. Dieses Verhalten kann entweder den zu erwartenden Veränderungen entsprechen oder abweichen. So werden geometrische Beziehungen der gewählten Objektpunkte quantisiert. Damit wird der Deformationszustand des überwachten Bauwerkes dokumentiert. Nachfolgend sind die wesentlichen Ziele der Bauwerksüberwachung aufgeführt:

- Nachweis der Stand- und Funktionssicherheit des zu überwachenden Bauwerkes, sowie der störungsfreie Betrieb
- Minderung des Gefährdungspotentials „Frühwarnung“ durch rechtzeitiges Erfassen von Veränderungen

- Beweissicherung zur Schadensdokumentation und zur Klärung der Ursachen von Schäden
- Möglichkeit zur Prognose des mutmaßlichen Verhaltens in der näheren Zukunft
- Überprüfung von Konstruktions- und Materialeigenschaften mit dem Ziel der Verbesserung der mechanischen Modellvorstellung
- Erkenntnisgewinn und Übertragbarkeit auf vergleichbare Objekte

Zur Realisierung von geodätischen Überwachungsmessungen werden zunehmend Multi-Sensor-Systeme verwendet.

Das ingenieurgeodätische Monitoring ist in zwei unterschiedliche Themengebiete einzuteilen, das Bauwerks-Monitoring und das Geo-Monitoring. In Tabelle 1.2 ist eine Übersicht über diese Monitoring-Bereiche aufgestellt worden.

Tabelle 1.2 Übersicht über die geodätischen Monitoring-Bereiche

Bauwerks-Monitoring	Geo-Monitoring
Brücken	Rutschhänge
Hochhäuser	Steilküsten
Talsperren	Grabenbrüche
Tunnel	Senkungsgebiete
Bergwerke	Ästuare

Aus den verschiedenen Anwendungsgebieten ergeben sich unterschiedliche Anforderungen an die zu beobachtenden Objekte. Die Aufgaben können in verschiedene Kategorien unterteilt werden. Dies wird in Tabelle 1.3 verdeutlicht.

Tabelle 1.3: Übersicht der Anforderungen an das Monitoring [Staiger, 2009]

Zweck der Messung	Beispiele	Art der Bewegung	Größen-ordnung der Bewegung	Anforderungen an…				
				Latenzzeit	Messzeit pro Epoche	Synchronisation	Wiederholungs-rate	Messgenauig-keit
Beschreiben von Verände-rungen	Geotektonik	kriechend, abrupt	0 – 20 mm pro Jahr	0	1	1	1	[5]
	Gletscher-bewegung	gleichförmig	0 – 10 m pro Jahr	0	1	1	1	2 – 4
	Gezeiten	kontinuierlich	bekannt	0	3	2	3	1 – 5
	Materialtests, Instrumenten-untersuchungen	kriechend-abrupt plastische Verformung	<100 mm/s	0	3 – 5	3 – 5	3 – 5	4 – 5
Stand- und Funktions-sicherheit von Bau-werken oder Gelände-abschnitten	Talsperren, Brücken, Hochhäuser, usw.	Elastisch, abhängig von Tages- und Jahreszeit, Füllgrad, …	0 – 20 mm pro Jahr	3	2	2	2	2 – 5
	Hang-rutschung	kriechende Rutschung, Felsabbruch, usw.	0 – 50 mm pro Jahr	4	3	3	3	2 – 5
Prozess-steuerung	Steuerung einer Bau-maschine	gleichförmig	0 – 10 m/s	5	5	4	5	3 – 5

1.5 Structural health monitoring

Unter *Structural Health Monitoring (SHM)* wird die kontinuierliche oder die periodische und die automatisierte Methode zur Bestimmung und Überwachung des Zustandes eines Überwachungsobjektes innerhalb der Zustandsüberwachung verstanden. [DIN, 2018]

Insbesondere komplexe Strukturen von Messobjekten müssen kontinuierlich überwacht werden. Zu diesen Messobjekten können Großbauwerke, große Maschinen, Windenergieanlagen oder auch Flugzeuge gehören. Diesen Objekten ist gemein, dass sie großen andauernden mechanischen Belastungen unterworfen sind. Ihre mechanischen Eigenschaften können geschwächt werden und es kann zum Ausfall eines Bauteils oder gar zur Zerstörung des Objektes kommen. Um dies zu verhindern werden in zunehmenden Maße automatische Sensorsysteme zu permanenten Überwachung eingesetzt. Mittels

Structural-Health-Monitoring-Systemen können auftretende Unregelmäßigkeiten nicht nur detektiert und lokalisiert, sondern auch beurteilt werden. Solche intelligente *Structural-Health-Monitoring-Systeme*, die Prognosen ableiten, sind aber noch im Forschungsstadium. [Fraunhofer-Institut für Nauturwissenschaftlich-Technische Trendanalysen, 2015]

Zu einem *Structural-Health-System* gehören neben den Messsensoren auch die Datenspeicher und das Auswertesystem. Es soll die Schadensdetektion, die Bestimmung der Art des Schadens bis hin zur Beurteilung der Auswirkungen erfolgen. Um diesen Anforderungen gerecht zu werden, ist es erforderlich eine Vielzahl von Sensoren zu verwenden, da möglichst viele Einflussfaktoren auf die zu erwartende Zustandsänderung erfasst werden sollen.

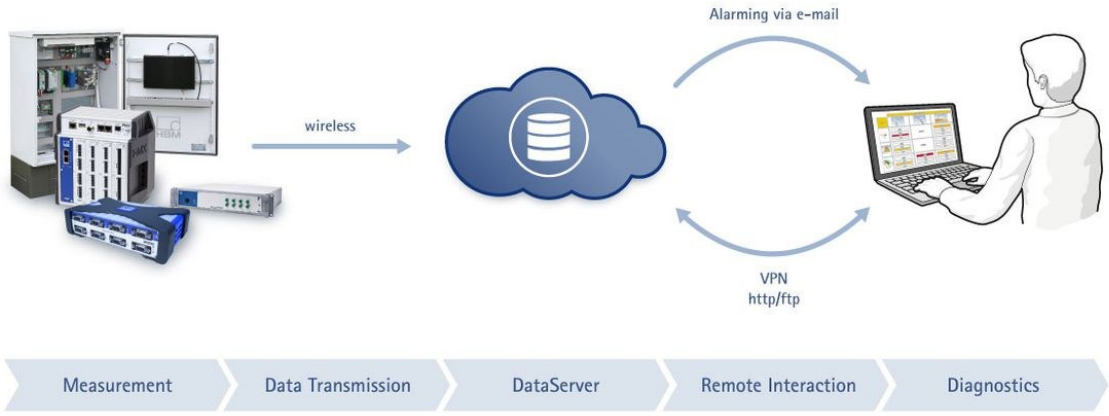


Abbildung 1.6: Bestandteile und Eigenschaften eines Structural-Health-Monitoring-System [HBM, 2018]

1.6 Multi-Sensor-Systeme

Ein Sensor ist ein Bauteil, welches physikalische Eigenschaften erfasst und in elektrische Signale umformt. Diese werden eingeteilt in aktive und passive Sensoren. Aktive Sensoren wandeln die empfangene Energie, z.B. Licht mittels des Photoeffektes, direkt in elektrische Signale um. Passive Sensoren benötigen hingegen eine Versorgung mit elektrischer Energie. [Larisch, 2013]

Zusätzlich wird zwischen digitalen und analogen Sensoren unterschieden. Bei analogen Sensoren sind die Ausgangsgrößen elektrische Signale, während bei digitalen Sensoren das Signal schon im Sensor selbst in eine digitale Form umgewandelt wird.

Eine Vielzahl von Sensoren ist in den gängigen Messgeräten verbaut. So benötigt ein

Tachymeter mindestens einen Sensor zur Erfassung der Distanz und zwei Sensoren zur Erfassung der Horizontal- und Vertikalrichtungen. Damit ist selbst ein Tachymeter schon ein Multi-Sensor-System. Daran ist ersichtlich, dass das Messen heutzutage fast ausschließlich mit Multi-Sensor-Systemen erfolgt.

Tabelle 1.4: Einteilung von Multi-Sensor-Systemen [Schweiger, et al., 2014]

Bezeichnung	räumlich verteilt	redundant	sich ergänzend
Charakteristika	gleiche Sensoren erfassen dieselben Messgrößen an verschiedenen Orten	unterschiedliche Sensoren erfassen dieselbe Messgröße am selben Ort mit unterschiedlichen Messprinzipien	unterschiedliche Sensoren erfassen unterschiedliche Messgrößen am selben Ort, die Kombination der Messgrößen ist zur Berechnung der Zielgröße erforderlich
Beispiel	GNSS-Empfänger bestimmen die Koordinaten einer Staumauerkrone an zehn verschiedenen Orten	die Koordinaten eines Fahrzeuges werden gleichzeitig mittels GNSS-Empfängern und Tachymeter bestimmt	die mit einem Tachymeter bestimmten Koordinaten setzen eine Strecken- und zwei Richtungsmessungen voraus

1.7 MEMS-Sensoren

MEMS ist die Abkürzung für *Micro-Electro-Mechanical-Systems*. Im deutschen Sprachraum ist neben diesem aus den USA stammenden Begriff auch die Bezeichnung Mikrosystem gebräuchlich. Dieser Überbegriff bezeichnet ein Bauteil, dessen Elemente aus sehr kleinen Schaltungen und mechanischen Strukturen bestehen, welche eine Größe in Mikrometerbereich aufweisen. Neben Sensoren und Aktoren können die Mikrosysteme auch Oszillatoren und Filter sein. [Wild-Pfeiffer, et al., 2011]

Hier sollen aber nur die Sensoren betrachtet werden. Bei den Sensoren sind die Anwendungsbereiche sehr mannigfaltig. Es gibt Sensoren für Temperatur, Druck, Feuchte oder Beschleunigung. In dieser Masterarbeit sollen nur Temperatursensoren und Beschleunigungssensoren betrachtet werden. Die letztgenannten Sensoren gehören in die Gruppe der Inertialsensoren. Inertialsensoren können die Beschleunigung eines Körpers gegenüber einem inertialen Bezugssystem wahrnehmen. So kann im dreidimensionalen

Raum mit der Kombination von drei Beschleunigungssensoren, welche orthogonal zueinander orientiert sind, die Beschleunigung gemessen werden. Außerdem muss zusätzlich mit Hilfe von Kreiseln oder allgemein Winkelgeschwindigkeitssensoren die Orientierung des Körpers bestimmt werden. [Knickmeyer, 2014]

Diese werden bei MEMS-Sensoren als Drehratensensoren oder Gyros bezeichnet. Sensoren, welche sowohl Beschleunigungen und Drehraten messen, sind häufig als *6DoF* für *‚six degrees of freedom‘*, also sechs Freiheitsgrade bezeichnet, siehe Abbildung 1.7.

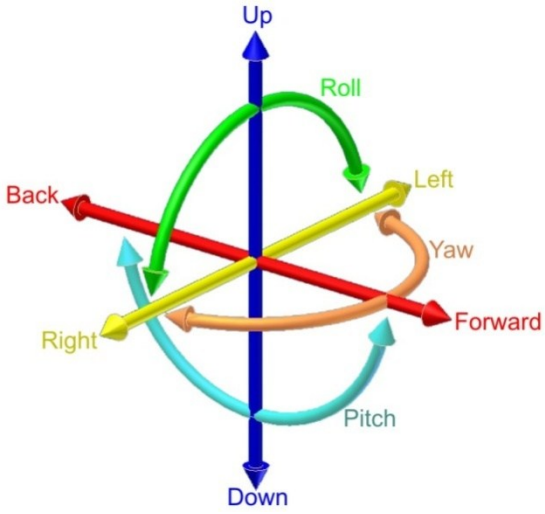


Abbildung 1.7: Prinzip der Messung von sechs Freiheitsgraden [Ionescu, 2016]

Viele MEMS-Beschleunigungssensoren sind nach dem Feder-Masse-Prinzip aufgebaut. Die folgende Abbildung 1.8 stellt dies schematisch dar.

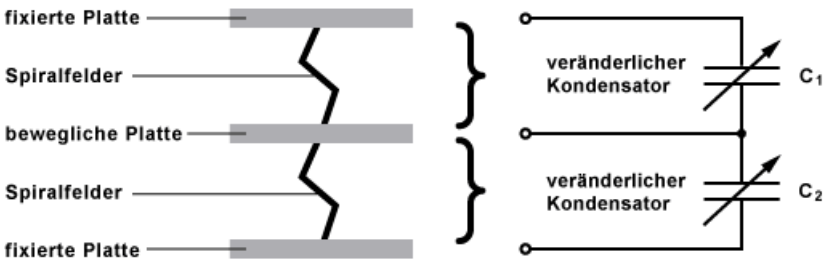


Abbildung 1.8: Aufbau eines MEMS-Beschleunigungssensors [Elektronik, 2016]

Der Aufbau dieses MEMS-Beschleunigungssensors sieht drei gestapelte Platten vor, die über Spiralfedern miteinander verbunden sind. Die äußeren Platten sind fest und die mittlere beweglich. Ihre Beweglichkeit wird durch die Spiralfedern begrenzt. Durch diesen Aufbau

entsteht eine Reihenschaltung von zwei Kondensatoren mit veränderlicher Kapazität. Veränderlich deshalb, weil sich der Plattenabstand der beiden Kondensatoren durch die mittlere Platte verändern kann. Bekanntlich beeinflusst der Plattenabstand die Kapazität eines Kondensators, vergleiche hierzu Abbildung 1.9.

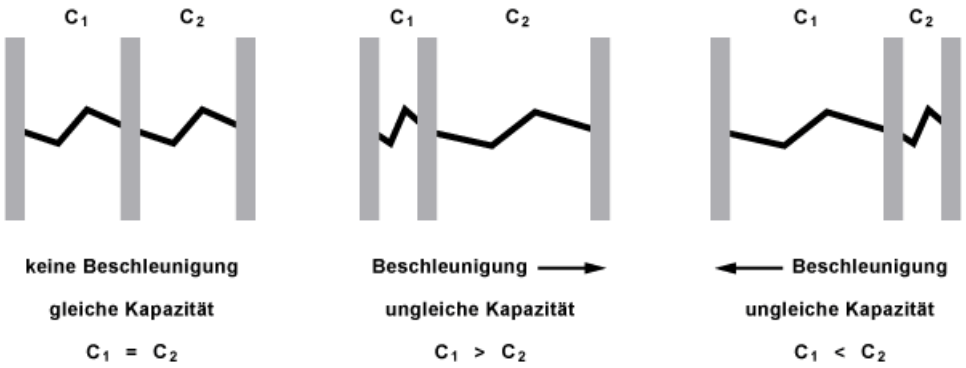


Abbildung 1.9: Funktionsweise eines MEMS-Beschleunigungssensors [Elektronik, 2016]

Wenn keine Bewegung stattfindet oder die Beschleunigung Null ist, dann befinden sich die Platten in gleichem Abstand zueinander. Findet nun eine Bewegung, also eine Beschleunigung statt, dann wird die mittlere Kondensatorplatte zur Seite gedrückt. Dadurch entsteht eine Kapazitätsänderung. Diese Kapazitätsänderung verläuft proportional zur Beschleunigung. Wenn eine konstante Geschwindigkeit eingehalten wird, dann findet zwar immer noch eine Bewegung statt, aber keine Beschleunigung mehr. Die mittlere Kondensatorplatte geht in ihre Ursprungsposition zurück. [Elektronik, 2016]

2 Sensoren und Hardware

2.1 Notwendige Sensortypen

In dieser Masterarbeit werden zwei verschiedene Sensoren auf ihre Tauglichkeit für ingenieurgeodätische Anwendungen getestet. Zuerst wird ein Temperatursensor verwendet. Dieser ist zwar kein vollwertiger MEMS-Sensor, aber die korrekte Messung der Temperatur ist für viele Geodäsie- und Messtechnikanwendungen von besonderer Bedeutung. Außerdem wird ein Sensor aus dem Bereich der Beschleunigungssensoren verwendet. Hier liegt der Haupteinsatzbereich der MEMS-Sensoren. Daher wird ein 3-achsiger Beschleunigungssensor getestet, welcher ein Einsatzgebiet innerhalb der

Bauwerksüberwachung haben könnte.

2.1.1 Temperatursensor *Linker Temperature*

Der Temperatursensor hat eine Größe von 34 mm x 24 mm inklusive Platine. Der eigentliche Sensor hat dagegen nur eine Größe von einigen Millimetern. Dieser Sensor gibt nur ein analoges Datensignal ab. Dieses muss mittels eines Analog-Digital-Wandlers in ein digitales Signal umgewandelt werden. Die Pin-Belegung ist denkbar einfach. Neben der Versorgungsspannung (PIN 2: VCC) und Masseverbindung (PIN 1: GND) wird hier nur ein analoger Pin zur Datenausgabe (PIN 4: OUT) genutzt. Der vierte Pin (PIN 3: NC) ist nicht belegt, vgl. Abbildung 2.1.

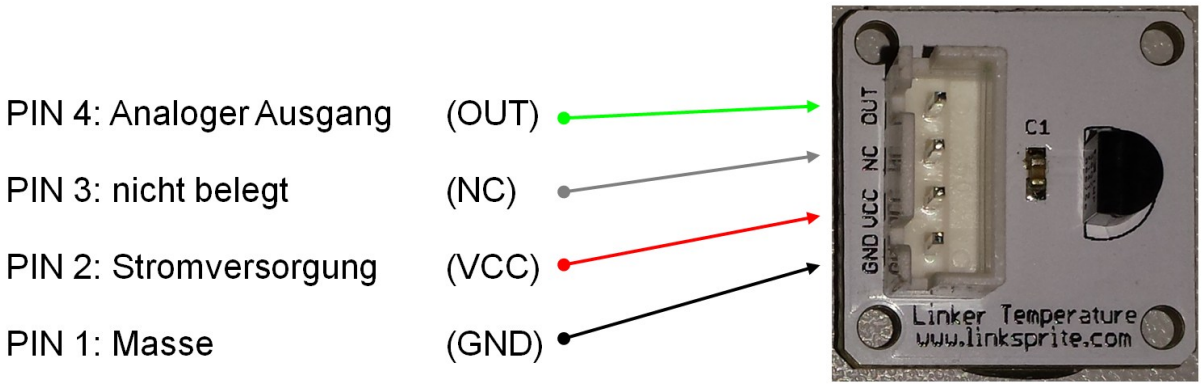


Abbildung 2.1: Pinbelegung Linker Temperature Temperatursensor

Dieser Sensor ist im einschlägigen Versandfachhandel für einen Betrag von ca. 5 € pro Stück erhältlich. Ein Vorteil an dieser Ausführung ist der integrierte Stecker, mit welchem der Sensor direkt an die Linker Kit Erweiterungsplatine angeschlossen werden kann. Diese Platine sollte nach den Vorüberlegungen auch verwendet werden, um diesen Sensor mit dem Einplatinencomputer *Raspberry Pi* zu verbinden. Darauf wurde in weiteren Verlauf verzichtet, weitere Ausführungen hierzu werden im Kapitel 3.3 beschrieben. Das Temperatursignal wird als Spannungswert von 0 bis 5 Volt ausgegeben. Nach dem Datenblatt entspricht die Spannung 750 mV der Temperatur von 25 °C und 1 °C entspricht einer Spannungsdifferenz von 10 mV. Daher muss der zu verwendende AD-Wandler eine Auflösung von mindestens 10 mV haben. Dies wird ebenfalls im Kapitel 3.3 näher erläutert.

2.1.2 Beschleunigungssensor *Linker 3-Axis-Accelerometer (ADXL335)*

Der Beschleunigungssensor hat ebenfalls eine Größe von 34x24 Millimeter inklusive Platine. Es können Beschleunigungen in drei aufeinander senkrecht stehenden Achsen X, Y und Z gemessen werden. Dieser Sensor gibt ebenfalls ein analoges Datensignal ab. Hier sind 5 Pins belegt. Neben der Masseverbindung (PIN 1: GND) und der Versorgungsspannung (PIN 2: VCC) wird hier für jede Achse ein analoger Pin zur Datenausgabe (PIN 3, 4, 5: Z, Y, X) genutzt, vergleiche hierzu Abbildung 2.2.

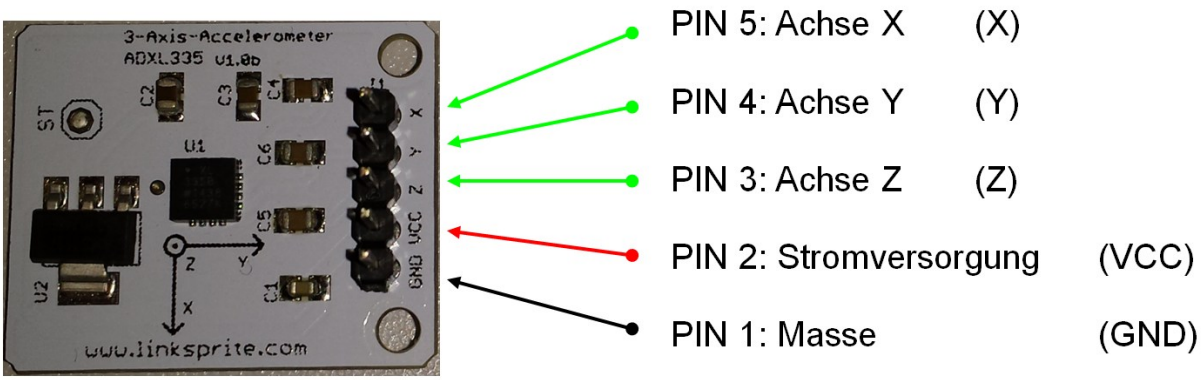


Abbildung 2.2: Pinbelegung Linker 3-Axis-Accelerometer

Das Beschleunigungssignal wird als Spannungswert von 0 bis 3,3 V ausgegeben. Die Spannung von 1,150 V entspricht der Beschleunigung von 0 m/s². Für eine Beschleunigung von ca. 10.00 m/s² (1 g) steigt die Spannung um 200 mV an. Somit liegt der Arbeitsbereich bei ± 60 m/s², also ± 6 g. Um eine Beschleunigung von 1cm/s² messen zu können, muss der AD-Wandler eine Auflösung von 0,2 mV aufweisen.

2.1.3 Beschleunigungssensor *ADXL354*

Der *ADXL354* von Analog Devices ist ebenfalls ein nach MEMS Prinzip arbeitender Beschleunigungssensor. Auch mit diesem Sensor können Beschleunigungen in drei aufeinander senkrecht stehenden Achsen X, Y und Z gemessen werden. Dieser Sensor gibt ebenfalls ein analoges Datensignal ab. Im Gegensatz zum *ADXL 335* ist sein Messrauschen um den Faktor sechs geringer. Der Messbereich ist zwischen ca. ± 20 m/s² und ± 80 m/s² wählbar.

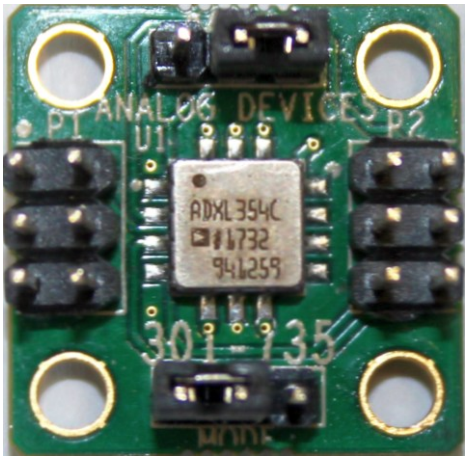


Abbildung 2.3: EVAL-ADXL354 Evaluation-Board

Bei diesem Sensor kann neben den Beschleunigungswerten auch die Temperatur gemessen werden. Die Temperaturmessung direkt im Sensor ist insofern wichtig, weil der Sensor eine Temperaturdrift von ca. 1.5 mm/s² pro Grad Celsius aufweist und diese dann später durch Kalibrierung ausgeschaltet werden kann.

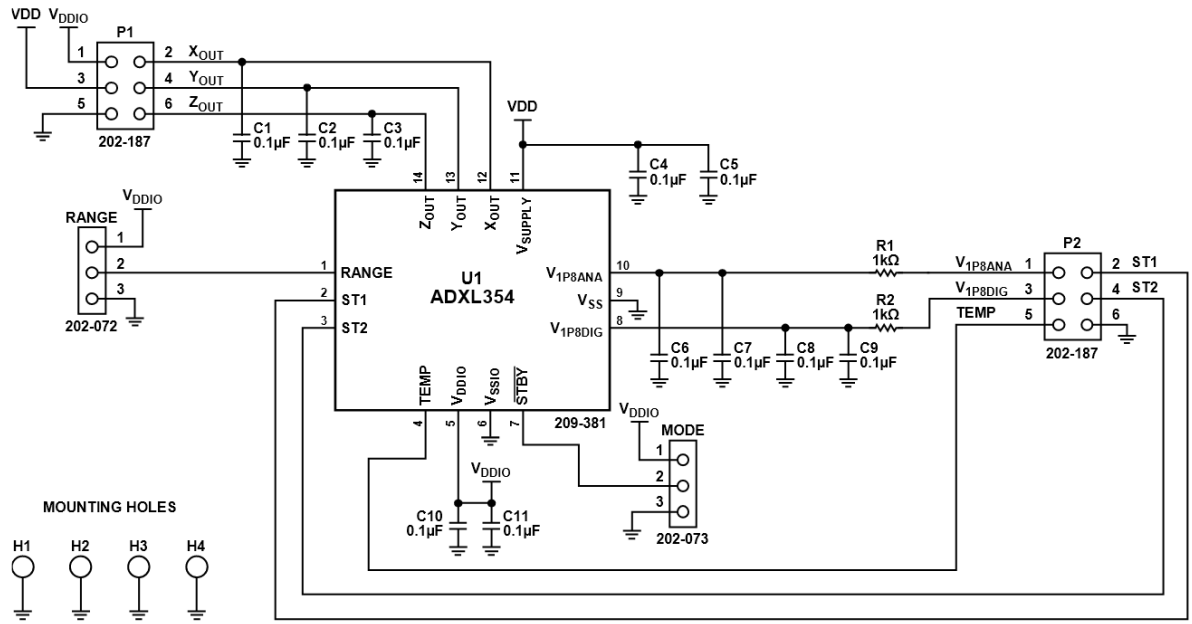


Abbildung 2.4: Schaltbild EVAL-ADXL354 [Analog Devices, 2018]

Innerhalb dieser Arbeit wird das Evaluation-Board *EVAL_ADXL354* genutzt.

2.1.4 Beschleunigungssensor *ADXL355*

Die bisher vorgestellten Sensoren weisen eine analoge Datenübertragung auf. Dies führt zu Problem mit der Analog-Digital-Wandlung. Die Probleme und deren Lösungen sind im Kapitel

3.3 erläutert. Daher ist auch ein Sensor mit digitalem Ausgang verwendet worden. In Form, Größe und Leistung ist der *ADXL355* dem im Kapitel 2.1.3 vorgestellten *ADXL354* vergleichbar. Hier können aber die Sensordaten in digitaler Form über verschiedene Schnittstellen ausgelesen werden.

Abbildung 2.5: EVAL-ADXL355 Evaluation-Board

Auch bei diesem Sensor können neben den Beschleunigungswerten die Temperaturwerte gemessen und mittels der zu Verfügung stehenden Schnittstellen übertragen werden. Der Analog-Digital-Wandler des *ADXL355* hat eine Auflösung von 20 Bit, was einer Abstufung von ca. einer Millionen Werten entspricht. Dieser Wert deutet auf eine höhere erreichbare Auflösung dieses Sensors hin.

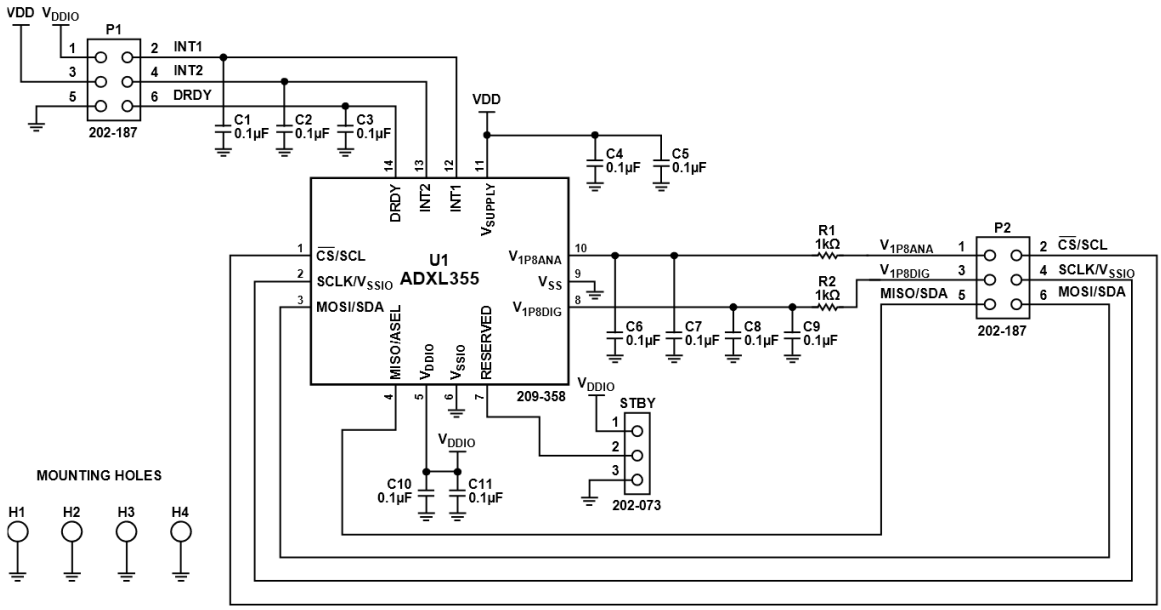


Abbildung 2.6: Schaltbild EVAL-ADXL355 [Analog Devices, 2018]

Auch dieser Sensor ist dieser zusammen mit dem Evaluation-Board *EVAL_ADXL355* verwendet worden.

2.2 Möglichkeiten der Datenübertragung

Einige der zu untersuchenden Sensoren geben analoge Datensignale ab. Um diese nutzen zu können, müssen die analogen Signale in digitale Signale umgewandelt werden. Dazu wird mit einem AD-Wandler gearbeitet. Im Rahmen dieser Masterarbeit werden AD-Wandler mit unterschiedlicher Auflösung getestet. Die Tests mit den AD-Wandlern sind eigentlich nicht vorgesehen gewesen. Da die Sensoren in der Ursprungskonfiguration für geodätische Zwecke nicht zu verwenden waren, ist auf die Probleme mit den AD-Wandlern besonderes Ausgenmerk gerichtet worden, siehe Kapitel 3.3.

2.2.1 *Linker Kit* Erweiterungsboard für den Raspberry Pi

Das Linker Kit Erweiterungsboard ist ein Zusatzmodul für den Einplatinencomputer *Raspberry Pi*, der die Ansteuerung unterschiedlichster Sensoren ermöglicht. Es können sowohl analoge als auch digitale Sensorsignale verwendet werden. Zur Wandlung der analogen Signale ist in dem Erweiterungsmodul *Linker Kit Baseboard* ein AD-Wandler mit einer Auflösung von 10 Bit integriert. Dieses Erweiterungsmodul gibt seine Daten über die SPI Schnittstelle weiter.

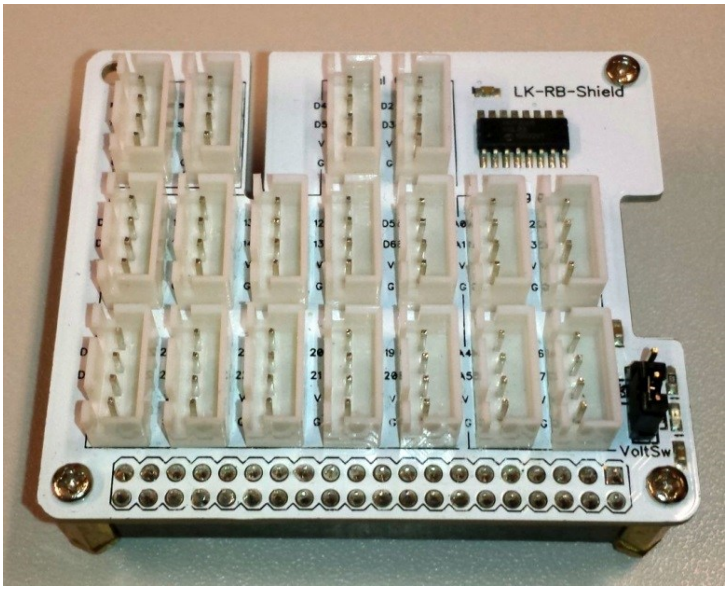


Abbildung 2.7: Erweiterungsmodul Linker Kit Baseboard

2.2.2 Mikrocontroller *Arduino*

Die Daten können auch mit einem *Arduino* Mikrocontroller an den Computer übertragen werden. Dieser Mikrocontroller hat ebenfalls einen AD-Wandler integriert. Die gewandelten Daten werden dann über die USB-Schnittstelle übertragen. Die in Abbildung 2.8 abgebildeten *Arduino* Mikrocontroller eignen sich alle zum Einstieg in die Programmierung. Umfangreiche Bibliotheken und Beispiele erleichtern die Bedienung der Geräte. Ein *Arduino*-Board besteht aus einem Prozessor, Schnittstellen, Buchsen und Steckleisten für Erweiterungen. Typische Anwendungen liegen in Bereichen der Elektronik, Robotik, Ansteuerungen oder der Sensordatenerfassung. [Elektronik, Kompendium, 2017]

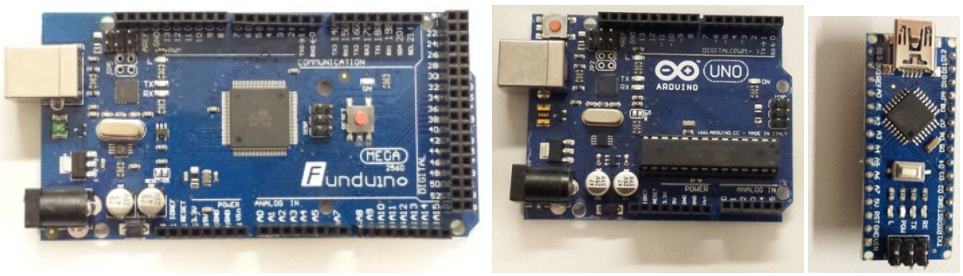


Abbildung 2.8: Mikrocontroller Arduino Mega/ Uno/ Nano

Da sich der Mikrocontroller *Arduino* hauptsächlich zur Sensordatenerfassung und Ansteuerungsaufgaben eignet, stellt sich die Frage, warum hier nicht ein *Raspberry Pi* verwendet werden sollte, der diese Aufgaben auch beherrscht. Der *Arduino* hat im Gegensatz zum *Raspberry Pi* kein Betriebssystem und kann nur die Programme ausführen, welche auf den Chip gespeichert wurden. Speicherung, Grafikausgabe oder Netzwerkanbindung sind nur über Umwege zu realisieren. Daher sind die beiden Systeme keine Konkurrenten, sondern können sich sinnvoll ergänzen.

2.2.2.1 Mikrocontroller *Arduino Nano*

Der in Abbildung 2.9 abgebildete Mikrocontroller *Arduino Nano* ist ein kleines, vollständiges und Breadboard-freundliches Board. Es basiert auf dem ATmega328 und wurde speziell für die Arbeit mit Steckboards entwickelt.

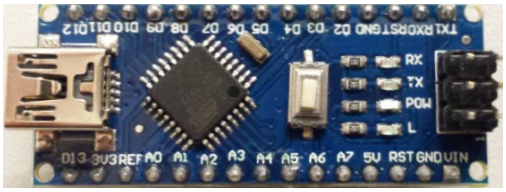


Abbildung 2.9: Mikrocontroller Arduino Nano

Das Board verfügt über eine praktische Größe von ca. 45 mm x 17 mm und Pins, die über Stiftleisten nach unten herausgeführt werden. Dadurch ist es kompatibel mit fast allen Stecksystemen. Es verfügt nahezu über die gleiche Funktionalität wie andere *Arduino* Mikrocontroller, allerdings in einem anderen Formfaktor. Es fehlt lediglich eine DC-Strombuchse und es funktioniert mit einem Mini-B-USB-Kabel anstatt eines normalen Kabels. Die Mini-USB-Buchse ermöglicht sowohl die Stromversorgung als auch die Übertragung von neuen Programmen auf den Controller. Der AD-Wandler des Mikrocontroller *Arduino Nano* hat eine Auflösung von 10 Bit.

2.2.2.2 Mikrocontroller *Arduino Due*

Bei dem Mikrocontroller *Arduino Due* handelt es sich um ein Mikrocontroller-Board, das auf der ATMEL SAM3X CPU basiert. Der AD-Wandler hat eine Auflösung von 12 Bit. Es hat eine Größe von ca. 100 mm x 55 mm. Das Board ist unter anderem mit 54 digitalen Input/Output Pins ausgestattet, ebenso wie mit 12 analogen Inputs, 4 UARTs und einem 84 MHz Quarz Oszillator. Darüber hinaus verfügt es über einen SPI Header, Reset-Button sowie zwei USB-Verbindungen und eine Strombuchse.

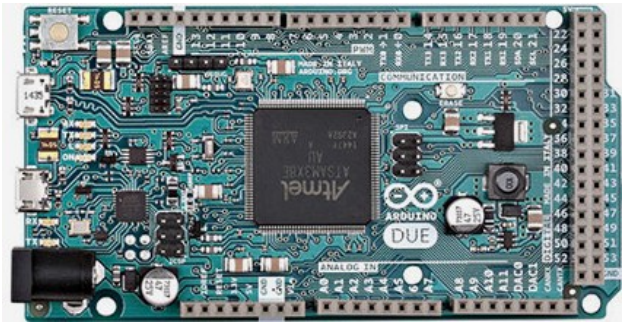


Abbildung 2.10: Arduino Due

2.2.3 Andere Mikrocontroller

Neben den klassischen *Arduino* Mikrocontrollern sind auch andere Mikrocontroller in Betracht gezogen worden. Die Anforderungen an eine hohe Datenübertragungsrate, eine zuverlässige Datenübertragung, eine große Schnittstellenauswahl und einen geringen Preis ließen neben den *Arduino* Boards auch andere Mikrocontroller in die nähere Auswahl kommen.

2.2.3.1 Mikrocontroller *Blue pill*

Als *Blue pill* wird ein kompaktes mit dem Chip STM32F103 bestückten Boards bezeichnet. Wie bei dem Mikrocontroller *Arduino Due* arbeitet hier eine 32 Bit CPU, Die ARM Cortex M3 CPU und der dazugehörende STM32F103 Mikrocontroller ist mit dem Mikrocontroller *Arduino* kompatibel. Das heißt, die *Arduino* Entwicklungsumgebung kann auch mit diesem Board genutzt werden. Durch die hohe Leistung der CPU ist eine hohe Datenübertragungsrate, wie bei dem Mikrocontroller *Arduino Due* zu erwarten. Der Preis dieses Mikrocontrollers liegt bei unter 5 € pro Stück.

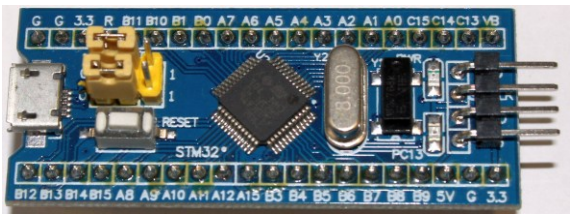


Abbildung 2.11: "Bluepill" Mikrocontroller

Leider ist in diesem Board ein falscher Widerstand werksseitig eingebaut worden. Der USB-Standard erfordert einen 1,5 kΩ Pullup-Widerstand an D+, aber diese Platine hat einen Widerstand mit dem falschen Wert. Er wird entweder mit einem 10 kΩ-Widerstand oder einem 4,7 kΩ-Widerstand geliefert. [Clarke, 2018]

Auch ein Tausch dieses Widerstandes führte nicht zu einer Kommunikation über USB. Daher ist dieser Mikrocontroller nicht weiter verwendet worden.

2.2.3.2 Mikrocontroller *Maple Mini*

Der Mikrocontroller *Maple Mini* ist wie der Mikrocontroller *Blue pill* mit einer 32 Bit CPU ausgestattet. Der verbaute STM32F103 Mikrocontroller ist *Arduino* kompatibel. Das heißt, die *Arduino* Entwicklungsumgebung kann auch mit diesem Board genutzt werden. Wie bei

dem Mikrocontroller *Blue pill* ist eine hohe Datenübertragungsrate zu erwarten. Der Preis dieses Mikrocontrollers liegt bei unter 10 € pro Stück. Die Bauform ist mit ca. 51 mm x 18 mm im Gegensatz zum Mikrocontroller *Arduino Nano* mit ca. 45 mm x 17 mm nur geringfügig größer.

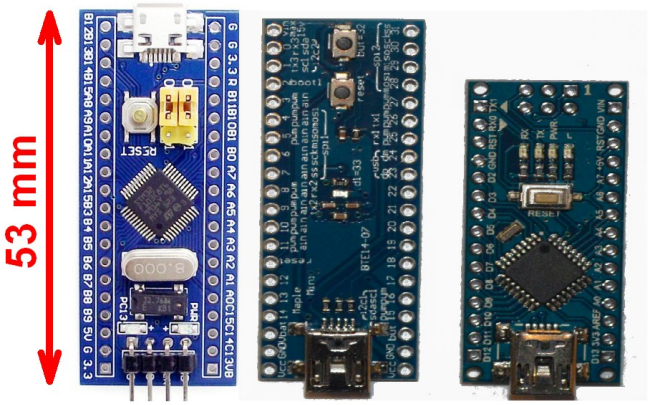


Abbildung 2.12: Bluepill, Maple Mini und Arduino Nano im Größenvergleich

Im Gegensatz zum Mikrocontroller *Blue pill* weist der Mikrocontroller *Maple Mini* eine volle Funktionalität zu USB auf. Durch die kompakte Bauform, die höhere Leistungsfähigkeit und den geringen Preis ist der *Maple Mini* der bevorzugte Mikrocontroller in dieser Masterarbeit geworden.

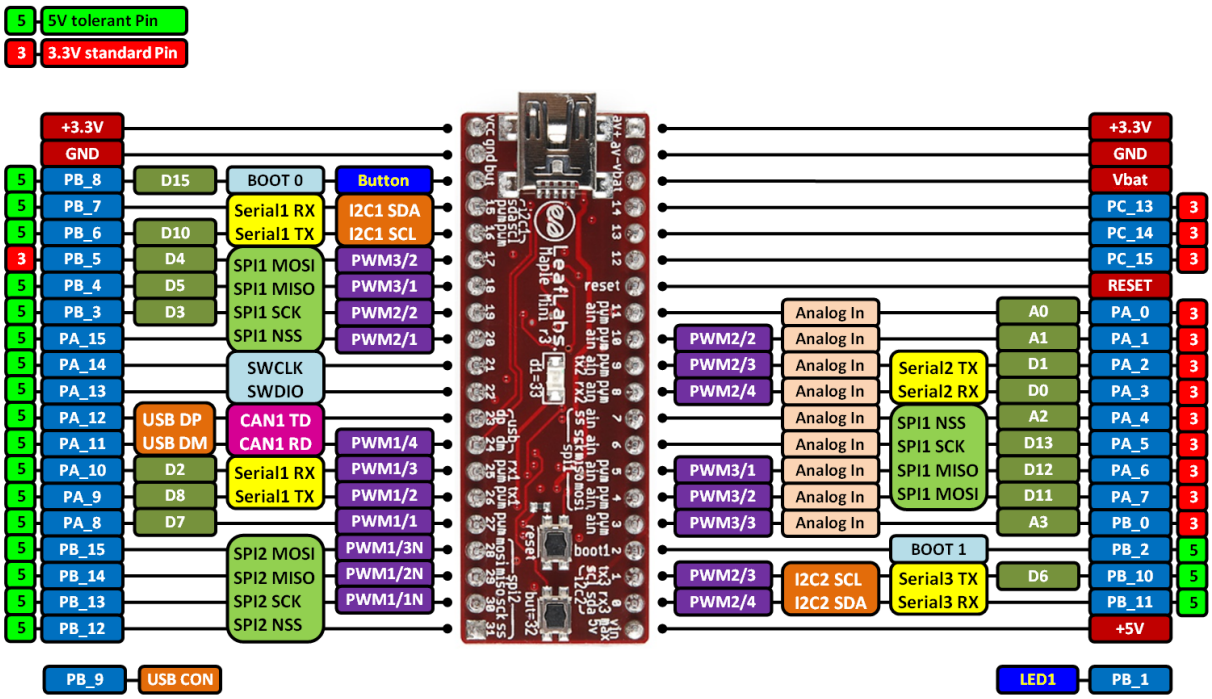


Abbildung 2.13:Pinbelegung des Maple Mini Boards [Huiskamp, 2018]

2.3 Der *Raspberry Pi*

Ein Ziel dieser Masterarbeit ist der Einsatz des Python Programmes auf dem *Raspberry Pi*. Da in Zukunft ein ganzer Sensorblock gefertigt werden soll und die Kosten für die verwendete Hardware eine wichtige Rolle spielt, wurde überlegt, welche Kosteneinsparungen möglich sind. Daher bietet sich ein Computer an, welcher in der Anschaffung und im Unterhalt sehr preiswert ist. Deshalb ist die Wahl auf den *Raspberry Pi* gefallen. Ein Rechner dieser Art kostet um die 30 € und ist ein sehr energieeffizienter Rechner. Der *Raspberry Pi* ist ein Einplatinencomputer der britischen Raspberry Pi Foundation. Mit den Abmessungen von 93 mm x 63,5 mm x 20 mm ist dieser sehr kompakt und mit einer Stromaufnahme von ca. 1 bis 2 Watt (ohne Monitor) sehr stromsparend. Trotzdem kann fast die Funktionalität eines normalen Personalcomputers erreicht werden. Auf dem *Raspberry Pi* können verschiedene Betriebssysteme installiert werden. Es ist mit dem Betriebssystem Raspbian gearbeitet worden. Ein großer Vorteil bei diesem Betriebssystem ist, neben der freien Verfügbarkeit, die Vorinstallation von Python. So konnte das Python Programm fast unverändert verwendet werden, da zuerst auf dem PC programmiert wurde. [Weigend, 2015]

Abbildung 2.14: Raspberry Pi mit und ohne Gehäuse

2.4 Kostenfaktor

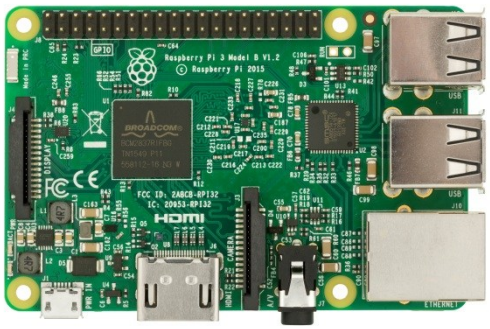
Eine Motivation für die Realisierung war eine mögliche Herstellung von Sensorblöcken mit geringen Materialkosten. Die Kosten für einen handelsüblichen Beschleunigungssensor für

ingenieurgeodätische Anwendungen liegt bei ca. 5000 €. Als Beispiel sei hier der in der Hochschule Neubrandenburg vorhandene *SBG IG-500E* genannt.

Abbildung 2.15: professioneller Beschleunigungssensor SBG IG-500E

Für den Sensorblock konnten die Materialien im einschlägigen Fachhandel erworben werden und bestehen aus preiswerter Massenware, siehe Tabelle 2.1. Die Untersuchungen werden zeigen ob der neu entwickelte Sensorblock eine ähnliche Genauigkeit wie der professionelle Sensor und damit im selben Anwendungsbereich arbeiten kann. Die Materialkosten für die Herstellung des Sensorblockes betragen eine Summe von ca. 60 €. Durch den geringeren finanziellen Aufwand können mehr Sensorblöcke pro Messobjekt angebracht werden. Dadurch entsteht eine hohe Redundanz der Messelemente und der Ausfall eines Sensors kann kompensiert werden. Dieser Sensorblock ist in der Lage Objektbewegungen mit Frequenzen von mehr als einem Hertz zu erfassen. Damit kann der Einsatz des ingenieurgeodätischen Monitorings an Bauwerken erweitert werden. Die Verwendung dieses Sensorblocks ermöglicht eine erhebliche Kosteneinsparung, da die klassischen geodätischen Sensoren, wie z.B. Tachymeter oder GNSS, in der Anschaffung sehr teuer sind. Außerdem kann von diesen preiswerten Sensoren eine höhere Anzahl eingesetzt werden, um das Messsystem redundanter zu gestalten. So ist auch trotz Kosteneinsparung ein Genauigkeitsgewinn zu erwarten.

Tabelle 2.1: Kostenzusammenstellung

Bild	Beschreibung	Kosten
	Der Beschleunigungssensor ist als mikro-elektro-mechanisches System (MEMS) ausgeführt und misst die Trägheitsbeschleunigungen, welche auf das zu überwachende Objekt wirken.	40 €
	Auf dem *Arduino*-Mikrocontroller läuft das Programm zur Sensordatengewinnung. Der Mikrocontroller kommuniziert mit dem AD-Wandler und gibt die erfassten Daten über den universalen seriellen Bus (USB) weiter.	10 €
	Die über USB übertragenden Daten werden auf einem Einplatinen-computer gespeichert und zur weiteren Datenübertragung über das Internet aufbereitet und versendet. Es wird ein *Raspberry Pi* verwendet, welcher sich durch einen niedrigen Energieverbrauch, hohe Zuverlässigkeit und einen niedrigen Anschaffungspreis auszeichnet.	45 €

3 Kommunikation mit den Sensoren

Als grundlegender Schritt bei der Entwicklung dieses Sensors ist die Kommunikation mit dem Sensor selbst zu nennen. Um überhaupt etwas zur Leistungsfähigkeit, Datenraten, Kalibrierung und Anwendungen aussagen zu können, müssen zuerst Daten gesammelt werden. Daher ist es wichtig den Sensor anzusprechen und die idealste Schnittstelle zur Kommunikation mit dem Sensor auszuwählen. In diesem Kapitel sollen die verwendeten Schnittstellen vorgestellt werden und ein Überblick über die erste Kommunikation mit den Sensoren gegeben werden.

3.1 Schnittstellen

Schnittstellen verbinden zwei Geräte oder Systeme. Diese Systeme sind über Schnittstellenleitungen verbunden. Charakterisiert sind die Schnittstellen durch ihre technische Beschaffenheit, die Art der zu übertragenden Signale und die Bedeutung der zu

übertragenden Signale. Viele Schnittstellen sind standardisiert, was die Nutzung erleichtert. In der Regel sind Schnittstellen Punkt zu Punkt Verbindungen.

3.1.1 RS232

Die RS232-Schnittstelle, auch V.24 Schnittstelle, ist eine der am meisten genutzten Schnittstellen im Elektronikbereich. Dies liegt zu einen daran, dass diese Schnittstelle schon in den 1960er Jahren entwickelt wurde und zum anderen daran, dass diese Schnittstelle eine sehr einfache Datenübertragung gewährleistet. Die Schnittstelle arbeitet seriell, das heißt, die Daten werden in Form einzelner Bits nacheinander übertragen. Die bekannteste Steckerform der RS232 Schnittstelle ist der klassische D-Sub Stecker und Buchse.

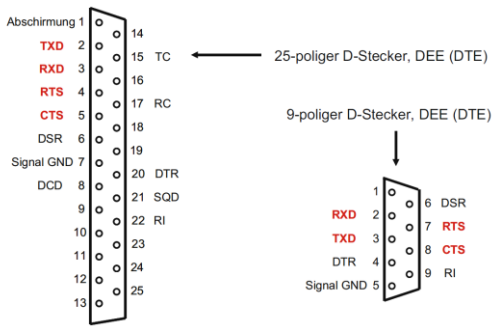


Abbildung 3.1: D-Sub Stecker einer RS232 Schnittstelle [Hering, et al., 2017]

Das Übertragungsprinzip der RS232 Schnittstelle beruht auf dem *Universal Asynchronous Receiver Transmitter (UART)*. Dies ist eine Schaltung, bei der die Daten als serieller Datenstrom übertragen werden. Der Datenstrom beginnt mit einem Start-Bit, dann folgen fünf bis neun Datenbits, danach kann ein optionales Parity-Bit zum Erkennen von Übertragungsfehlern folgen und zum Abschluss der Übertragung wird ein Stopp-Bit übertragen. Es wird kein eigenes Taktsignal benötigt. Die Synchronisation beruht auf der Länge der Übertragung, gemessen von Start-Bit bis zum Stopp-Bit. [Hering, et al., 2017]

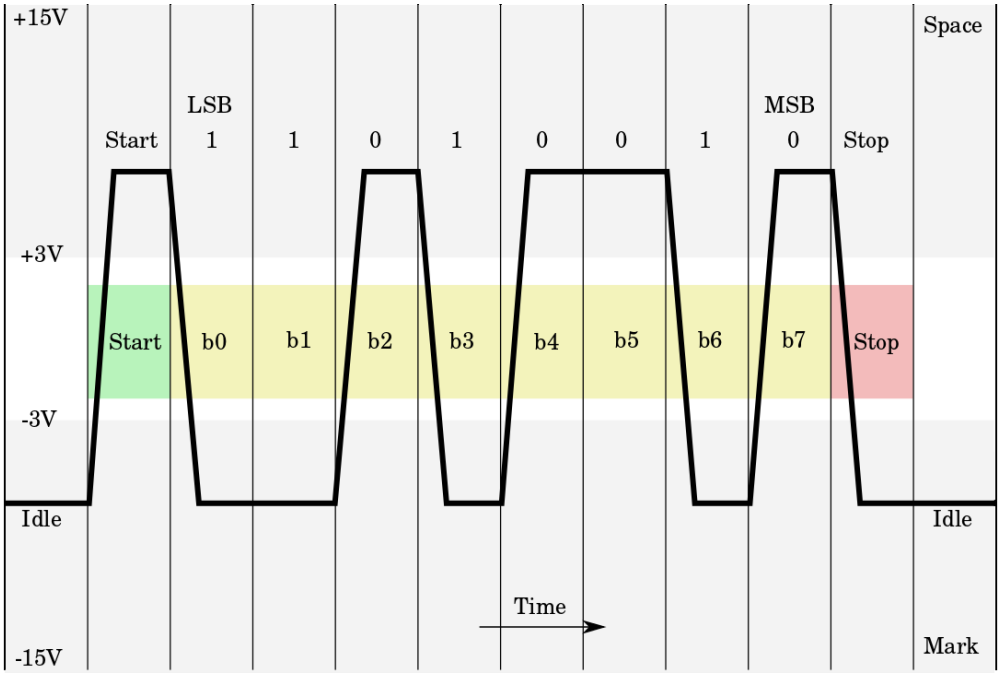


Abbildung 3.2: Spannungsverlauf einer Datenübertragung der RS232 Schnittstelle [Hering, et al., 2017]

Die Spannungen für ein jeweiliges Bit sind in den Grenzen von -15 V bis + 15 V definiert. Im Bereich von -3 V bis + 3 V ist dieser Bereich ungültig. Die Übertragungsgeschwindigkeiten liegen von 50 Bit/s bis 500 000 Bit/s. Ein Zusammenhang besteht zwischen der Übertragungsgeschwindigkeit und der Leitungslänge. Die maximale Leitungslänge L lässt sich nach Formel (1) berechnen.

$$L = \frac{1\,000\,000}{Bits\ pro\ Sekunde} \tag{1}$$

Bei einer Übertragungsgeschwindigkeit von 115200 Bit/s ergibt sich eine Leitungslänge von ca. einem Meter. Diese Restriktion ist bei der Verwendung der RS232 unbedingt zu beachten, da bei einer hohen Datenrate der Leitungsweg klein gehalten werden muss, um eine korrekte Datenübertragung zu gewährleiten.

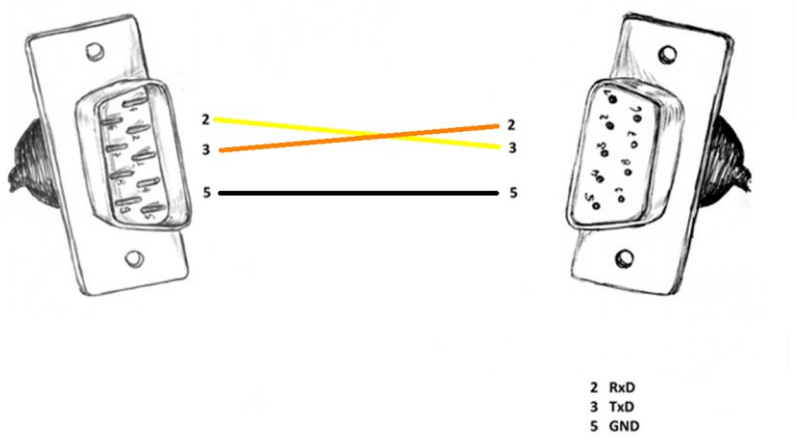


Abbildung 3.3: 3 Kabel-Lösung zur Datenübertragung

Durch die Belegung der Receive Data (RxD), der Transmit Data (Txd) und der Ground (GND) Pins gelingt die einfachste Möglichkeit einer Datenübertragung.

3.1.2 USB

Der Universal Serial Bus, abgekürzt USB, ist zur Harmonisierung der Schnittstellenvielfalt entwickelt worden. USB standardisierte die vieler Computerschnittstellen, wie z.B. Eingabegeräte, Drucker oder Kameras. [Hering, et al., 2017]

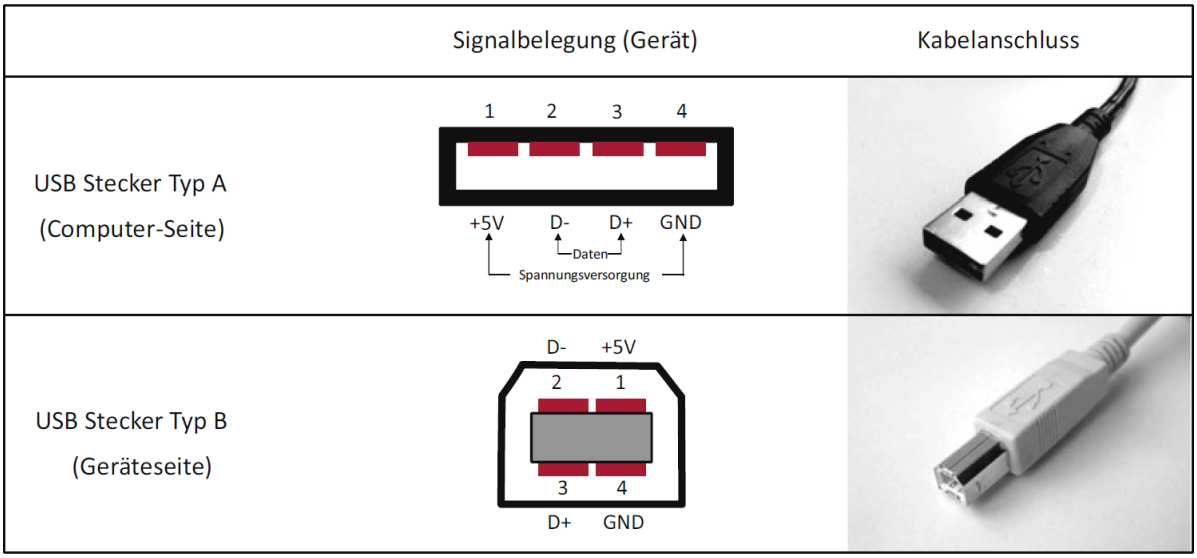


Abbildung 3.4: USB Stecker [Hering, et al., 2017]

Die Übertragungsraten des USB liegen bis zu 10 Gbit/s. Da die Daten aber nicht einzeln sondern in Paketen übertragen werden, kann es zu Verzögerungen bei der Datenübertragung kommen. Daher ist es notwendig, die Daten vor der Übertragung durch USB mit einem Zeitstempel zu versehen, um die korrekte Zeitinformation zu erhalten. Dies

kann durch die Erzeugung eines Zeitstempels während des Datenempfangs vom Sensor durch den Mikrocontroller erfolgen.

Tabelle 3.1: Geschwindigkeiten von USB

USB Version	USB 1.0	USB 1.1	USB 2.0	USB 3.0	USB 3.1
USB-Bezeichnung	Low-Speed	Full-Speed	High-Speed	Super-Speed	Super-Speed-Plus
Bit-Datenrate	1,5 Mbit/s	12 Mbit/s	480 Mbit/s	5 Gbit/s	10 Gbit/s
Max. Datenrate	188 kbyte/s	1,5 Mbyte/s	60 Mbyte/s	600 Mbyte/s	1,2 Gbyte/s
Netto-Datenrate	≈ 150 kbyte/s	≈ 1 Mbyte/s	≈ 60 Mbyte/s	≈ 450 Mbyte/s	≈ 800 Mbyte/s
max. Leitungslänge	5 m	5 m	5 m	3 m	1 m

3.1.3 I²C

I²C oder Inter-Intergrated Circiut ist ein serieller Datenbus. Im Jahre 1982 ist dieser von Philips zur seriellen Kommunikation von digitalen Bauelementen innerhalb eines elektronischen Systems entwickelt worden. Es ist eine taktsynchrone Datenschnittstelle, da hier ein Taktsignal für die zeitlichen Abläufe der Datenübertragung eingesetzt wird.

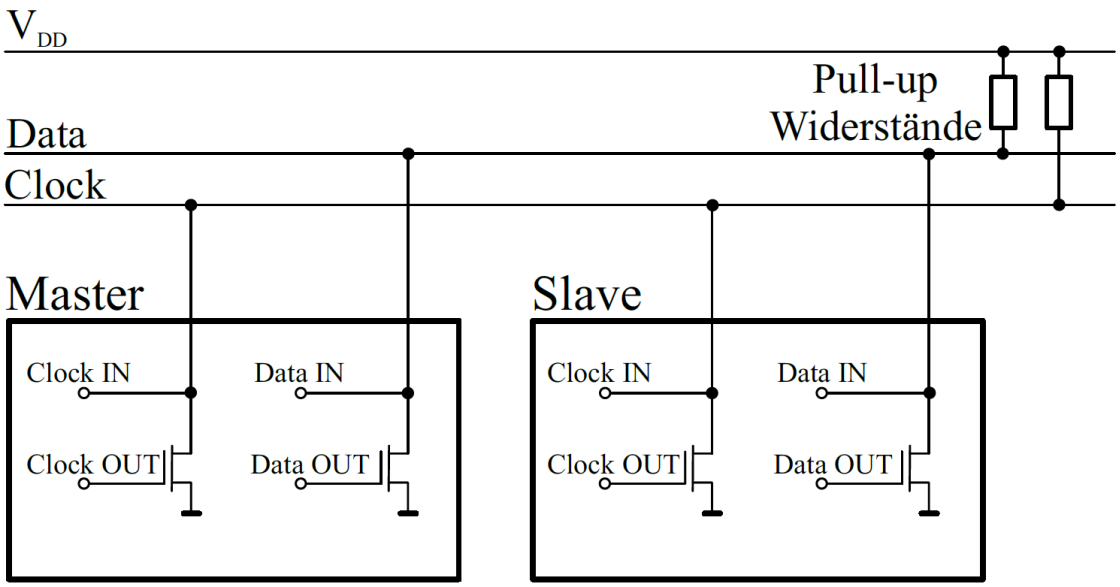


Abbildung 3.5: Beispiel einer I²C Bus Ansteuerung [Lerch, 2016]

Wie in Abbildung 3.5 zu sehen ist, arbeitet der I²C Bus als Master –Slave Bus. Der Datentransfer wird über den Master begonnen. Der Slave muss hier auf die Aufforderung des Masters warten, um seine Daten in die gemeinsame Datenleitung zu senden.

Wesentliche Eigenschaften des I²C Busses sind nach [Hering, et al., 2017]:

- 2-Draht Busleitung und Bezugspotential
- Jeder Teilnehmer hat eine eindeutige Adresse
- Die Kommunikation erfolgt nach dem Master-Slave Prinzip
- Die Baud-Rate wird variabel von der Takt-Leitung gesteuert

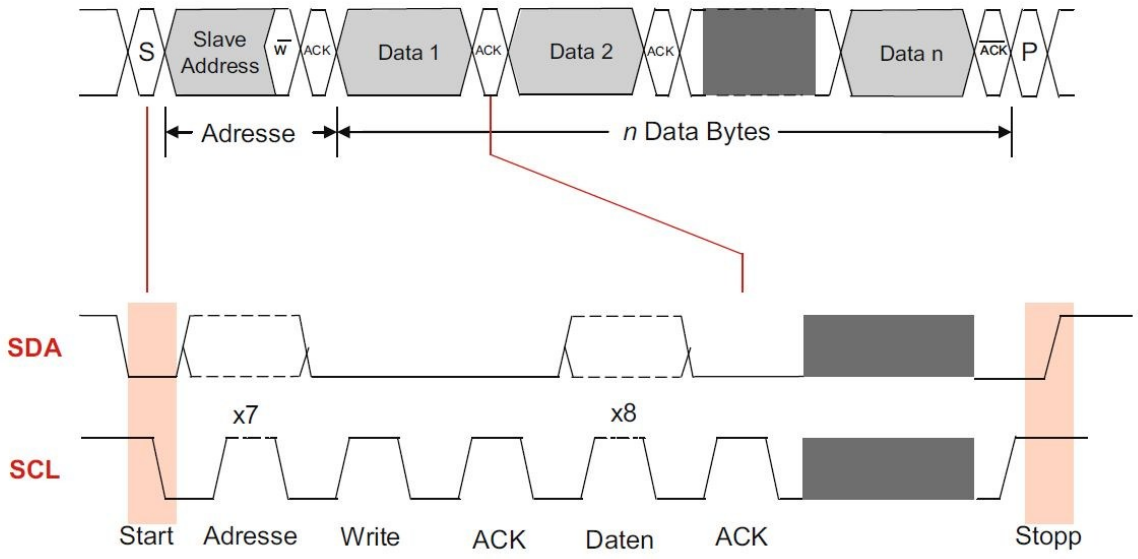


Abbildung 3.6: Ablauf der Datenübertragung auf dem I²C Bus [Hering, et al., 2017]

Der *ADXL355* Beschleunigungssensor und der *ADS1115* Analog-Digital Wandler verfügen beide über eine I²C Schnittstelle. Diese ist bei dieser Masterarbeit verwendet worden, um die Daten der Beschleunigungsmesser auszulesen.

3.2 Auslesen der Daten

Zuerst sollen die *Arduino*-Mikrocontroller auf ihre Funktion geprüft werden. Nach dem Verbinden mit dem Computer über die USB-Buchse wird die *Arduino* Software installiert. Diese erlaubt das Hochladen von Programmen (Sketches) auf den Mikrocontroller. Nachdem die Kommunikation mit dem *Arduino*-Mikrocontroller hergestellt ist, kann nun mit dem Verbinden der einzelnen Bauteile begonnen werden. Hier ist eine Steckplatine (Breadboard) verwendet worden. So können die Schaltungen durch einfaches Umstecken angepasst werden.

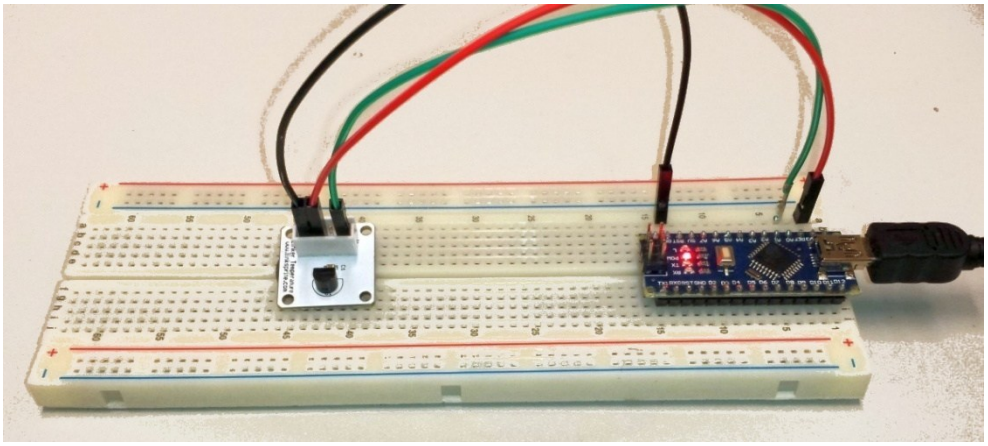


Abbildung 3.7: Versuchsaufbau mit Linker Temperature und Arduino Nano

Bei dem Temperatursensor müssen neben der Versorgungsspannung und Masseverbindung der analoge Pin zur Datenausgabe angeschlossen werden, siehe Abbildung 3.7. So sollte am analogen Eingang des Mikrocontrollers der analoge Spannungswert des Temperatursensors anliegen.

Für die Herstellung der Kommunikation zum Beschleunigungssensor wird ebenfalls auf diese Methode zurückgegriffen.

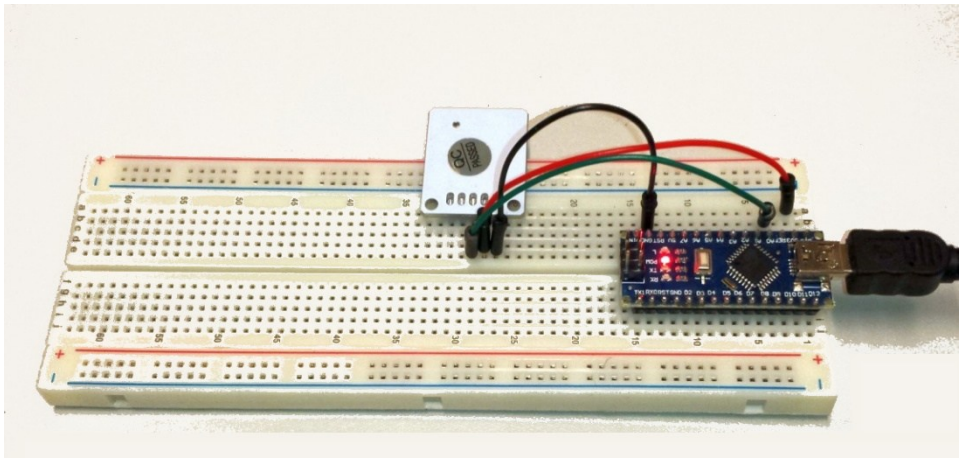


Abbildung 3.8: Versuchsaufbau mit Linker 3-Axis-Accelerometer und Arduino Nano

Hier müssen neben der Versorgungsspannung und Masseverbindung alle drei analogen Pins zur Datenausgabe angeschlossen werden, siehe Abbildung 3.8. An den drei verschiedenen analogen Eingängen des Mikrocontrollers liegen die analogen Spannungswerte der jeweiligen Achsen des Beschleunigungssensors an.

3.2.1 Temperatursensor *Linker Temperature*

Für das Auslesen ist ein für den Mikrocontroller *Arduino* erstelltes Programm verwendet worden. Dieses wird in der *Arduino*-Diktion Sketch genannt. Durch den AD-Wandler wird eine wertdiskretisierte Zahl im Bereich von 0 bis 1023 als Spannung abgerufen. Es wird hierbei das Problem der zu geringen Auflösung des AD-Wandlers deutlich. 10 Bit ergeben ca. 1000 Werte, dies ergibt bei einer Spannung von 0-5V eine minimale Spannungsdifferenz von 5mV. So liegt die Schrittweite der Temperatur bei 0,5 °C, was für die Auflösung der Temperatur in den meisten Fällen ausreichend ist.

linker_temperature2 §

```
void setup()
{
  Serial.begin(9600);  //Start the serial connection with the computer
                       //to view the result open the serial monitor
}

void loop()                     // run over and over again

 //getting the voltage reading from the temperature sensor
 //analogReadResolution(12);
 int reading = analogRead(sensorPin);

 // converting that reading to voltage, for 3.3v arduino use 3.3
  float voltage = reading * 3.3;
//  float voltage = reading ;
voltage /= 4096.0;

 // print out the voltage
 //Serial.print(voltage); Serial.println(" volts");

 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit 500 mV offset
                                               //to degrees ((volatge - 500mV) times 100)
 Serial.print(temperatureC); Serial.println(" degrees C");


 delay(2000);                                  //waiting a second
}
```

Abbildung 3.9: Sketch des Arduino Programmes zum Auslesen des Temperatursensors

3.2.2 Beschleunigungssensor *Linker 3-Axis-Accelerometer*

Hier wird ebenfalls ein für den Mikrocontroller *Arduino* zugeschnittenen Programm verwendet. Es müssen alle drei Achsen durch drei verschiedene Variablen angesteuert werden. Sonst ist dieses Programm(Sketch) ähnlich zu dem Sketch des Temperatursensors aufgebaut. Ein besonderes Problem ist beim Auslesen des Beschleunigungssensors aufgetreten. Es kann durch die Auflösung des AD-Wandlers von 10 Bit (1023 Schritte) nicht die volle Auflösung des Sensors genutzt werden. Es wird nur eine Auflösung von 0.25 m/s² erreicht. Die Auflösung des Sensors liegt bei 0,04 m/s². Der Ansatz und die Lösung dieses Problems wird in Kapitel 3.3 erläutert.

3.2.3 Beschleunigungssensor *ADXL354*

Die Funktionsweise des Auslesens des *ADXL354* gestaltet sich sehr äquivalent zum ACCEL-Board. Da der *ADXL354* ist ein analoger Sensor ist, kann wieder direkt mit dem AD-Wandler des Mikrocontroller *Arduino* gearbeitet werden. Auch hier wird nur eine Auflösung von 0.25 m/s² erreicht. Da die Auflösung des Sensors aber bei 0,002 m/s² (Abtastrate 100 Hz) liegt, muss auf jeden Fall auf einen separaten AD-Wandler zurückgegriffen werden. Der Ansatz und die Lösung dieses Problems wird im Kapitel 3.3 erläutert.

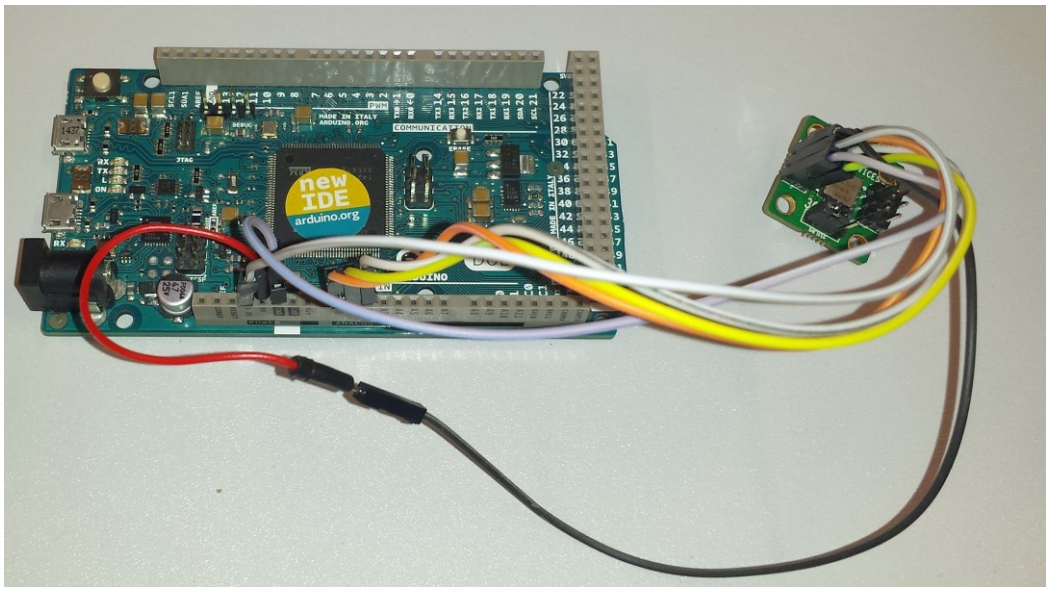


Abbildung 3.10: ADXL354 Beschleunigungssensor und Arduino Due

Der *ADXL354* Beschleunigungssensor kann auch in Kombination mit dem Analog-Digital-Wandler *ADS1115* eingesetzt werden. Die Vorteile sind im Kapitel 3.3.1 beschrieben.

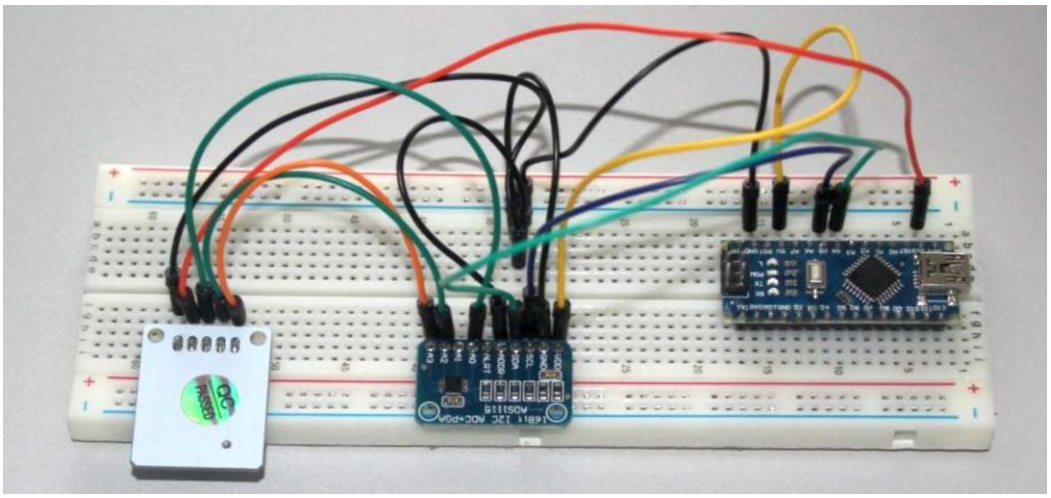


Abbildung 3.11: ADXL354 Sensor mit ADS1115 und Arduino Nano

3.2.4 Beschleunigungssensor *ADXL355*

Im Gegensatz zu den bisher vorgestellten analogen Sensoren arbeitet der Beschleunigungssensor *ADXL355* digital. Für die Verkabelung ist dies ein Vorteil, da hier neben der Spannungsversorgung und Masse nur noch *serial clock (SCL)*, *serial data (SDA)* und ein Adresskabel *(ADDR)* verbunden werden müssen. Nun muss der I²C Bus direkt vom *Arduino* Sketch angesteuert werden. Daher ist eine grundlegende Änderung des Programmes nötig. Ein großer Vorteil hier ist, dass der im *ADXL355* eingebaute Temperatursensor auch direkt mit ausgelesen werden kann.

3.3 Probleme bei der Auflösung der AD-Wandlung

Eine Erhöhung der Auflösung des AD-Wandlers ist für die Nutzung des Beschleunigungssensors im geodätischen Bereich erforderlich. Dies kann über verschiedene Ansätze verfolgt werden.

3.3.1 Mikrocontroller Arduino *Due* und AD-Wandler *ADS1115*

Es kann ein anderer Mikrocontroller, wie z.B. der Mikrocontroller *Arduino Due* verwendet werden, welcher über eine Auflösung von 12 Bit verfügt. Durch den Einsatz des *Arduino Due* wird die Auflösung um den Faktor vier erhöht. Das ist aber in meisten Fällen noch nicht ausreichend, daher muss eine Lösung mit höherer Auflösung gefunden werden. Durch den Einsatz eines hochwertigeren AD-Wandler wird die Auflösung um den Faktor 64 erhöht.

Genutzt wurde in dieser Masterarbeit der AD-Wandler *ADS1115* mit 16 Bit Auflösung. Bei der Nutzung dieses Wandlers ist eine Datenrate von maximal einem Kilohertz erreichbar, was völlig ausreichend ist.

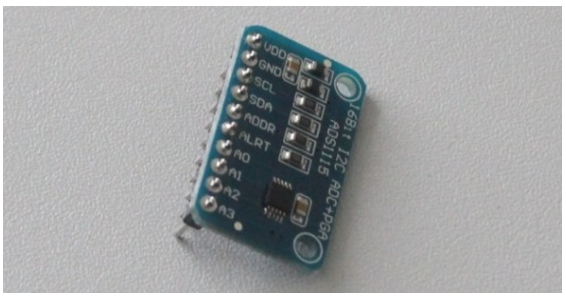


Abbildung 3.12: 16-Bit AD-Wandler ADS1115

3.3.2 Änderung des Spannungsbereiches des AD-Wandlers

Ebenfalls kann der Spannungsbereich des AD-Wandlers verändert werden. Der *ADS1115* hat einen Arbeitsbereich von 0 bis 6 Volt. Dieser kann auf 0 bis 2 Volt angepasst werden, da die verwendeten Sensoren nur in diesem Bereich arbeiten. Dies ergibt eine Verbesserung der Auflösung um den Faktor 3. Nachfolgend ist dieses in Tabelle 3.2 für den Temperatursensor aufgeführt worden.

Tabelle 3.2: AD-Wandlung und Temperaturauflösung

AD-Wandler	*Arduino Nano*	*Arduino Due*	*ADS1115*
Auflösung [Bit]	10	12	16
Auflösung [°C]	0,5	0,12	0,01

Bei dem Temperatursensor ist eine Erhöhung der Auflösung des AD-Wandlers nicht wichtig gewesen, da dieser nur eine Sensorauflösung von 0,1°C hat und die Temperatur meist mit 1°C ausreichend ist. Der Beschleunigungssensor ist nur mit der höchsten zu erreichenden Auflösung nutzbar. Dieser Sensor hat eine eigene Auflösung von 2 mm/s². Anhand der Tabelle 3.3 wird deutlich wie die Auflösung des AD-Wandlers die Einsatzfähigkeit des Beschleunigungssensors bestimmt.

Tabelle 3.3: AD-Wandlung und Beschleunigungsauflösung

AD-Wandler	*Arduino Nano*	*Arduino Due*	*ADS1115*
Auflösung [Bit]	10	12	16
Auflösung [mm/s²]	200	50	3

3.3.3 AD-Wandlung des Beschleunigungssensors *ADXL355*

Ein großer Vorteil des *ADXL355* Beschleunigungssensors ist der interne AD-Wandler. Dieser hat eine Auflösung von 20 Bit. Wie in Tabelle 3.4 zu erkennen ist, spielt eine Senkung der Auflösung der AD-Wandlung bei dem *ADXL355* keine Rolle mehr. Hier liefert der AD-Wandler eine zehnmal so hohe Auflösung wie gefordert.

Tabelle 3.4: AD-Wandlung und Beschleunigungsauflösung inklusive ADXL355

AD-Wandler	*Arduino Nano*	*Arduino Due*	*ADS1115*	*ADXL355*
Auflösung [Bit]	10	12	16	20
Auflösung [mm/s²]	200	50	3	0,2

Als Fazit ergibt sich, dass entweder der Beschleunigungssensor *ADS1115* für mittlere oder der *ADXL355* für höchste Genauigkeitsanforderungen verwendet werden kann. Der Vorteil des AD-Wandlers *ADS1115* ergibt sich daraus, dass er unabhängig vom eingesetzten Sensor arbeitet und somit auch für andere Sensorkombinationen einsetzbar ist.

4 Programmierung und Bau der Sensoren

4.1 Programmierung

Ein grundlegender Bestandteil der Entwicklung eines neuen Sensors ist die Programmierung. Da die nötige Hardware ausgewählt ist, muss diese nun angesteuert werden können. Dazu sind Softwarelösungen nötig, um einen einfachen, störungsfreien und zuverlässigen Betrieb des Sensors zu gewährleisten.

4.1.1 Wahl der Programmiersprachen

Bevor mit der Entwicklung des Programmes begonnen werden kann, stellt sich zuerst die Frage, welche Programmiersprache für diese Anwendung die geeignetste ist. Da ein Ziel dieser Masterarbeit die Entwicklung eines preiswerten Sensorblocks für geodätische Monitoring-Aufgaben ist, sollten auch preiswerte Auslesegeräte verwendet werden. Ein Beispiel hierfür ist der *Raspberry Pi*. Das Pi im Namen dieses preiswerten Einplatinencomputer steht für die Programmiersprache Python.

Der Mikrocontroller, welcher den Sensor ausliest und die Daten verarbeitet und weiterleitet, ist auf *Arduino*-Basis aufgebaut. Hier sollen einfache *Arduino* Programme, die Sketche genannt werden, zur Anwendung kommen. Die *Arduino* Plattform arbeitet mit einer auf C und C++ basierenden Programmiersprache.

Mit dem Mikrocontroller *Arduino* wird das Messsystem auch unabhängig vom jeweiligen Rechnersystem zur Weiterverarbeitung. Es wird einerseits angestrebt, den *Raspberry Pi* als weiterführenden Rechner zu verwenden. Es gibt hier aber auch eine Variante, bei der ein auf dem Windows-Betriebssystem basierender Rechner die Weiterverarbeitung übernimmt. Eine dritte Variante nutzt keinen Rechner zur Weiterverarbeitung, sondern sendet die Messdaten gleich in ein datenbankorientiertes Monitoringsystem. Daher soll auf die Programmierung des Auslesens des Sensors besonders eingegangen werden. Da vom preiswerten Komplettsystem mehrere Prototypen entwickelt werden sollen, ist auch die Programmierung des *Raspberry Pi* mit Python ein Bestandteil dieser Masterarbeit.

4.1.2 Nutzung der Programmiersprache Python am Beispiel des Raspberry Pi

Die Programmiersprache Python ist eine leicht zu erlernende Programmiersprache und so gestaltet, dass sie besonders gut lesbar ist. Sie ist eine objektorientierte Programmiersprache, welche aber auch Programmierparadigmen, wie funktionale und imperative Programmierung, erlaubt. Somit ist sie eine universelle Programmiersprache mit vielen Einsatzmöglichkeiten. Sie wird sowohl in Wissenschaft und Technik, z.B. beim Deutschen Zentrum für Luft- und Raumfahrt, als auch für visuell-kreative Projekte, z.B. bei Disney zur Entwicklung von Computerspielen, verwendet.

Die Programmiersprache Python hat gegenüber älteren Programmiersprachen drei Vorteile:

- Leichte Erlernbarkeit und eine niedrige Eingangsschwelle, sodass ohne theoretische Vorkenntnisse die ersten Programme geschrieben werden können.
- Python Programme sind kurz und verständlich. Der Programmtext muss für jedermann gut lesbar sein, damit er auf einfache Weise verändert, erweitert und verbessert werden kann. Der amerikanische Informatiker Donald Knuth hat vor drei Jahrzehnten vorgeschlagen, Programme als Literatur zu betrachten, so wie Romane und Theaterstücke. Nicht nur die Korrektheit und Effizienz, auch die Lesbarkeit ist ein Qualitätsmerkmal.
- Programme können mit der Programmiersprache Python nachweislich in kürzerer Zeit entwickelt werden als mit anderen Programmiersprachen. [Weigend, 2015]

4.1.3 Nutzung der Programmiersprachen C und C++ am Beispiel der Arduinoplattform

In dieser Masterarbeit werden *Arduino* Mikrocontroller verwendet. *Arduino* ist eine aus Software und Hardware bestehende Plattform. Die Komponenten sind im Sinne von Open Source quelloffen. So kann neben der Hardware, also den *Arduino*-Mikrocontrollern, auch die *Arduino* Entwicklungsumgebung, die *Arduino* IDE, genutzt werden. Diese Umgebung arbeitet mit einer Programmiersprache, die auf C und C++ basiert. Die Programme selbst werden Sketches genannt und die Bearbeitung innerhalb der *Arduino* IDE ist einfach. Hierbei müssen natürlich Grundkenntnisse der Programmiersprachen C und C++ vorhanden sein.

Die Programmiersprache C/C++ hat sich im Laufe der Jahre zu einer der bedeutendsten Universalsprechen im Bereich der Systemprogrammierung und auf den Gebieten der technischen und wissenschaftlichen Entwicklung entwickelt. Die Programmiersprache C unterstützt system- und hardwarenahe Probleme und die Programmiersprache C++ strebt eine bessere Unterstützung der Anwendungsprogrammierung mit der integrierten Objektorientierung an. [Küveler, et al., 2017]

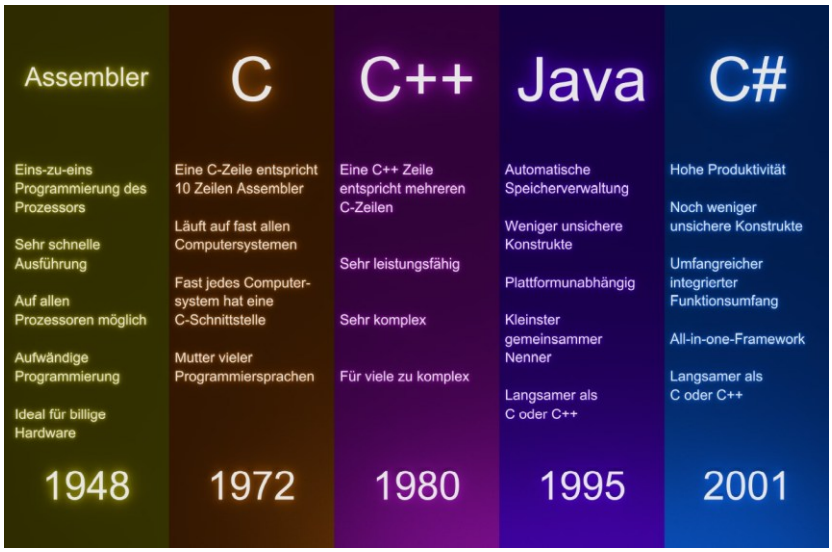


Abbildung 4.1: Übersicht der Entwicklung der Programmiersprache C [Ingenieurbüro Heimann, 2018]

Die Programmiersprache C ist im Zusammenhang mit dem Betriebssystem UNIX erfolgt. Seit 1973 ist das Betriebssystem UNIX fast vollständig in der Programmiersprache C geschrieben worden. C ist eine Programmiersprache der 3. Generation und die wichtigste Programmiersprache im Ingenieurbereich. Die wesentlichen Merkmale der Sprache sind ein breites Anwendungsspektrum, knappe Befehle und ein sehr klares Sprachkonzept.

C++ ist als Erweiterung zur Programmierspreche C im Jahre 1979 von Bjarne Stroustrup entwickelt worden. Hier sind zusätzliche objektorientierte oder prozedurale Paradigmen implementiert worden. C ist vollständig in C++ enthalten, daher funktioniert (fast) alles von C auch in C++. Die knappen Befehle und das klare Sprachkonzept erfordern vom Programmierer Grundkenntnisse über Aufbau und Arbeitsweise von Computern. [Küveler, et al., 2017]

4.1.4 Anforderungen an die Auslese- und Verarbeitungsprogramme

Nach der Wahl der jeweils geeigneten Programmiersprache sind die Anforderungen an die zu entwickelnden Programme festzulegen. Hier sind die wichtigsten Kriterien festzuhalten und zu prüfen, ob diese umsetzbar sind.

Das Ausleseprogramm soll folgende Anforderungen erfüllen:

- Möglichkeit zur Kommunikation mit der I²C Schnittstelle, d.h. Import der Bibliothek zur Kommunikation mit der I²C Schnittstelle
- Auslesen der Datenregister des Sensors und stabile Kommunikation
- Umwandlung der Sensordaten, welche in Binärform empfangen werden, in eine Zahl im ASCII Code
- Möglichkeit zur Kommunikation mit der seriellen Schnittstelle, d.h. Import der Bibliothek zur Kommunikation mit der seriellen Schnittstelle
- Öffnen des seriellen Portes für das Ausgeben der Daten
- Importieren der Bibliothek zur Zeitbestimmung und Anbringen eines Zeitstempels an die Datenreihe zur zeitlichen Festlegung des Datenursprungs

Diese Anforderungen sind leicht in ein *Arduino* Sketch zu implementieren. Der geschriebene Sketch wird dann auf den Mikrocontroller übertragen und dauerhalt ausgeführt. Auch nach jeder Trennung des Mikrocontrollers vom Stromnetz wird nach entsprechendem Einschalten dieses Programm wieder ausgeführt.

Während das Programm des -Mikrocontrollers *Arduino* die Datenübertragung sicherstellt, wird das Verarbeitungsprogramm auf dem *Raspberry Pi* oder auf einem Windows basierendem PC zur Datenspeicherung und Weiterverarbeitung benötigt. Im Folgenden wird aufgelistet, welche Anforderungen an das Verarbeitungsprogramm zu stellen sind:

- Möglichkeit zur Kommunikation mit der seriellen Schnittstelle, d.h. Import der Bibliothek zur Kommunikation mit der seriellen Schnittstelle
- Importieren der Bibliothek zur Zeitbestimmung
- Öffnen des seriellen Portes für das Auslesen
- Programmende bei fehlender Möglichkeit des Öffnens des seriellen Portes
- Abfrage der zeitlichen Länge der Programmaktivität
- Anlegen einer Datei mit dem Sensornamen und aktuellem Datum plus Uhrzeit
- Abfrage der zeitlichen Länge einer Datei
- Umwandeln der Datentyps Bit in String zum Schreiben in die Datei
- Aufteilen der Gesamtdatenpakets der Sensordaten in Einzelpakete zur Speicherung
- Anbringen eines Zeitstempels an die Datenreihe zur zeitlichen Festlegung des Datenursprungs
- Speicherung in die festgelegte Datei, solange die zeitliche Länge der Datei nicht überschritten ist
- Schließen und Speicherung der Einzeldatei
- Schließen des seriellen Portes

Nach Festlegung dieser Anforderungen muss nun überprüft werden, in welcher Weise diese Aufgaben abgearbeitet werden können und welche Zwischenschritte notwendig sind.

4.1.5 Entwicklung des Programmschemas

Nach der Festlegung der Anforderungen an das zu entwickelnde Programm wird das Programmschema erstellt. Hierfür bietet sich ein Programmablaufplan an, in dem die Folge der Operationen angegeben ist, die zur Lösung der Aufgabe erforderlich sind. Die Symbole für Programmablaufpläne sind in der DIN 66001 genormt. Dort werden auch Symbole für Datenflusspläne definiert. Programmablaufpläne werden oft unabhängig von Computerprogrammen auch zur Darstellung von Prozessen und Tätigkeiten eingesetzt.

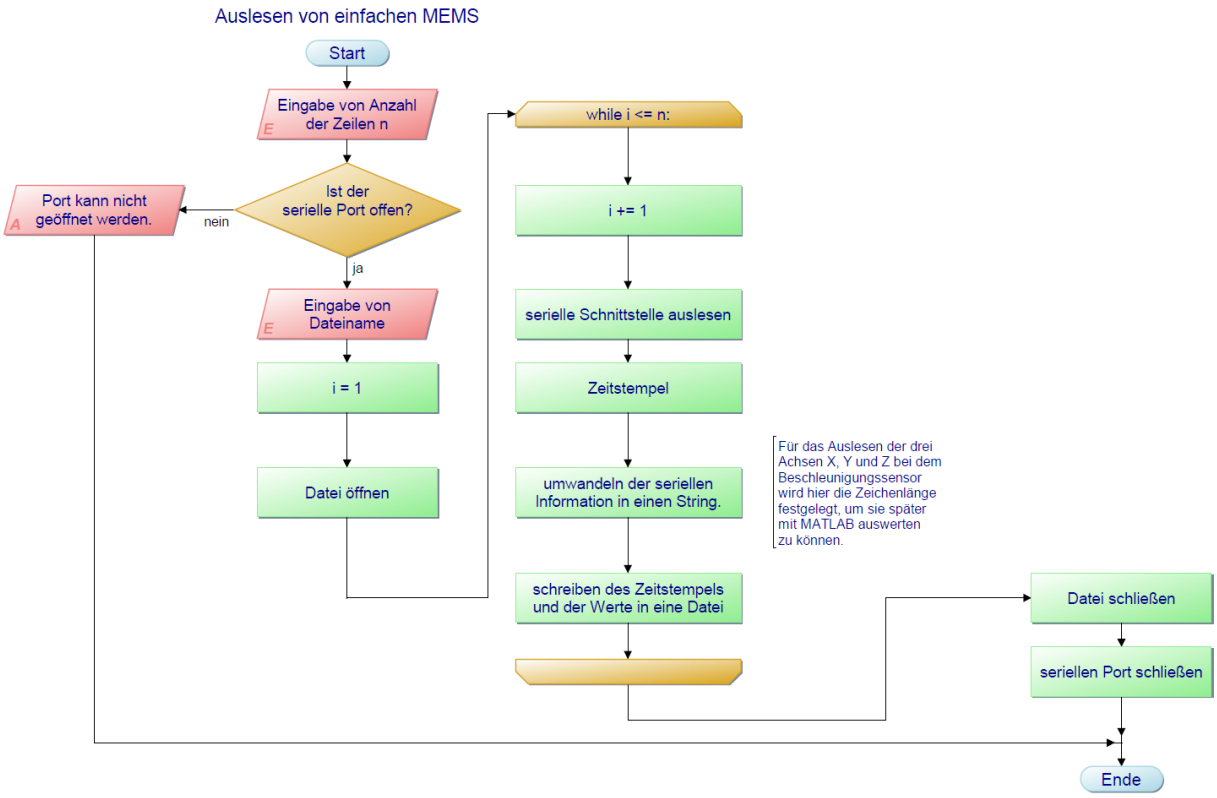


Abbildung 4.2: vorläufiger Programmablaufplan zum Auslesen von MEMS-Sensoren

Zuerst soll geprüft werden, ob der serielle Port offen ist. Ist dies der Fall, liest schon ein anderes Programm die Daten aus und dieses Programm wird geschlossen. Ist der serielle Port offen kann das eigentliche Programm beginnen. Es soll eine Textdatei angelegt werden, in der die Daten geschrieben werden sollen. Diese soll als Dateinamen den Namen des Sensors und das aktuelle Datum sowie die aktuelle Uhrzeit erhalten. Es sollen die empfangenen Daten zeilenweise in die Datei geschrieben werden. Da drei Werte pro Zeiteinheit empfangen werden sollen diese mit Leerzeichen getrennt in eine Zeile geschrieben werden. Nach Ablauf einer bestimmten Zeit soll das Programm sich selbstständig schließen.

4.1.6 Umsetzung des Programmschemas

Nach dem Erstellen des Programmablaufplans muss dieser nun in Programmiersprache Python umgesetzt werden. Hierfür wurde die Python Version 3 verwendet. Bei der Erstellung des Skriptes wurde mit dem Modul *PyScripter* gearbeitet. Die folgende Abbildung 4.3 zeigt die Oberfläche des Moduls *PyScripter* mit einem Teil des ersten Programmentwurfs.

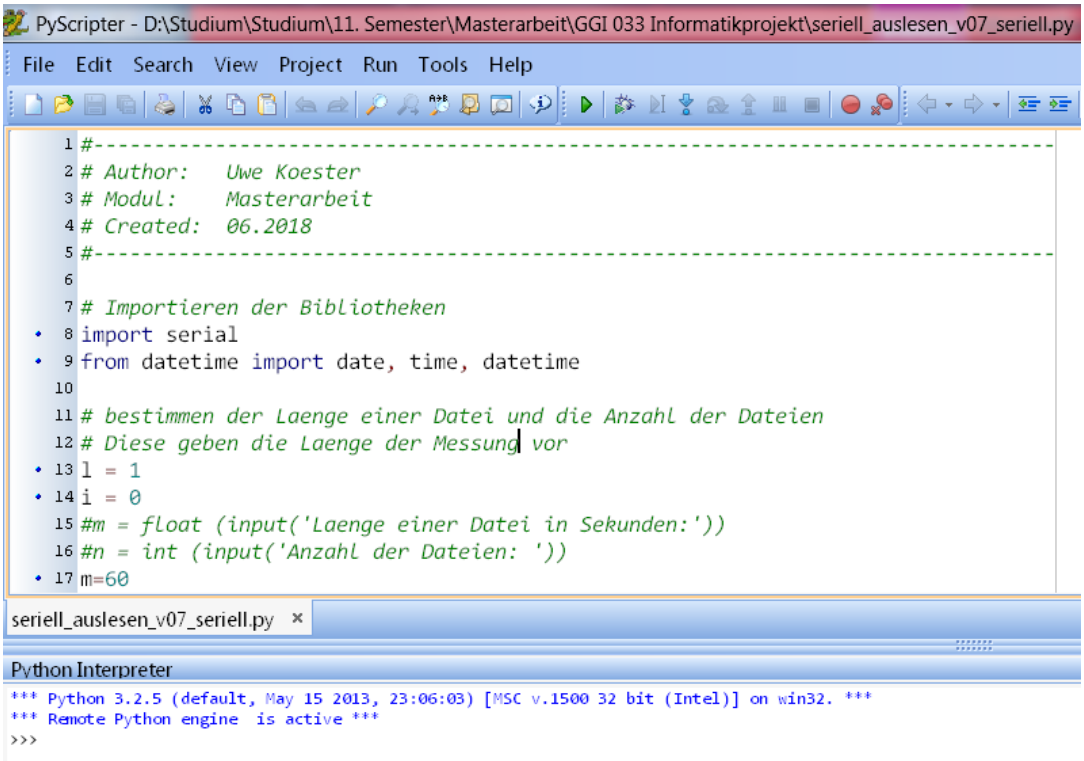


Abbildung 4.3: PyScripter-Oberfläche und erster Programmentwurf

Das Modul *PyScripter* hat als Entwicklungsumgebung eine klar strukturierte und ansprechende Oberfläche. Die Bedienung des Moduls *PyScripter* ist leicht zu erlernen und ermöglicht es, sich auf die eigentliche Skriptentwicklung zu konzentrieren. Er bietet eine große Auswahl an Funktionen, wie zum Beispiel die Syntaxanalyse, die Codevervollständigung oder das Anzeigen der Inhalte der verwendeten Variablen.

Bei der Erstellung des Programms werden zunächst die erforderlichen Bibliotheken importiert und die Herstellung der Kommunikation mit dem Mikrocontroller *Arduino* über das Öffnen des seriellen Ports getestet. Dabei zeigten sich bereits erste Probleme, die im Kapitel 4.1.7 näher beschrieben werden.

Nachdem die Kommunikation mit der seriellen Schnittstelle hergestellt ist, wird nun der Dateiname angelegt und bis zum zeitlichen Abbruchkriterium in eine Datei geschrieben. Diese Datei beinhaltet in einer Zeile den Zeitstempel und die Beschleunigungswerte aller drei Achsen des Beschleunigungssensors bzw. die Temperaturwerte des Temperatursensors.

Tabelle 4.1: Beispiel für die Ergebnisdatei des Beschleunigungssensors

Zeitstempel	Beschleunigung in X	Beschleunigung in Y	Beschleunigung in Z
1481641955.602	-10.6312503	-1.64999997	-2.35312509
1481641955.736	-10.6687498	-1.64999997	-2.35312509
1481641955.868	-10.6593751	-1.64062500	-2.38124990
1481641956.002	-10.6218748	-1.64999997	-2.36249995

Wird nun die angegebene zeitliche Länge einer Datei überschritten, so werden die Daten in eine neue Datei geschrieben. Dies hat den Vorteil, dass die Dateien nicht unendlich groß werden und später noch sinnvoll ausgewertet werden können. Denn bei einer Frequenzanalyse hängt die Rechenzeit vom Quadrat der Einzelwerte ab. Nach Ablauf der Gesamtlaufzeitzeit des Programmes wird der serielle Port geschlossen.

4.1.7 Fehlerbehebung und Optimierung

Das Öffnen des richtigen COM-Portes ist wichtig, da nur an diesem Port die seriellen Daten empfangen werden können. Dabei sind auf die Unterschiede von Windows basierten Rechnern und dem *Raspberry Pi* zu achten.

```
try:
    ser = serial.Serial(port='/dev/ttyACM0',
                        baudrate=115200,
                        bytesize=serial.EIGHTBITS,
                        parity=serial.PARITY_NONE,
                        timeout=200)
    print ("Serial port is open")
except Exception as e:
    print ("error open serial port: " + str(e))
    exit()
```

Abbildung 4.4: Syntax zum Öffnen des seriellen Portes

Des Weiteren gab es Probleme mit dem Öffnen und Schließen des seriellen Ports. War dieser schon von einem anderen Programm geöffnet, konnte keine Datenspeicherung vorgenommen werden. Verschiedene Probleme traten dabei auf, die empfangenen Informationen als Zahlen oder Strings in eine Datei zu speichern. So war das Einlese-Format in Bytes vorgegeben. Dieses Format musste erst in ein Zahlenformat und dann in ein String-Format umgewandelt werden. Ein String-Format ist vorteilhafter für die Weiterverarbeitung

der Daten. Nach dem Start des Programmes wurde die Verbindung mit dem seriellen Port hergestellt. Danach hat sich das Programm in die Kommunikation eingeschaltet und den Datenstrom abgehört. Der Start des Empfangs der empfangenen Daten erfolgte aber mitten in einer Übermittlung. Dadurch ist die erste Zeile des Empfangs meistens nicht vollständig übertragen worden.

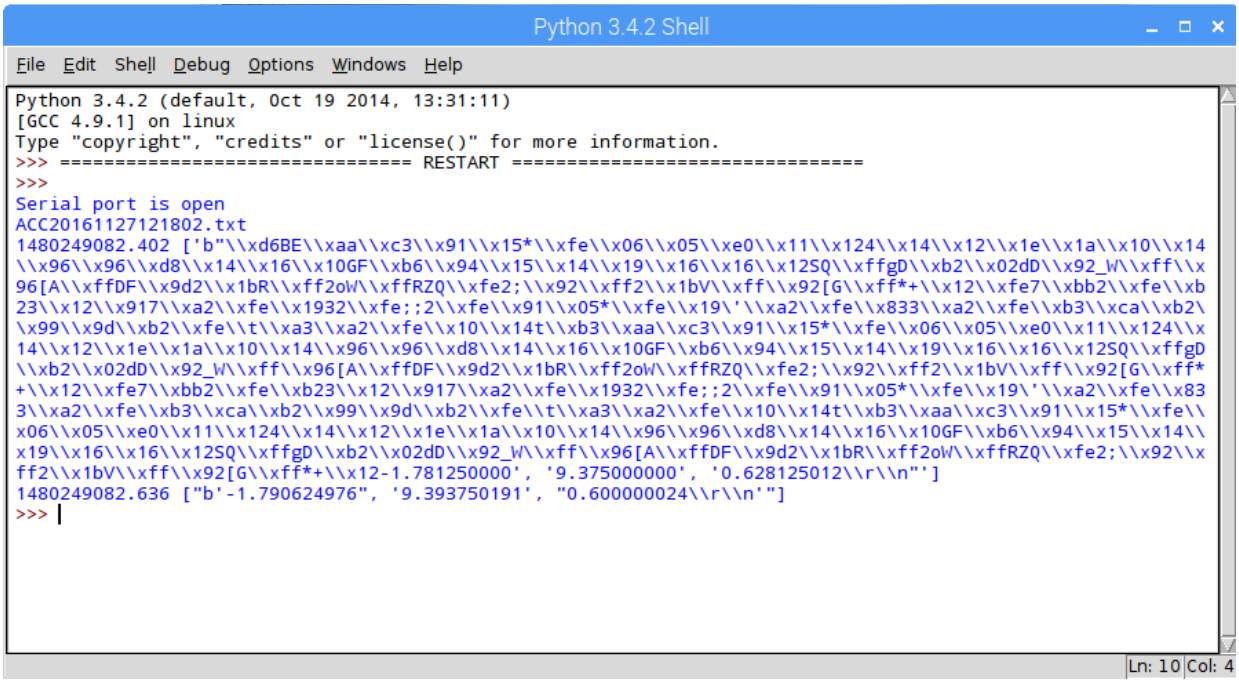


Abbildung 4.5: Probleme mit dem Start des Empfangs der seriellen Daten

Dieses Problem ist dadurch gelöst worden, indem die ersten Zeilen in der ersten Datei nicht abgespeichert werden. Dies ist kein Problem, da der Sensor Daten mit einer Abtastrate von bis zu 500 Hz liefert und somit nur die ersten zehn Millisekunden nicht gespeichert werden.

Des Weiteren traten Probleme mit den Füllnullen des Datums und Uhrzeitformates auf. So wurde z.B. bei einer Sekundenabgabe zwischen 0 und 9 keine führende Null ausgegeben. Für die Abspeicherung der Datei ist diese aber von Bedeutung, da die Dateien mit nach Uhrzeit und Datum automatisch über den Dateinamen sortiert werden sollten. Es musste eine Null entsprechend eingefügt werden, um die Sortierung zu gewährleisten. Ausgehend vom ersten Programmablaufplan, der die grundlegenden Funktionen beinhaltete, ist dieser erweitert worden. So ist z.B. eine zweite Schleife für die automatische Vergabe der Dateinamen hinzugekommen, siehe Abbildung 4.6.

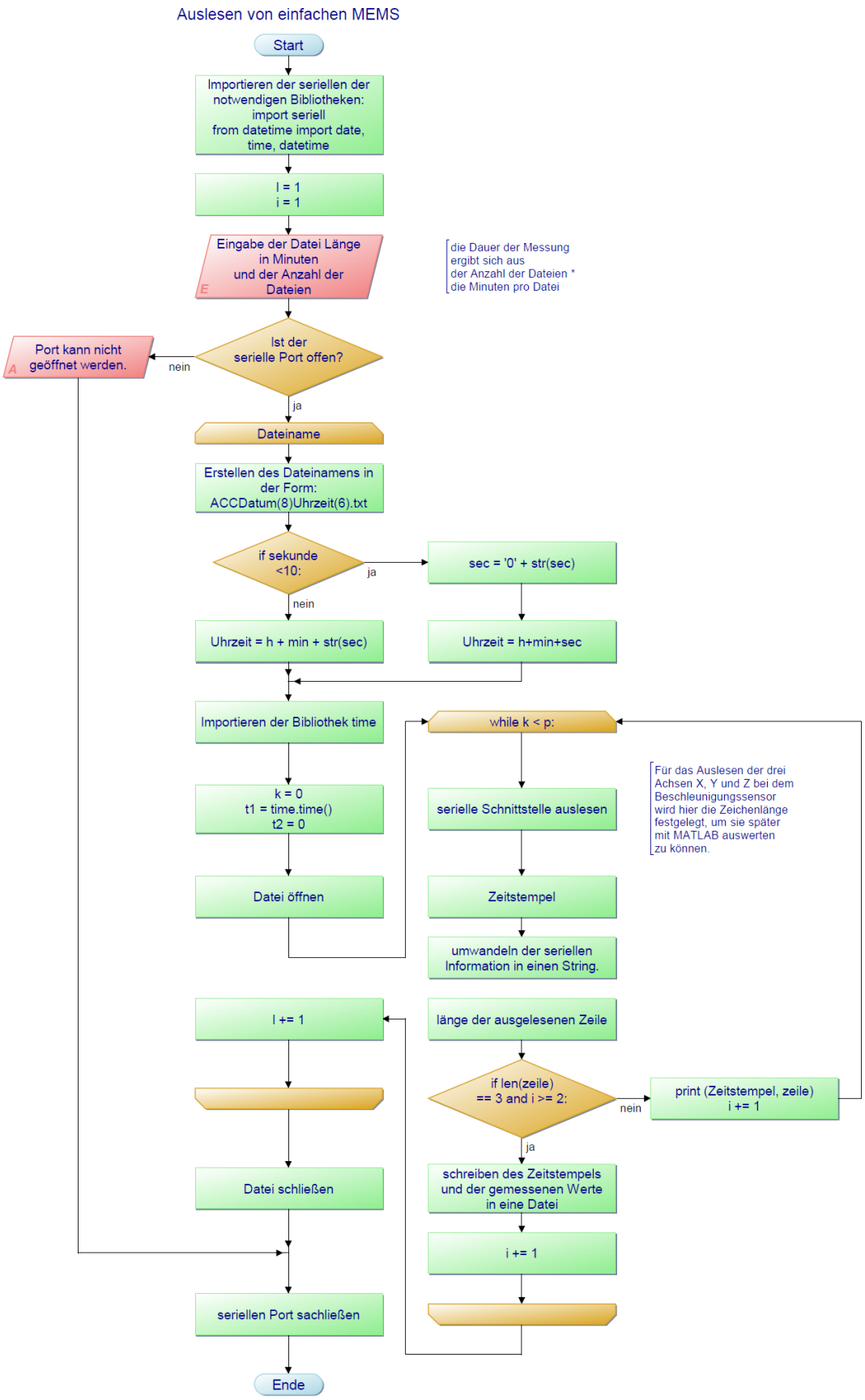


Abbildung 4.6: überarbeiteter Programmablaufplan

4.2 Entwurf und Konstruktion der Sensorblöcke

Das Ziel dieser Masterarbeit ist die Entwicklung eines funktionierenden Sensors für geodätische Überwachungsmessungen. Bisher ist sind schon einige Aussagen zur Hard- und Software getroffen worden. Allerdings ist der Sensor selbst nur als Testobjekt in Kombination mit einem Mikrocontroller und einem Breadboard getestet worden. Um daraus einen funktionierenden Sensor zu erhalten, müssen alle Komponenten in ein entsprechendes Gehäuse eingebaut werden und fest miteinander verbunden werden. Dies stellt besondere Anforderungen an die Komponenten und den Zusammenbau.

4.2.1 Anforderungen an die Hardware

Der hier zu entwickelnde Sensor soll universell für viele Einsatzbereiche der Ingenieurgeodäsie konzipiert werden. Eine Hauptanforderung ist der preiswerte Aufbau, da nur so eine große Verbreitungsmöglichkeit gegeben ist. Daneben soll der Sensor selbst sehr robust aufgebaut sein, um den Vandalismus zu erschweren. Außerdem soll der Sensor staubdicht und wasserdicht ausgelegt sein, um im Außenbereich eingesetzt werden zu können. Damit können Brücken und andere Verkehrsbauwerke überwacht werden. Auch eine schnelle Austauschbarkeit soll möglich sein. Falls ein Sensoraustausch nötig ist, muss dieser innerhalb kürzester Zeit ausgeführt werden können. Für die Energieversorgung und den Datentransfer müssen entsprechende Anschlussöffnungen am Sensorgehäuse vorhanden sein.

Tabelle 4.2 Anforderungen an Sensoren in verschiedenen Bereichen des ingenieurgeodätischen Monitorings

Bereich	Anforderungen
Talsperren	geschützt gegen Umwelteinflüsse
Brücken	robust und resistent gegen Vandalismus
Hochhäuser	fähig zur Implementierung in eine Monitoring-Software netzwerkfähig
Windenergieanlagen	Wartungsarm
allgemein	preiswert und austauschbar

Aus den Anforderungen an die Hardware ergab sich, dass ein Aluminiumgehäuse alle

Anforderungen erfüllt. Ausgewählt wurde ein Gehäuse mit wasserdichter Verschraubung und Befestigungsmöglichkeiten.

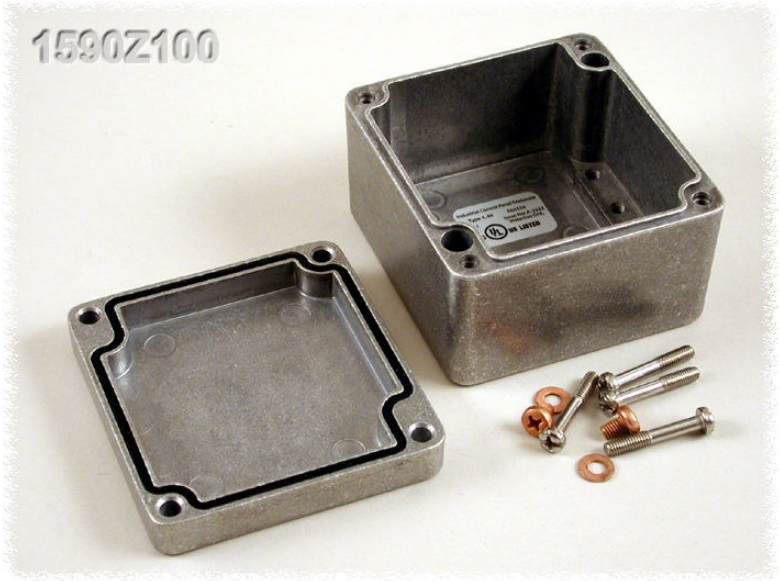


Abbildung 4.7: wasserdichtes Gehäuse

Eine besondere Herausforderung bestand darin, dass alle Elektronikbauteile im Inneren des Gehäuses Platz finden mussten. Dazu wurden im Zuge der Gehäusefindung alle passenden Bauteile miteinander verbunden und abgelegt. Mithilfe dieser Konstellation ist eine Schablone angefertigt worden, die dann mit im Handel verfügbaren Gehäusegrößen verglichen wurde. Eine Mindestgröße von 70 mm x 70 mm ist erforderlich. Nun konnten Befestigungsmöglichkeiten und die Wasserdichtheit überprüft werden.

4.2.2 Schnittstellen

Die verwendeten Schnittstellen sind im Kapitel 3.1 näher beschrieben worden. Bei dem zu konstruierenden Sensor ist eine Prämisse, dass so wenige wie mögliche Anschlusskabel verwendet werden sollten und die spätere Handhabung auch für technisch wenig versierte Nutzer möglich ist. Aus dem Sensorgehäuse sollte nur ein USB-Anschluss vorhanden sein, da USB eine gängige Schnittstelle ist und die Verkabelung einfach auszuführen ist, siehe Abbildung 4.8. Der Anschluss kann auch von Geodäsie- und Technikunkundigen problemlos durchgeführt werden. Alle anderen verwendeten Schnittstellen sind nur im Inneren des Gehäuses vorhanden und führen nicht in das Äußere. Eine zweite Variante wird dann im Kapitel 4.2.4 vorgestellt. Hier stand die Implementierung des Sensors in ein bestehendes

datenbankorientiertes Monitoringsystem, DABAMOS, im Vordergrund. Daher ist bei der zweiten Variante der Datenfluss nicht mittels USB auf den *Raspberry Pi*, sondern mittels serieller Schnittstelle über einen seriell zu einem Ethernet Umsetzer erfolgt. Hier sind dann die Anschlüsse RXD und TXD aus dem Sensor herausgeführt worden.

Die Herausführung des USB Anschlusses, die bei Nichtbenutzung auch mittels einer Kappe verschließbar ist, lässt sich über eine Bohrung am Gehäuse verschrauben. Es sind Dichtungen vorhanden, die die Wasserdichtheit gewährleisten. Der USB-Anschluss wird im Inneren über vier Anschlüsse mit dem Mikrocontroller elektrisch verbunden.

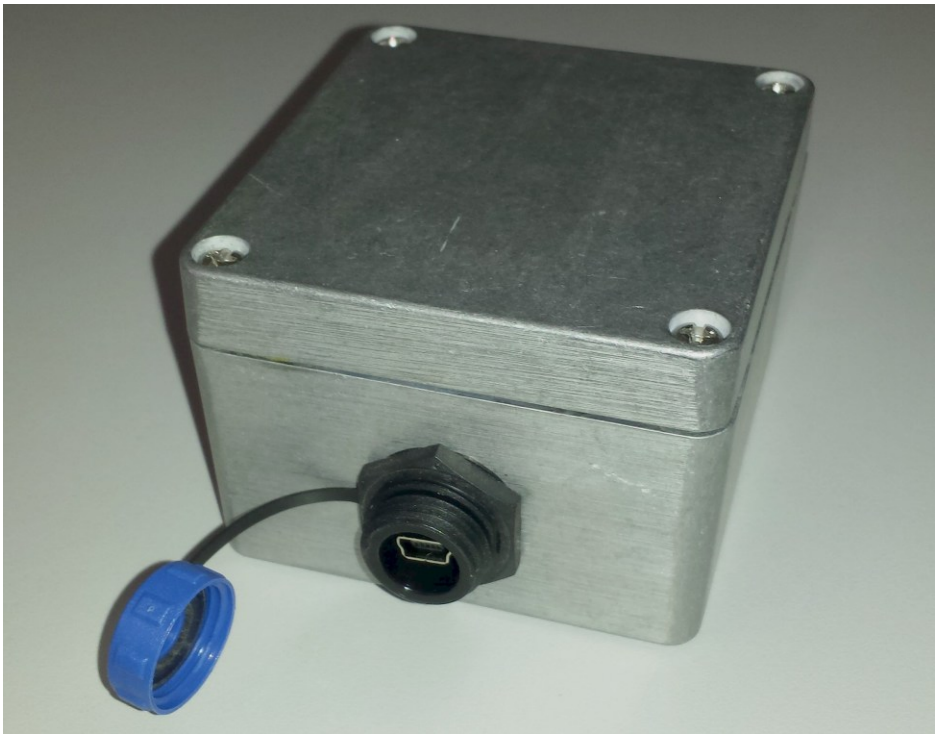


Abbildung 4.8: USB Anschluss am Gehäuse

Nun kann mittels eines entsprechenden wasserdichten Kabels der Mini-USB-Anschluss verwendet werden.

4.2.3 Stromversorgung

Die Stromversorgung wird ebenfalls über den Mini-USB-Anschluss gewährleistet. In Verbindung mit dem *Raspberry Pi* wird von einer USB-Buchse des Einplatinencomputers sowohl die Datenübertragung als auch die Stromversorgung gewährleistet.

Abbildung 4.9: Stromversorgung über den Raspberry Pi

Bei der zweiten Variante, die für den Einsatz mit dem Monitoringsystem vorgesehen ist, kommt ein Stromanschlusskabel zum Einsatz, das die Spannung des am Einsatzort vorhandenen 12 Volt Gleichstromnetz in 5 Volt für den Sensor konvertiert.

Abbildung 4.10: !2 Volt zu 5 Volt Konverter

4.2.4 Konstruktion eines Prototyps

Nachdem alle Komponenten definiert und beschafft worden sind, kann mit der eigentlichen Konstruktion begonnen werden. Die Komponenten werden elektrisch und mechanisch verbunden.

Eine Leiterplatte wurde benutzt, um die einzelnen Komponenten bestmöglich zu verbinden. Der Sensor selbst ist am Boden des Gehäuses mit einem 2-Komponentenkleber befestigt

worden. Auf eine Verschraubung ist aufgrund möglicher Verwindungen nicht in Betracht gezogen worden. Anschließend wurde die Leiterplatte über dem Sensor angebracht. Es ist darauf geachtet worden, dass beide Komponenten keine mechanische Verbindung aufweisen, damit die Leiterplatte keine Kräfte auf den Sensor selbst übertragen kann. Der USB-Anschluss des Mikrocontrollers ist mit dem Mini-USB Stecker verlötet worden. Die Leiterplatte ist ebenfalls mit Pullup-Widerständen für den störungsfreien I^2C Betrieb ausgestattet worden.

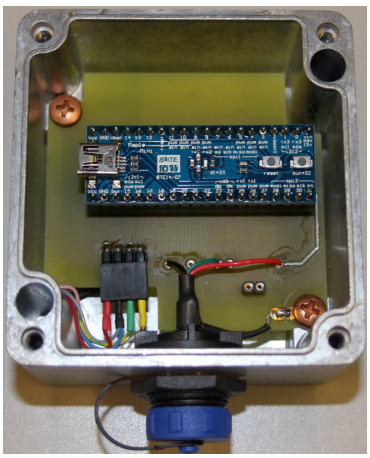


Abbildung 4.11: fertiger Prototyp

5 Zeitreihenanalyse

Diese Masterarbeit befasst sich mit dem Entwurf, der Konstruktion, der Kalibrierung und der Anwendung eines MEMS-basiertem Sensors zur geodätischen Überwachungsmessung. Hierbei ist die Auswertung der Messwerte von besonderer Bedeutung. Die Messwerte, die gesammelt und ausgewertet werden sollen, sind zeitlich abhängig. Man spricht auch von stochastischen Prozessen. Stochastische Prozesse sind nicht vollständig vorhersehbar und unterscheiden sich dadurch von deterministischen Prozessen, deren Realisierungen x(t) vollständig vorhersehbar sind. Die zeitlich geordnete Folge von Beobachtungen x(t) zu diskreten Zeitpunkten ergibt dann eine Zeitreihe. [Neuner, et al., 2009]

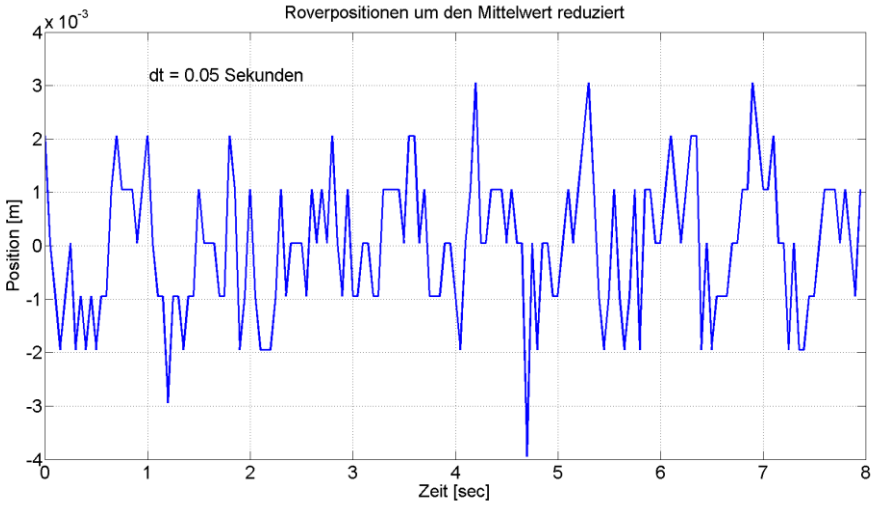


Abbildung 5.1: Positionen eines GNSS Rovers, ein Beispiel für eine Zeitreihe

5.1 Die Zeitreihe im Zeitbereich

In dem in Abbildung 5.1 dargestellten Beispiel für eine Zeitreihe beträgt Δt=0,05 Sekunden. Dies wird als Schrittweite der Abtastung bezeichnet. Sie soll als konstant angenommen werden. Ist die Schrittweite nicht konstant, so ist eine korrekte Analyse einer Zeitreihe nicht möglich. Der Kehrwert 1/ Δt ergibt die Abtastfrequenz f. Aus Abbildung 5.1 ist ersichtlich, dass die Messwerte selbst, hier schon auf den Mittelwert reduziert, sowie der zeitliche Verlauf, hier acht Sekunden einer Zeitreihe, dargestellt sind. Informationen über mögliche Frequenzen sind hingegen nur dann möglich, wenn eindeutige sinusförmige Schwingungen mit großer sichtbarer Amplitude auftreten. Es ist auch nur das Abschätzen von maximal zwei Frequenzen aus dem Schaubild, nachfolgend Plot genannt, möglich.

5.1.1 Kovarianzfunktionen

Bei einem Zufallsprozess X(t) sind $x_i(t)$ mit i = 0, 1, 2, …, n die Realisierungen dieses Prozesses. Der Erwartungswert des stochastischen Prozesses wird in Formel (2) angegeben.

$$\mu_x(t) = E\{X(t)\} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} x_i(t) \tag{2}$$

Die Varianz eines stochastischen Prozesses ergibt sich aus Formel (3).

$$\sigma^2{}_x(t) = E\{[X(t) - \mu_x(t)]^2\} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} [x_i(t) - \mu_x(t)]^2 \quad (3)$$

Die Autokovarianzfunktion gibt die Ähnlichkeit einer um die Zeit τ verschobene Folge x(t+ τ) mit der ursprünglichen Folge x(t) wieder. Da Folgeglieder nicht stochastisch unabhängig sind beschreibt die Autokovarianzfunktion C(τ) diese Abhängigkeiten. [Neuner, et al., 2009]

$$\begin{aligned} C_{xx}(t,\tau) &= E\{[X(t) - \mu_x(t)][X(t+\tau) - \mu_x(t+\tau)]\} \\ &= \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \{[x_j(t) - \mu_x(t)][x_j(t+\tau) - \mu_x(t+\tau)]\} \end{aligned} \quad (4)$$

Da die Autokovarianzfunktion nicht skaleninvariant ist, wird diese normiert und man erhält die Autokorrelationsfunktion.

$$R_{xx}(t,\tau) = \frac{C_{xx}(t,\tau)}{\sigma_x(t) * \sigma_x(t+\tau)} \quad (5)$$

Die Periodizitäten können in C(τ) und R(τ) abgelesen werden. Ebenfalls kann aus R(τ) entnommen werden, inwiefern die Werte der Zeitreihe miteinander korreliert sind.

Zwei wichtige Bedingungen von Zeitreihen sind Stationarität und Ergodizität. Für die Einhaltung der Stationaritätsbedingung dürfen Mittelwert und Varianz der Zeitreihe über den ganzen Verlauf hin keine Änderungen aufweisen. Deshalb ist es besonders wichtig eine Zeitreihe vor ihrer Analyse von möglichen Trends zu befreien. Ergodisch ist eine Zeitreihe dann, wenn von dieser Zeitreihe auf eine andere gefolgert werden kann.

5.1.2 Datenaufbereitung

Um eine korrekte Analyse der Zeitreihe gewährleisten zu können müssen die Daten aufbereitet werden. Vor allem Ausreißer, Sprünge, Trends und Datenlücken sind zu beseitigen. [DVW, 2004]

Ein Ausreißer ist ein Messwert der Zeitreihe, der nicht in die erwartete Reihe passt. Es handelt sich hierbei um einzelne Ereignisse, dessen Werte um ein Vielfaches größer oder

kleiner als das Mittel der Zeitreihe sind. Ausreißer entstehen vor allem durch fehlerhafte Datenübertragung oder einen Defekt des Sensors.

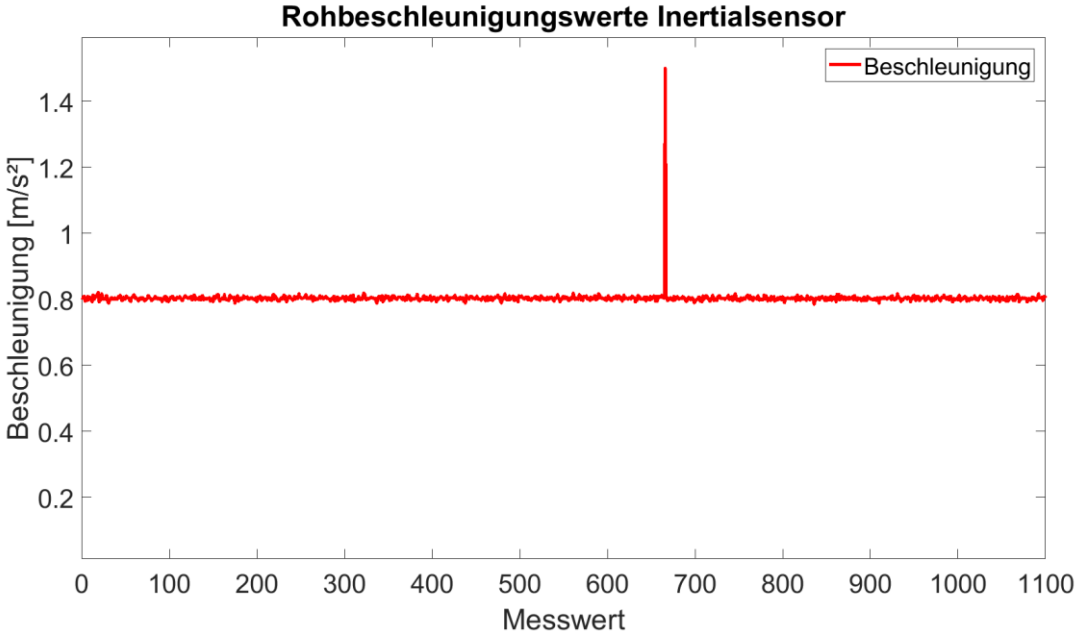


Abbildung 5.2: Beispiel für einen Ausreißer in einer Zeitreihe

Ausreißer müssen erkannt und entsprechend beseitigt werden. Da es sich hierbei meist nur um einzelne Ereignisse handelt, können Methoden des nächsten Nachbarn oder die Mittelung beider Nachbarn genutzt werden.

Ein Sprung in den Daten lässt auf einen Defekt des Sensors oder auf abrupte Änderungen der Eingangsgröße schließen.

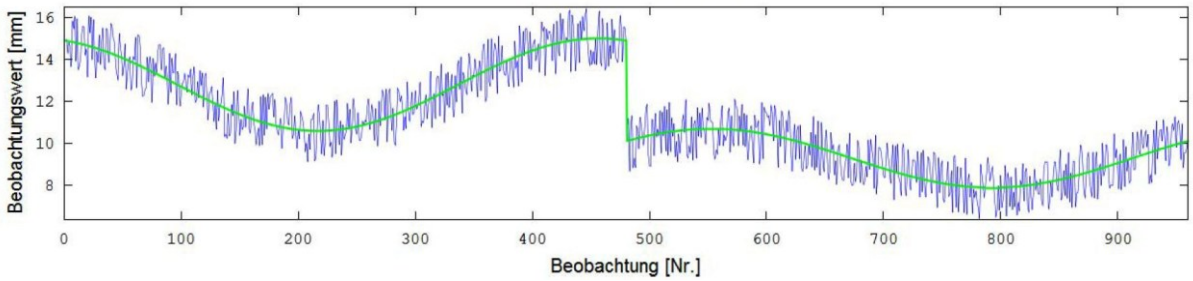


Abbildung 5.3: Beispiel für einen Sprung in den Messdaten

Sprünge können eliminiert werden, indem die Zeitreihe auf beiden Seiten des Sprunges auf einen separaten Mittelwert reduziert wird.

Trends sind langfristige Zunahmen oder Abnahmen der Größe der Messwerte. Sie können die Form von Polynomen verschiedener Ordnung annehmen. Die Ursachen von Trends

liegen oft in Nichtlinearität der Sensordatenwerte oder fehlerhafter Kalibrierung.

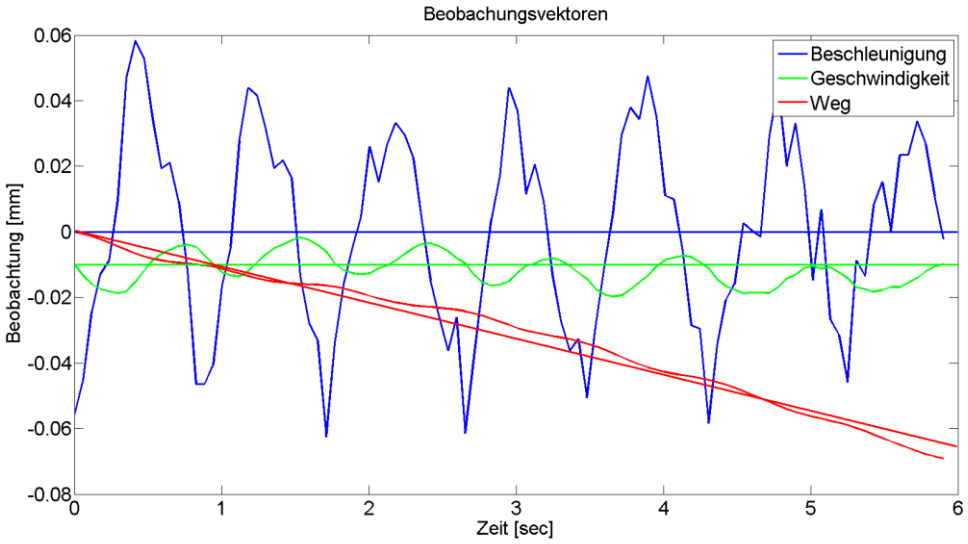


Abbildung 5.4: Beispiel für Offset (grüne Gerade) und linearen Trend (rote Gerade) innerhalb einer Zeitreihe

Trends lassen sich am elegantesten mittels einer Ausgleichung nach vermittelnden Beobachtungen, einer linearen Regression beseitigen.

Datenlücken sind durch temporäre Ausfälle der Aufzeichnung der Messwerte bedingt. Hier kann entweder die Datenübertragung nicht funktioniert haben, der Datenspeicher kann voll sein oder der Sensor weist ein Totalausfall auf. Datenlücken müssen auf jeden Fall beachtet werden, um den korrekten Ablauf einer Analyse zu gewährleisten.

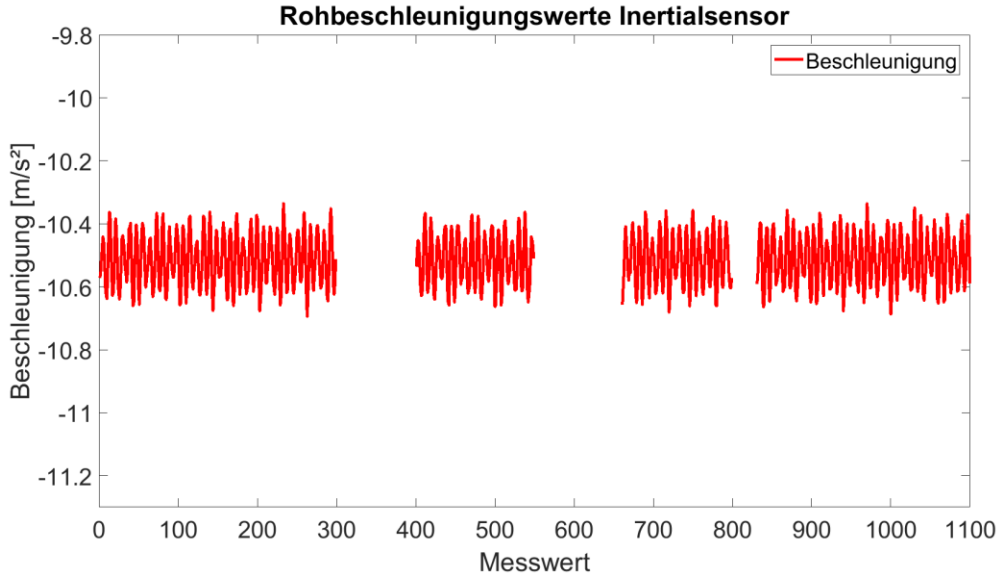


Abbildung 5.5: Zeitreihe mit Datenlücken

Um Datenlücken zu beseitigen sind viele Ansätze möglich. Lineare Modelle, die sehr einfach

anzuwenden sind, sowie Spline-basierte Verfahren, die rechenintensiver sind, können verwendet werden.

Das Ziel ist eine Zeitreihe, die frei von Ausreißern, Sprüngen, Trends und Datenlücken ist. Diese kann dann mit den nachfolgend beschriebenen Analysemethoden ausgewertet werden.

5.2 Die Zeitreihe im Frequenzbereich

Grundbaustein für die spektrale Datenanalyse ist ein Satz, den der französische Mathematiker Jean Baptiste Joseph [seit 1808] Baron de Fourier (1768–1830) Anfang des 19. Jahrhunderts formulierte:

Jede periodische Funktion mit einer Periodendauer T lässt sich als Summe von Sinus- und Kosinusfunktionen mit ganzzahligen Vielfachen einer Grundfrequenz $F = \frac{1}{T}$ ausdrücken.

Für eine gegebenen Funktion $f(t)$ mit der Periodendauer T (bzw. auf dem Intervall $[a; b]$) wird das trigonometrische Polynom der Periode $b - a$, das f auf dem Intervall $[a; b]$ optimal approximiert gesucht. Dazu ist es notwendig, die Amplituden der einzelnen aufsummierten Schwingungen in passender Weise zu gewichten. Die mathematische Formulierung der sogenannten FOURIER-Reihe einer periodischen Funktion ist folglich:

$$f(t) = \frac{1}{2}a_0 + (a_1\,cos(\omega t) + b_1\,sin(\omega t)) \;+\; (a_2\,cos(2\omega t) + b_2\,sin(2\omega t)) + \cdots$$
$$= \frac{1}{2}a_0 + \sum_{k=1}^{\infty}(a_k\,cos(k\omega t) + b_k\,sin(k\omega t))\,, wobei\;\; \omega = \frac{2\pi}{b-a} = \;\frac{2\pi}{T} \tag{6}$$

Es sind also $2n + 1$ unbekannte Koeffizienten $a_0, \ldots a_n,\; b_1 \ldots b_n$ zu bestimmen. Die einzelnen Sinus-und Kosinusschwingungen mit Frequenzen kF, aus denen die Funktion f mit Hilfe der Reihenbildung konstruiert wird, werden also mit den FOURIER-Koeffizienten a_k und b_k gewichtet und in geeigneter Weise gegeneinander verschoben. [Nitschke, 2012]

Einfach ausgedrückt kann gesagt werden, dass die Koeffizienten angeben, wie "stark" der jeweilige Frequenzanteil in der betrachteten Funktion ist. Deshalb sind es gerade diese beiden Größen, die es zu bestimmen gilt, wenn man eine Funktion durch ihre FOURIER-

Koeffizienten beschreiben möchte. [Thuselt, et al., 2013]

Die dafür notwendigen Ausdrücke lauten:

$$a_k = \frac{2}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t)\,cos(k\omega t)\,dt \qquad und \qquad b_k = \frac{2}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t)\,sin(k\omega t)\,dt \tag{7}$$

Wie bei dem folgenden Beispiel erkennbar, wird in der Praxis eine endliche Anzahl an Koeffizienten berechnet. Dies ist vor allem zweckmäßig, wenn die Fourier-Reihe konvergiert. Bei dem Beispiel einer periodischen Rechteckfunktion können die Koeffizienten mittels Schleife berechnet werden, wobei aus Abbildung 5.6 ersichtlich ist, dass hier lediglich die Sinusanteile der Koeffizienten b_k eine Rolle spielen.

```
tmax = 6*pi; %max von t
t = linspace(0,tmax,1000); %Zeitvektor
h= 1; %Amplitude
N1 = 2; N2 = 20;
y1 = zeros(1,length(t)); y2=y1;
for k = 1:N1
    y1 = y1 + sin((2*k-1)*t)/(2*k-1);
end
for k = 1:N2
    y2 = y2 + sin((2*k-1)*t)/(2*k-1);
end
y1 = (4*h/pi) * y1; %Normierung
y2 = (4*h/pi) * y2;
plot(t,y1,'k',t,y2,'b',t,square(t),'r');
title('FOURIER-Approximation eines
Rechtecksignals')
xlabel('t'); ylabel('y=f(t)');
legend('N=2','N=20','Rec-Impuls');
grid on
```

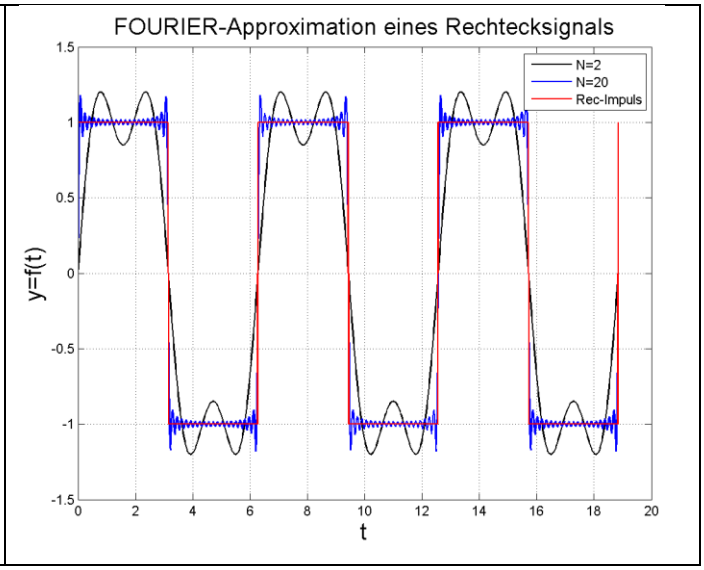


Abbildung 5.6: FOURIER-Approximation eines Rechtecksignals

Durch Anwendung der EULERschen Formel $e^{i\phi} = \cos(\phi) + i\,sin(\phi)$ lässt sich die Reihe stattdessen auch mit einer komplexen Exponentialfunktion darstellen

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{ik\omega t} \qquad ,mit \qquad c_k = \frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t)e^{-ik\omega t}\,dt \tag{8}$$

Diese kompakte Darstellung wird in der Regel bevorzugt, da nur noch ein Set von FOURIER-Koeffizienten c_k berechnet werden muss

5.2.1 Fouriertransformation

Während im Abschnitt 5.2 die Zerlegung von periodischen Funktionen mittels FOURIER-Reihe beschrieben wurde, handelt es sich in der Praxis vor allem um die Analyse

aperiodischer Funktionen. Diese nichtperiodischen Funktionen mit einer Periodendauer, die gegen unendlich geht, werden in ähnlicher Weise zerlegt. Der Abstand der einzelnen Frequenzanteile kF wird immer kleiner, da $F = \frac{1}{T}$ für $T \to \infty$ gegen null geht, und die Summe der FOURIER-Reihe geht in ein Integral über. Außerdem wird nun anstatt diskreter Koeffizienten eine kontinuierliche Funktion $\hat{f}(F)$ aller möglichen Frequenzen betrachtet. Diese Funktion heißt FOURIER-Transformierte und enthält das Frequenzspektrum. Dieses gibt die Zusammensetzung eines Signals aus seinen Bestandteilen an. [Thuselt, et al., 2013]

Für das Anwendungsbeispiel der Schwingungsüberwachung in dieser Masterarbeit, bietet sich der Begriff *Frequenzanalyse* an. Korrekterweise sei darauf hingewiesen, dass bei den FOURIER-Transformationen nicht nur Signale vom Zeit- in den Frequenzbereich transformiert werden, sondern auch beispielsweise räumliche Signale analysiert werden können. Im allgemeinen Fall spricht man vom Original- und Bildbereich. Die kontinuierliche FOURIER-Transformation von $f(t)$ nach $\hat{f}(F)$ und die entsprechende Rücktransformation ist in Formel (9) angegeben.

$$\hat{f}(F) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i F t} dt \;\; und \;\; f(t) = \int_{-\infty}^{+\infty} \hat{f}(F) e^{2\pi i F t} dt \tag{9}$$

In der Praxis stehen, wie in diesem Anwendungsbeispiel, die Werte einer Zeitreihe jedoch normalerweise nur an diskreten Zeitpunkten und in endlicher Anzahl zur Verfügung. Der Frequenzbereich eines zeitdiskreten Signals kann über eine diskrete FOURIER-Transformation (DFT) numerisch in einen Frequenzspektrum umgerechnet werden. Der Frequenzbereich muss dafür jedoch auch diskretisiert werden, das heißt, die FOURIER-Transformation wird nur bei bestimmten Frequenzen ausgewertet.

Die diskrete FOURIER-Transformation von dem Signal y_n zur diskreten FOURIER-Transformierten Y_{k+1}, bei N zeitäquidistanten Werten und die Rücktransformation zu den Abtastpunkten ist in Formel (10) angegeben.

$$\begin{aligned} Y_{k+1} &= \sum_{n=0}^{N-1} y_{n+1}\, e^{-2\pi i \frac{nk}{N}} \quad , für\ k = 0,1,2, \dots N-1 \\ y_{n+1} &= \frac{1}{N} \sum_{k=0}^{N-1} Y_{k+1}\, e^{2\pi i \frac{nk}{N}} \quad , für\ k = 0,1,2, \dots N-1 \end{aligned} \tag{10}$$

5.2.1.1 Fast-Fourier-Transformation

Die Berechnung nach diesem Schema erfordert jedoch viele Rechenoperationen. Abhilfe schafft die schnelle FOURIER-Transformation (FFT), ein Algorithmus, der durch geschickte Faktorisierung von Matrizen eine deutlich effizientere Berechnung der diskreten FOURIER-Transformierten ermöglicht. Voraussetzung dafür ist allerdings, dass die Anzahl der Abtastpunkte N durch eine Zweierpotenz gegeben ist, also $N = 2^x$.

Als anschauliches Beispiel für die FFT und eine spätere Betrachtung durch eine STFT, siehe Kapitel 5.3.1, dient in Anlehnung an die Literatur „Praktische Mathematik mit MATLAB, Scilab und Octave“ der Analyse eines akustischen Signals.

Es werden zunächst die Schwingungen eines Dreiklangwechsels von den C-Dur Tönen c‘, e‘ und g‘, zu den F-Dur Tönen f‘, a‘ und c‘‘ simuliert, siehe Abbildung 5.7. Die Frequenzen von C- zu F-Dur stehen im Verhältnis 4/3. Daraus ergeben sich die simulierten Frequenzen zu: 264Hz, 330Hz und 396Hz sowie 352Hz, 440Hz und 528Hz. Nach dem *Nyquist-Shannon Abtasttheorem* muss ein Signal mit mindestens der doppelten Frequenz der maximal im Signal enthaltenen Frequenz abgetastet werden $(f_{abtast} > 2 * f_{max})$, um es exakt rekonstruieren zu können. Da die vom Menschen maximal wahrnehmbare Frequenz bei $\sim 22kHz$ liegt, werden aus den simulierten Schwingungen nun diskrete Datenpunkte mit einer Abtastfrequenz von 44,1 kHz gewählt. Aus Gründen der Übersichtlichkeit soll der Tonwechsel bereits nach einer ¼ Sekunde erfolgen und der Zeitvektor eine Länge von einer ½ Sekunde haben.

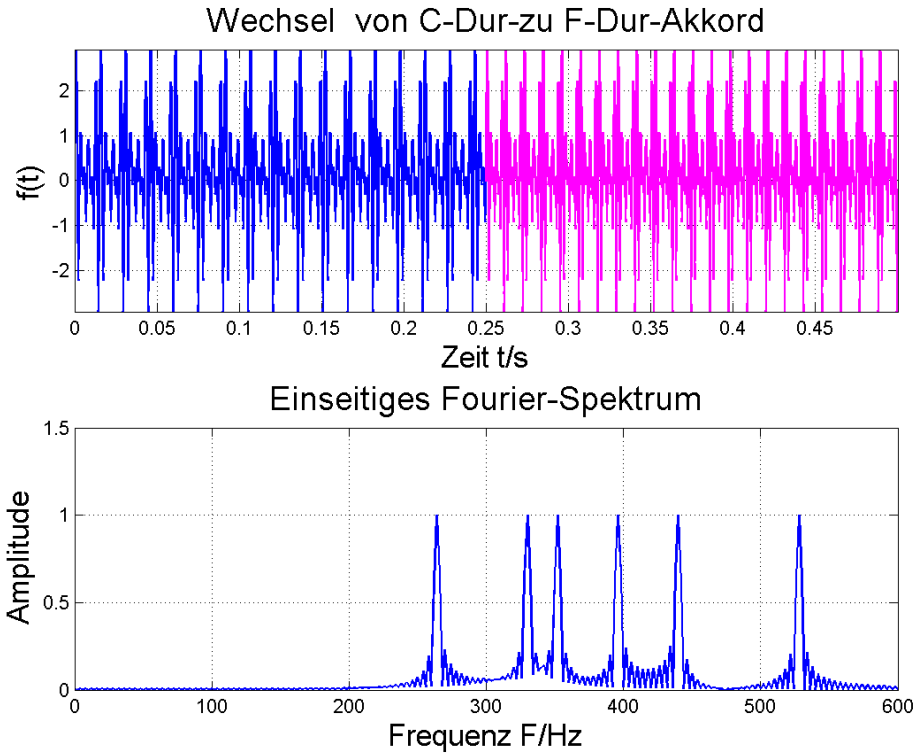


Abbildung 5.7: FFT eines Tonleiterwechsels

Aus Abbildung 5.7 geht hervor, dass die simulierten Frequenzen und auch deren Amplituden durch die FFT im Spektrum gut erfasst wurden. Obwohl die Frequenzen richtig erkannt wurden, kann nicht zugeordnet werden, welche Töne zu welcher Zeit erklungen sind. Um neben der Frequenz- auch die Zeitinformationen des Signals sichtbar zu machen, kann die *Kurzzeit-FOURIER-Transformation* verwendet werden.

5.3 Die Zeitreihe im Zeitbereich und Frequenzbereich

Bisher ist die Betrachtung einer Zeitreihe immer in zwei Kategorien eingeteilt worden. Zuerst erfolgt die Betrachtung der Zeitreihe im Zeitbereich und danach die Betrachtung derselben Zeitreihe im Frequenzbereich. Erfolgen Analysen im Zeitbereich, wie z.B. die Analyse der Amplitude, sind die Frequenzen nicht verfügbar. Wird der Frequenzbereich analysiert, dann fehlen die Zeitinformationen. So kann nicht festgestellt werden, welche Frequenz an welchem Zeitpunkt auftritt. Damit ist es nicht möglich, einzelne Schwingungen ihrem zeitlichen Auftreten zuzuordnen. Dies ist eine sehr wichtige Aufgabe, da die auftretenden Schwingungen bei Monitoringaufgaben ihrer Ursache zugeordnet werden sollen. Dies ist nur mit einer Zeitinformation möglich. Nun soll eine Möglichkeit aufgezeigt werden, wie eine

kombinierte Analyse im Zeitbereich und Frequenzbereich möglich ist.

5.3.1 Kurzzeit Fouriertransformation

Bei der *short-time FOURIER transform (STFT*) wird das Signal in Zeitintervalle unterteilt und die Transformation für jedes Segment separat durchgeführt. Dafür wird das Signal $f(t)$ mit einer sogenannten Fensterfunktion $g(t)$ multipliziert, wobei die Fensterfunktion nur einen begrenzten Zeitbereich abdeckt und über einen Verschiebungsparameter τ entlang der Zeitachse verschoben werden kann. Definiert ist die *STFT* in Formel (11).

$$\hat{f}(F,\tau) = \int_{-\infty}^{+\infty} f(t)g(t-\tau)e^{-2\pi iFt}dt \tag{11}$$

Bei zeitdiskreten Signalen muss anstatt des Integrals wieder auf eine Summe übergegangen werden. Bei der Auswertung durch das Numerikprogramm Matlab R2017a wurde jedoch die in der *„Signal Processing Toolbox“* enthaltene *spectrogam-Funktion* benutzt.

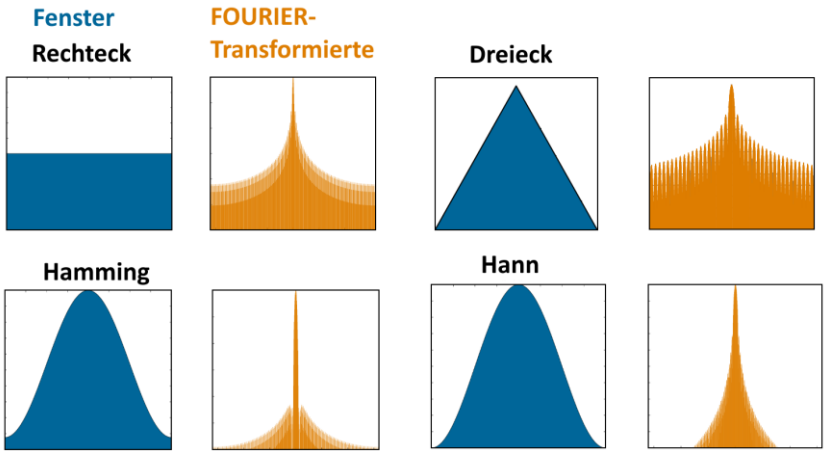


Abbildung 5.8: Auswahl von Fensterfunktionen [K, 2005]

Für das „Herausfenstern“ der Werte einer Zeitreihe innerhalb eines bestimmten Zeitbereichs existieren je nach Anwendung verschiedene Fensterfunktionen, siehe Abbildung 5.8. Maßgeblichen Einfluss auf das Ergebnis hat neben $g(t)$ die Breite des Fensters τ. Die *STFT* unterliegt der *HEISENBERGschen Unschärferelation*, die besagt das zwei komplementäre Eigenschaften eines Teilchens nicht gleichzeitig beliebig genau bestimmbar sind. [Mertins, 2013]

Abbildung 5.9 verdeutlicht den Zusammenhang zwischen der Auflösung von Frequenz- und Zeitinformation.

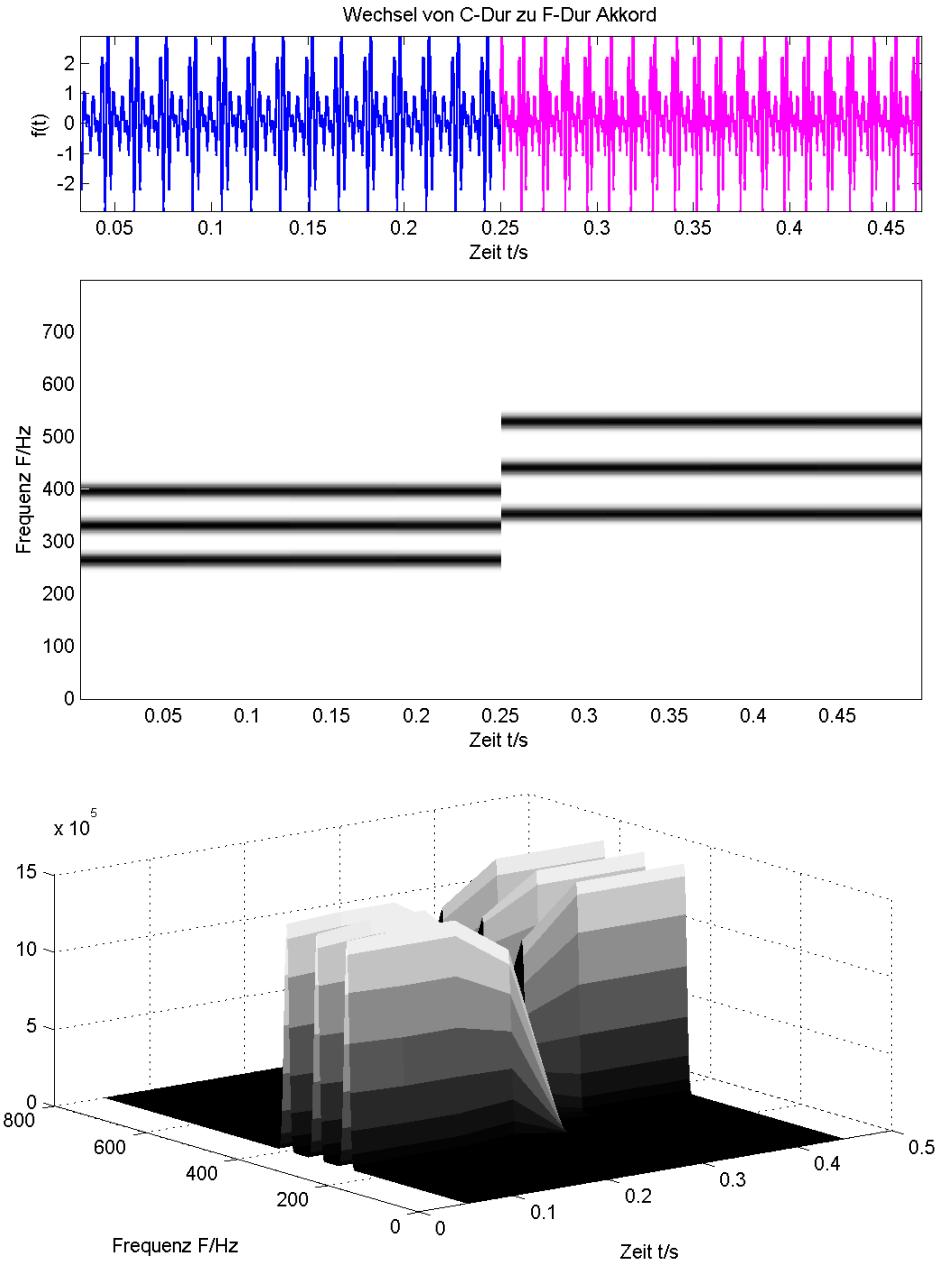


Abbildung 5.9: STFT eines Tonleiterübergangs in zwei- und dreidimensionaler Darstellung

Abbildung 5.9 zeigt, dass sowohl der Zeitpunkt des Dreiklangwechsels, als auch die einzelnen Töne mit ihren Amplituden erkennbar sind. Die gefensterte FOURIER-Transformation gibt ein Ergebnis mit gleichbleibender Frequenz-/ Zeitauflösung. Je schmaler die verwendete Fensterfunktion ist, desto besser ist die Zeitauflösung, desto schlechter aber die Frequenzauflösung.

5.3.2 Wavelet Transformation

Die Kurzzeit-FOURIER-Transformation bietet sich vor allem dann an, wenn sich die im Signal enthaltenen Frequenzen nicht stark unterscheiden, für einen bestimmten Frequenzbereich lässt sich eine optimale Fensterfunktion und Breite finden. Wenn sich die enthaltenen Frequenzen zu stark unterschieden, ist es schwierig, ein geeignetes Fenster zu finden, da die Zeit-Frequenzauflösung für alle Frequenzkomponenten fest durch das gewählte Fenster vorgegeben ist. Als Alternative dient die 1982 von Jean Morlet eingeführte kontinuierliche Wavelet-Transformation (CWT). Der mathematische Ausdruck zum Berechnen der Wavelet-Koeffizienten ist dem der STFT sehr ähnlich, allerdings werden die Fensterfunktion und die komplexe Exponentialfunktion durch die sogenannte *Wavelet-Funktion* ψ_0 ersetzt:

$$W_\psi(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t)\, \psi_0\left(\frac{t-\tau}{s}\right) dt \tag{12}$$

Das bedeutet, die Wavelet-Transformierte wird als inneres Produkt des Signals $f(t)$ mit verschobenen und skalierten Versionen einer einzigen Wavelet-Funktion ψ_0 berechnet. Der Vorfaktor $\frac{1}{\sqrt{s}}$ dient Normierungszwecken und sorgt dafür, dass alle betrachteten Signalabschnitte die gleiche Energie besitzen. Ein in der Datenanalyse häufig angewendetes Wavelet ist das komplexe MORLET-Wavelet, bei dem es sich im Wesentlichen um ein Produkt aus einer Gaußfunktion und einer komplexen Exponentialfunktion handelt:

$$\psi_0(t) = \frac{1}{\sqrt{\pi\gamma b}} e^{2\pi i F_c t}\, e^{-\frac{t^2}{\gamma_b}} \tag{13}$$

Wenn über die Parameter F_c und γ_b die „Grundform“ des Wavelets definiert wird, können die Wavelet-Koeffizienten für verschiedene Werte von τ, dem *Translationsparameter*, und s, dem *Skalenparameter*, berechnet werden. Die Variation von τ bedeutet eine Zeitverschiebung, so dass die Transformation für einen festen Wert *s* als Faltung des Signals $f(t)$ mit dem zeitlich gespiegelten Wavelet verstanden werden kann.

Der entscheidende Unterschied zur *STFT* liegt dabei darin, dass mit diesen Parametern nicht nur die Lage des Wavelets, welches jetzt die Rolle der Fensterfunktion einnimmt, während des Transformationsprozesses variiert werden kann, sondern über den Skalenparameter s auch dessen Form. Je größer der Wert von s ist, desto mehr wird das Wavelet gestreckt und

desto größer ist seine Breite. Man berechnet die Koeffizienten also nicht für unterschiedliche Frequenzen F sondern für unterschiedliche Skalen, siehe [Thuselt, et al., 2013].

Abbildung 5.10 soll das Translations- und Skalierungsverhalten des Realteils des komplexen *MORLET-Wavelets* ($F_c = 1$; $\gamma_b = 1$) bei verschiedenen Parametern τ und *s* verdeutlichen.

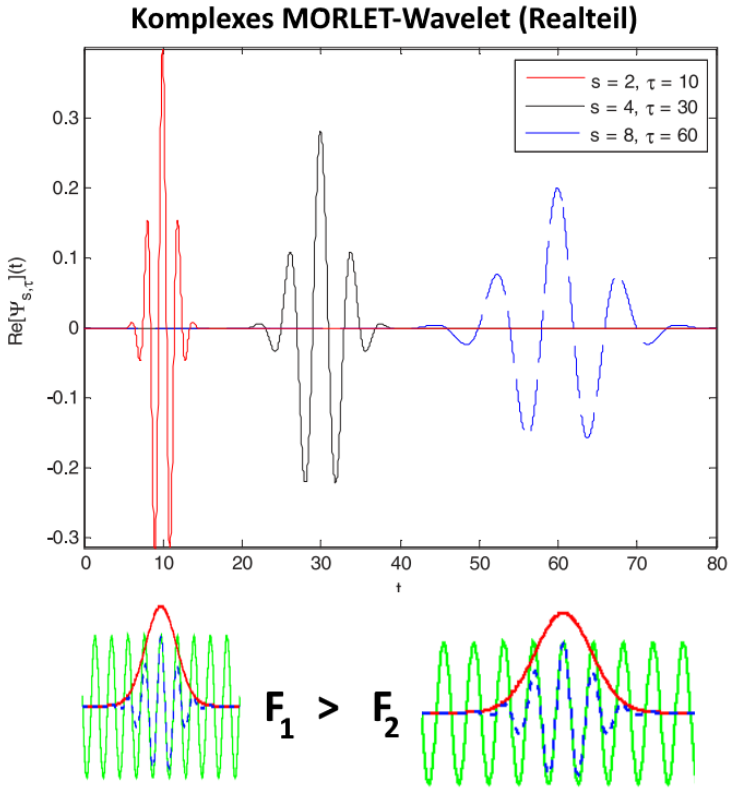


Abbildung 5.10: Skalierungs- und Translationsverhalten 'cmorl' [Thuselt, et al., 2013]

Ein großer Wavelet-Koeffizient bei hohem Skalenparameter deutet folglich auf eine niedrige Frequenzkomponente hin (F_2 bei $s = 8;\ \tau = 60$), ein großer Koeffizient bei niedrigem Skalenparameter hingegen auf eine hohe Frequenzkomponente (F_1 bei $s = 2; \tau = 10$).

Das Ergebnis der CWT ist ein Wavelet-Skalogramm, aus dem die Zeit und Frequenzinformationen herausgefiltert werden können. Für eine beispielhafte Anwendung der CWT wird ein Klangwechsel mit weit auseinander liegenden Tönen simuliert (264Hz u. 1056 sowie 220 Hz u. 880 Hz). Die Ergebnisse können mittels Matlab entweder über die *„Wavemenu-Toolbox“* oder gemäß *Anhang A* und Abbildung 5.11 über die Kommandozeile mittels *cwt-Funktion* berechnet werden.

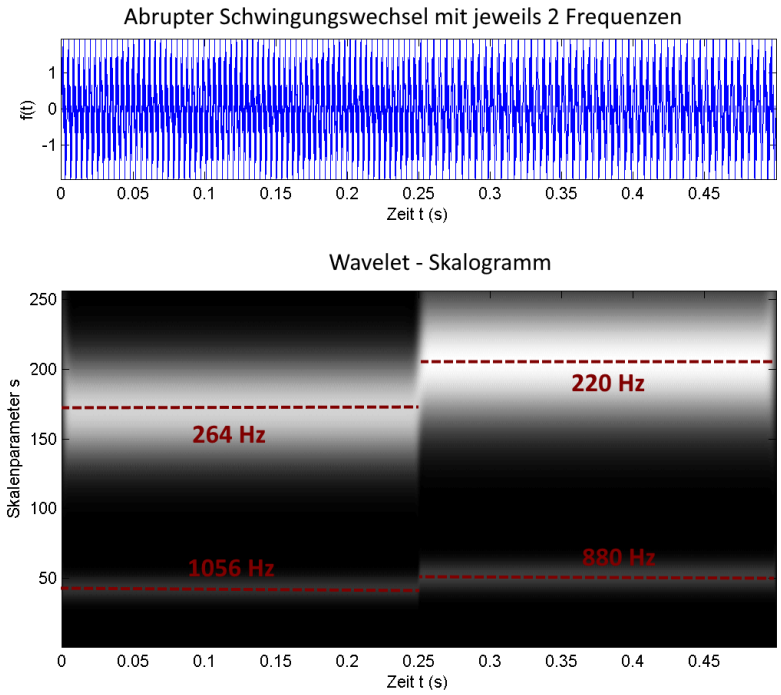


Abbildung 5.11: Wavelet Skalogramm

In dem Skalogramm werden die Koeffizienten auf die Zeit und die entsprechende Skala geplottet. Obwohl die Vermutung nahe liegt, lassen sich über die lokalen Koeffizientenmaxima keine Rückschlüsse auf die Amplituden der Schwingungen schließen. Der Zusammenhang zwischen Frequenzen und Skalenparameter ergibt sich durch Formel (14).

$$F_p = \frac{F_c * F_S}{s} \tag{14}$$

Hierbei ist F_p die Pseudofrequenz, F_c die CenterFrequenz, F_s die Abtastfrequenz und s der Skalenparameter.

Die Frequenz kann für einen bestimmten Skalenparameter, bei bekannter Wavlet-Funktion und Abtastfrequenz über die Funktion *„scal2freq“* bestimmt werden. Die Frequenzen für die Skalenparameter (42, 52, 171, 205) wurden der Abbildung 5.11 in roter Farbe hinzugefügt. Im weiteren Verlauf dieser Masterarbeit ist die Kurzeitfouriertransformation als Analysemethode ausgewählt worden, da die voerliegenden Frequenzen sich nicht in einem hohen Maße voneinander unterscheiden und eine *STFT* hier bessere Ergebnisse liefert.

6 Testmessungen der Sensoren

Ein Prototyp ist konstruiert worden, die mathematischen Algorithmen sind festgelegt und die Datenübertragung hergestellt worden. Nun können die ersten Testmessungen beginnen. Bevor die Parameter des Sensors aus dem Datenblatt verifiziert werden können, muss zuerst ein störungsfreier Betrieb gewährleistet sein.

6.1 Erste Tests und Plausibilitätsprüfung

Erste Tests sollten die Plausibilität der gewonnen Messdaten überprüfen. Es wurde mit den Parametern aus den Datenblättern der Einzelsensoren gearbeitet. Nach der richtigen Einstellung für die Werte der AD-Wandler konnte festgestellt werden, dass die Temperaturwerte in einem korrekten Bereich lagen. Eine Kalibrierung des Sensors entfällt vorerst.

Tabelle 6.1: Erste Temperaturmessungen

Zeitstempel	Temperatur in °C
1481641955.602	20.68
1481641955.755	20.72

Bei den Beschleunigungsmessern konnte festgestellt werden, dass die Werte der Datenblätter ungenau waren. So wird der Beschleunigungswert der Schwerebeschleunigung um den Faktor 1,12 falsch wiedergegeben.

Tabelle 6.2: Erste gespeicherte Beschleunigungswerte

Zeitstempel	Beschl. in X [m/s²]	Beschl. in Y [m/s²]	Beschl. in Z [m/s²]	Beschl. ges. [m/s²]
1481641955.602	-10.6312503	-1.64999997	-2.35312509	11,0
1481641955.736	-10.6687498	-1.64999997	-2.35312509	11,0
1481641955.868	-10.6593751	-1.64062500	-2.38124990	11,0
1481641956.002	-10.6218748	-1.64999997	-2.36249995	11,0

Daraufhin wurden die Werte des Sensors angepasst. Insgesamt lässt sich sagen, dass die ersten Testmessungen sehr zufriedenstellend waren und keine größeren Probleme auftraten.

6.2 Prüfung der Sensorparameter

Wichtige Kriterien für die Einsatzfähigkeit dieses Sensors sind Datenraten, Genauigkeit und Zuverlässigkeit. Eine hohe Datenrate gewährleistet eine Detektion auch von hochfrequenten Schwingungen. Des Weiteren kann durch Aufsummierung mehrerer aufeinanderfolgender Messwerte eine Genauigkeitssteigerung erfolgen. Die Genauigkeiten des Sensors sollen ebenfalls kritisch betrachtet werden. So kann festgelegt werden, im welchen Bereichen dieser Sensor eingesetzt werden kann. Eine weitere wichtige Größe ist die Zuverlässigkeit des Sensors. Hier soll auf eine korrekte zeitliche Äquidistanz geachtet werden. Auch Ausfälle des Sensorsystems sind sehr kritisch zu sehen, da für diese Zeitbereiche keine Messdaten vorliegen würden. Auf mögliche Ausreißer und Sprünge in den Messdaten muss eingegangen werden, da diese später nur mit Aufwand rechnerisch beseitig werden können.

6.2.1 Datenraten

Die Datenraten können nicht beliebig gesteigert werden. Durch den Einsatz des langsamen, aber sehr genauen AD-Wandlers *ADS1115* in Kombination mit dem Beschleunigungssensor *ADXL354* und dem *ADXL335* konnten nur Abtastfrequenzen bis maximal 30 Hz gewählt werden. Dies ist akzeptabel, da hier noch eine Genauigkeitssteigerung durch zeitliche Integration erfolgen kann. Werden die Werte nur mit 5 Hz benötigt, konnten die Rohdaten über die Anzahl von sechs Messungen integriert werden. So muss der Mittelwert aus diesen Messungen gebildet werden.

$$Mittelwert: \quad \hat{l} = \sum_{k=1}^{n} l_i$$

$$Standardabweichung\ des\ Mittelwertes: \sigma_{\hat{l}} = \sqrt{\frac{\sum_{k=1}^{n} v_i^2}{n * (n-1)}} \tag{15}$$

Dadurch lässt sich in diesem Fall eine Genauigkeitssteigerung von ca. 2,5 erzielen. Höhere Datenraten konnten mit dem AD-Wandler *ADS1115* nicht erhalten werden, nach dem Datenblatt des AD-Wandlers sollte dies aber möglich sein. Eine ausführliche Untersuchung hierüber war aufgrund des zeitlichen Rahmens nicht möglich. Die Abtastfrequenz von 30 Hz ist für die meisten Anwendungen ausreichend. Bei der Nutzung des *ADXL355*, der über einen integrierten AD-Wandler verfügt, konnten mit dem geeignetsten Mikrocontroller, dem *Blue*

pill, Datenraten bis zu 500 Hz erhalten werden. Dies ist ein sehr vielversprechender Wert, da somit Schwingungsfrequenzen von bis zu 250 Hz detektiert werden können.

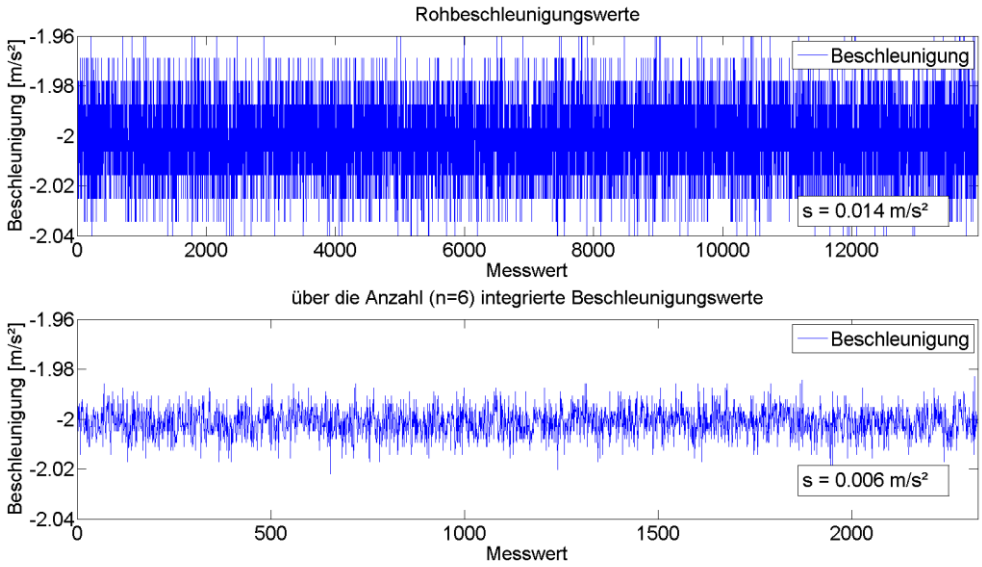


Abbildung 6.1: Erhöhung der Genauigkeit durch Mittelwertbildung (Integration über die Anzahl)

Gerade bei der Verwendung des *ADXL355* kann eine erhebliche Genauigkeitssteigerung durch die zeitliche Integration erhalten werden, da bei einer Mittelwertbildung der Frequenz von 500 Hz zu einer Frequenz von fünf Hz nach der Formel (15) eine Genauigkeitssteigerung mit dem Faktor zehn erhalten werden kann.

6.2.2 Genauigkeiten

Die Genauigkeiten der Sensoren sind über empirische Tests ermittelt worden und mit den Genauigkeiten aus den Datenblättern verglichen worden. Das Auslesen des Temperatursensors erfolgte über einen Zeitraum von einem Tag in einem klimatisierten Raum zusammen mit einem Referenzthermometer. Ausreißer innerhalb der Temperaturmessung sind nicht vorhanden. Das Rauschen der Einzelmessungen ist in Tabelle 6.3 angegeben. Der Temperatursensor ist bei beiden Beschleunigungssensoren, *ADXL354* und *ADXL355* identisch.

Tabelle 6.3: Genauigkeiten des Temperatursensors

Rauschen der Einzelmessung	0,2 °C
Temperatur des Prüflings	20,3 °C
Temperatur der Referenz	20,7 °C
Abweichung	0,4 °C

Die Beschleunigungsdaten sind ebenfalls geprüft worden. Hier wurden die Standardabweichungen der Einzelmessungen der drei Achsen ermittelt.

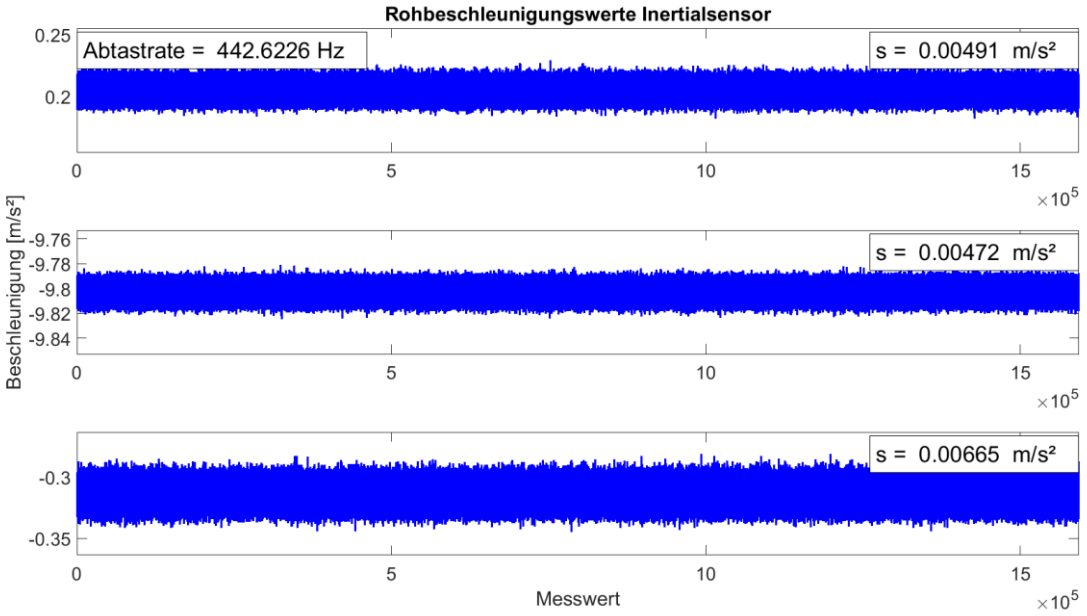


Abbildung 6.2: Standardabweichungen der drei Achsen des Beschleunigungssensors

Es fand eine Überprüfung aller drei getesteten Beschleunigungssensoren statt. Eine Auflistung der erhaltenen Werte findet sich in Tabelle 6.4.

Tabelle 6.4: Genauigkeiten der Beschleunigungssensoren

Standardabweichung	*ADXL335*	*ADXL354*	*ADXL355*
	[m/s²]	[m/s²]	[m/s²]
x-Achse	0,05	0,011	0,005
y-Achse	0,05	0,010	0,005
z-Achse	0,08	0,014	0,007

Es wird ersichtlich, dass die Messwerte der Z-Achse größere Ungenauigkeiten als die Messwerte der beiden anderen Achsen aufweisen. Dies ist bauartbedingt und hat die Ursache in der Konstruktion des Sensors. [Resnik, et al., 2013]

Aus der Tabelle 6.3 ist ersichtlich, dass der Beschleunigungssensor *ADXL355* die geringste Standardabweichung aufweist. Diese Tatsache, in Kombination mit der höchsten Abtastrate von ca. 500 Hz, lassen diesen Sensor als beste Wahl für die weitere Verwendung innerhalb

dieser Masterarbeit erscheinen. Daher wird im weiteren Verlauf nur dieser Sensor weiterverwendet.

6.3 Driftverhalten

Die meisten Beschleunigungssensoren haben vor allem während der Aufwärmphase nach der Inbetriebnahme einen hohen linearen Trend. Dieser hat zur Folge, dass in den ersten Minuten keine Messungen vorgenommen werden sollten.

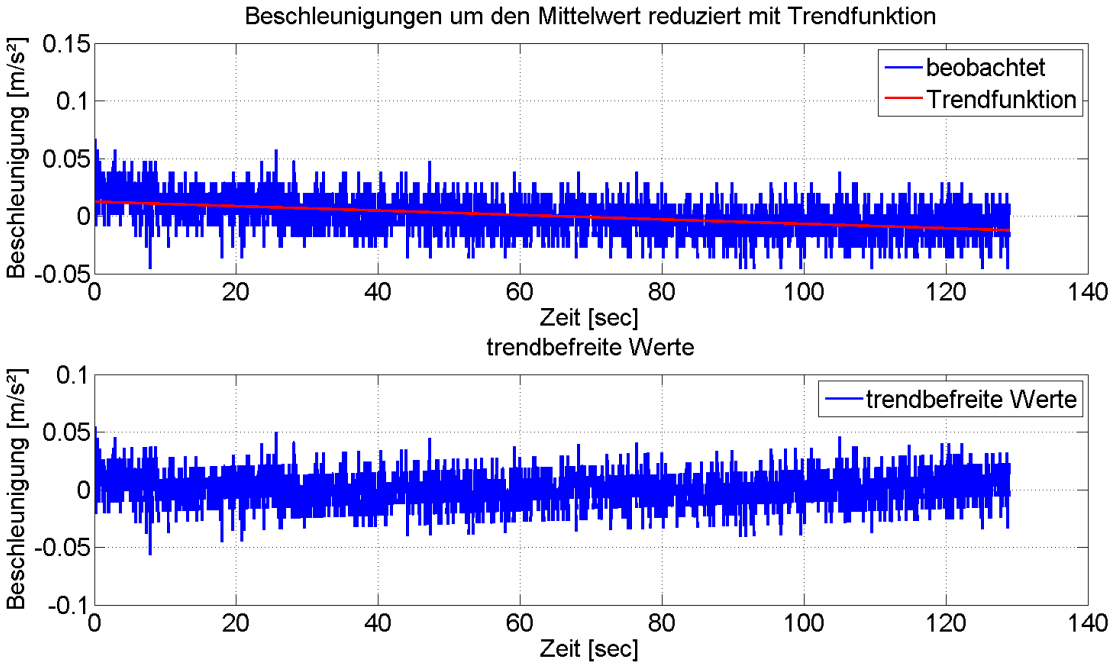


Abbildung 6.3: Drift des Sensors innerhalb der ersten 150 Sekunden

Im Laufe des Aufwärmvorganges kann sehr gut ein Driftverhalten beobachtet werden, welches sich mathematisch als linearer Trend nachweisen lässt. Nach ca. 120 Sekunden hat der Sensor seine Betriebstemperatur erreicht. Danach ist das Driftverhalten nicht mehr nachzuweisen. Bei weiteren Messungen ist daher der Sensor mindestens drei Minuten vor der Messung in Betrieb genommen worden.

6.4 Zuverlässigkeit

Die Zuverlässigkeit lässt sich am sichersten über eine Langzeitmessung ermitteln. Im Zeitraum von zehn Tagen wurde der Sensor ausgelesen, dabei ist vor allem auf den kompletten Ausfall und auf Datenlücken geachtet worden.

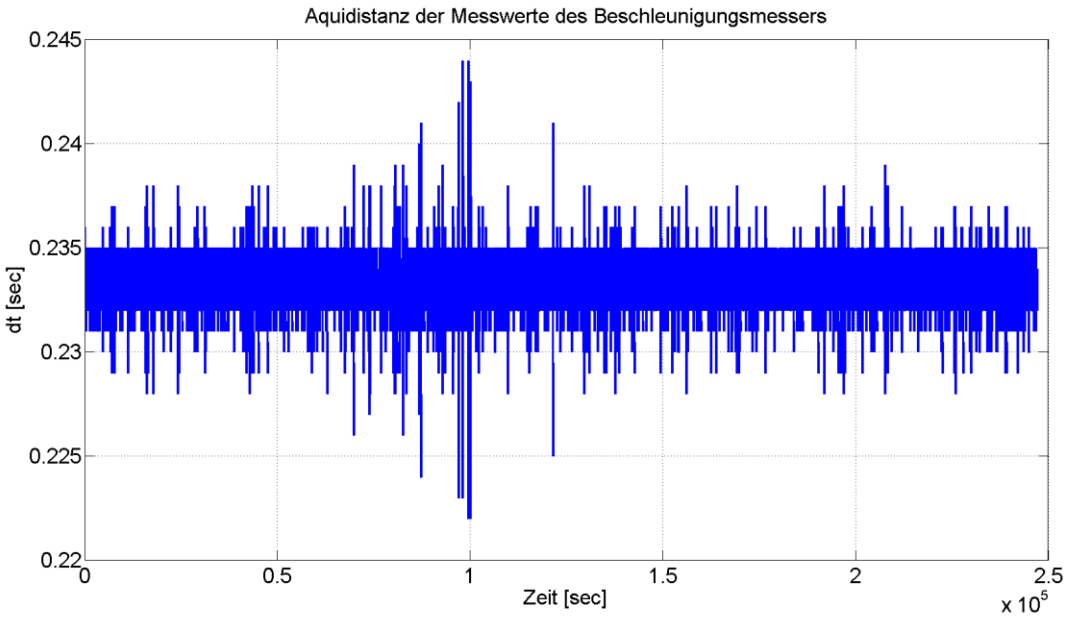


Abbildung 6.4: zeitliche Äquidistanz des Beschleunigungssensors

Aus Abbildung 6.4 ist ersichtlich, dass die zeitliche Äquidistanz mit einer maximalen Abweichung von 10 Millisekunden für den angestrebten Anwendungsbereich ein akzeptabler Wert ist. Die Messwerte werden in einem sehr gleichmäßigen Takt abgegriffen. Dadurch sollte eine gute Frequenzanalyse im weiteren Verlauf möglich sein. Falls Datenlücken vorhanden gewesen wären, könnten diese in der Auswertung mittels geeigneter Interpolationsverfahren beseitigt werden.

6.5 Test auf Detektion einer simulierten Bauwerksdeformation

In diesem Test soll die Eignung des Beschleunigungssensors auf die Detektion einer Bauwerksdeformation in Form einer Schwingung getestet werden. Dazu wurde die in der Bachelorarbeit “Multisensor-Monitoring zur Detektion hochfrequenter Schwingungen an der Marienkirche Neubrandenburg“ angefertigten Testvorrichtung zurückgegriffen.

Abbildung 6.5: Testvorrichtung zur Simulation von Bauwerksdeformationen

Die erzeugte schwingende Bewegung mit fester Amplitude und Frequenz sollte nachgewiesen werden. In Abbildung 6.6 sind die Rohdaten geplottet worden. Diese beobachtete Zeitreihe ist schon während des Einlesens zur besseren Darstellung um den Mittelwert reduziert worden und muss nun von eventuellen Trends befreit werden.

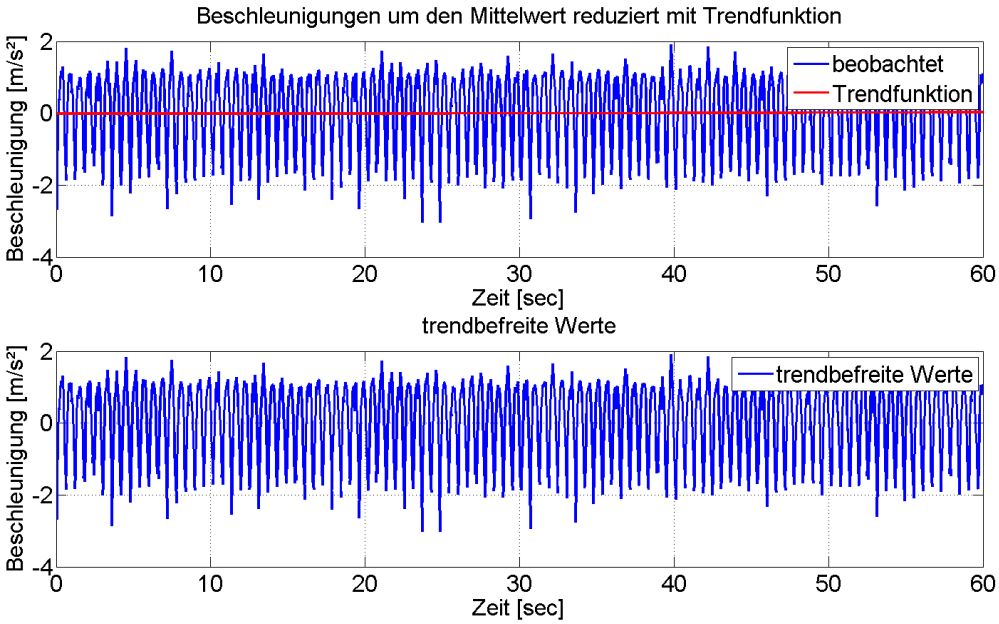


Abbildung 6.6: detektierte Schwingung

Durch die hohe Abtastrate und die hohe Messgenauigkeit lassen sich die Frequenz und die Amplitude schon aus dem Plot schätzen. Im nächsten Schritt wird die Autokorrelationsfunktion berechnet und geplottet, siehe Abbildung 6.7

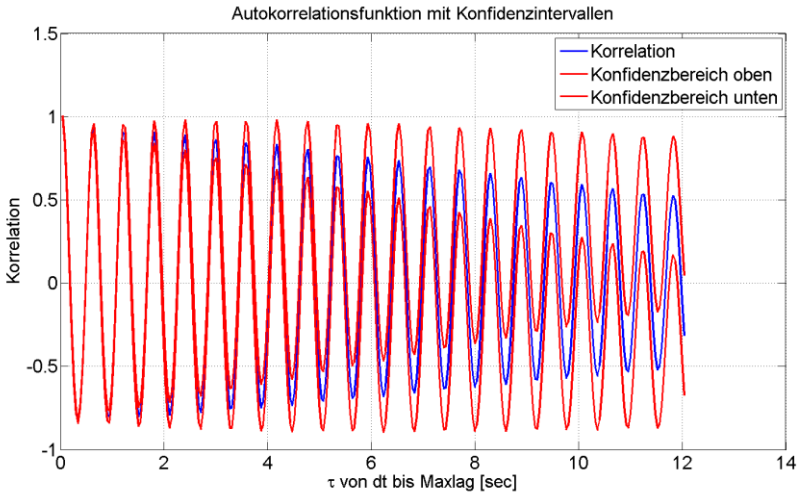


Abbildung 6.7: Autokorrelationsfunktion mit Konfidenzintervallen

Aus der Autokorrelationsfunktion ergibt sich die Frequenzanalyse, die in Abbildung 6.8 dargestellt ist. Hier ist zu sehen, dass die Frequenz und die Amplitude nachzuweisen sind.

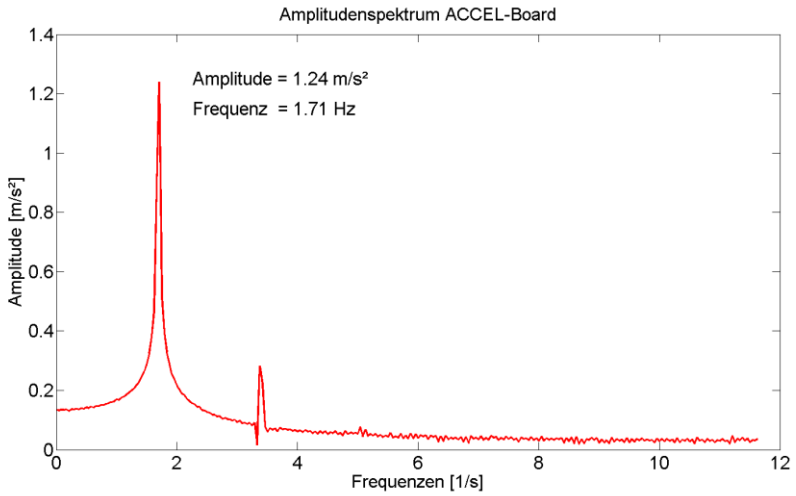


Abbildung 6.8: Amplitudenspektrum der Schwingung

6.6 Bestimmung der Amplitude der Auslenkung

Die Amplitude der Beschleunigung ist mit 1,24 m/s² bestimmt worden. Dieser Beschleunigungswert hat nur wenig praktischen Nutzen. Eine wichtigere Information ist hierbei die Amplitude der Auslenkung, also der Weg. Aus Beschleunigungswerten lässt sich durch doppelte Integration der Weg bestimmen. Somit muss die Sinusschwingung der Beschleunigung doppelt nach der Zeit integriert werden.

$$A(t) = A_B * sin(2\pi f t + \varphi) \quad (16)$$

A_B ist die Amplitude der Beschleunigung, f die Frequenz und φ die Phasenverschiebung.

Durch Integration von Formel (16) wird die Geschwindigkeit ermittelt, siehe Formel (17).

$$\int A\,dt = \int A_B * sin(2\pi f t + \varphi)\,dt = \frac{-A_B}{2\pi f} * cos(2\pi f t + \varphi) + C \quad (17)$$

Die Amplitude der Beschleunigung ist um den Faktor 2*π*f größer als die Amplitude der Geschwindigkeit. Die Integrationskonstante C muss berücksichtigt werden, da sie als Offset wirkt und die eigentliche Funktion verschiebt. Sie kann durch Zentrierung der erhaltenen Werte eliminiert werden. Es erfolgt im nächsten Schritt eine weitere Integration von der Geschwindigkeit nach der Zeit zum Weg.

$$\int\int A\,dt\,dt = \int\int A_B * sin(2\pi f t + \varphi)\,dt\,dt \quad (18)$$

$$\int\int A\,dtdt = \int \frac{-A_B}{2\pi f} * cos(2\pi f t + \varphi) + C dt = \frac{-A_B}{4\pi^2 f^2} * sin(2\pi f t + \varphi) + Ct + \tilde{C} \quad (19)$$

Aus Formel (19) wird ersichtlich das die Amplitude des Weges um $4*\pi^2*f^2$ kleiner ist als die Amplitude der Beschleunigung. So kann mit Kenntnis der Parameter der Schwingungsgleichung der Beschleunigung direkt die Amplitude des Weges bestimmt werden. In diesem Beispiel ist die Frequenz mit 1,71 Hz angegeben, somit ergibt sich aus den 1,24 m/s² ein Wert von 10,7 mm, da der Divisor einen Betrag von 115,3 s² aufweist. Der exakte Wert für die Amplituden beträgt 11 Millimeter, da die Testvorrichtung einen festen Arbeitsbereich hat. Der nachgewiesene Wert von 10,7 Millimeter weicht nur in sehr geringem Maße hiervon ab. Somit können von den Schwingungen sowohl die Frequenzen als auch die Amplituden nachgewiesen werden, siehe Abbildung 6.9.

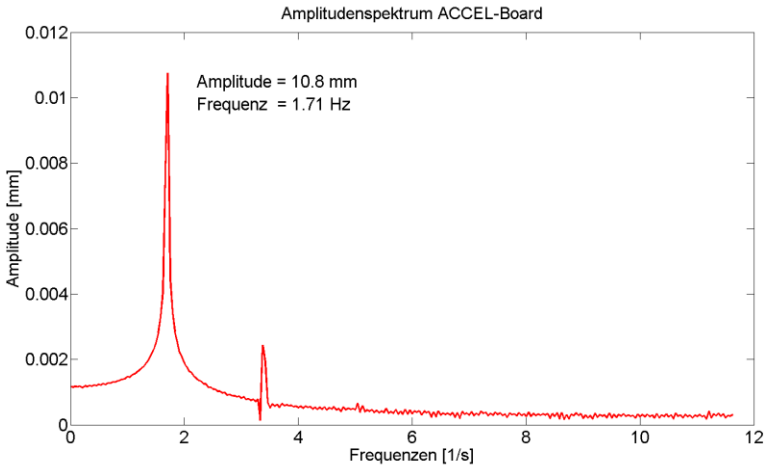


Abbildung 6.9: Amplituden- und Frequenzbestimmung mit dem Accel-Board

7 Kalibrierung

Kalibrieren, gemäß der DIN 1319, heißt die Messabweichung am fertigen Messgerät festzustellen. Es erfolgt kein technischer Eingriff in das Messsystem. Es soll die Abweichung der Anzeige des Messgerätes mit dem als richtig geltendem Wert festgestellt werden. Im Rahmen dieser Masterarbeit wird auf die Kalibrierung des entwickelten Sensors eingegangen. Wichtige Parameter finden Beachtung, wie die Temperaturabhängigkeit, die Abhängigkeit von der Eingangsspannung oder der Verglich mit einem Referenzgerät. Jeder MEMS-basierte Beschleunigungsmesser wird von seiner Umgebungstemperatur beeinflusst. Diese Abhängigkeit soll mittels von Kalibrierung über einen Temperaturbereich, der dem Bereich des zukünftigen Einsatzes abdeckt, festgestellt werden. Bei den Testmessungen im Kapitel 6.2 fiel auch eine Abhängigkeit der Messwerte von der verwendeten Eingangsspannung auf, dies wird auch betrachtet. Ebenso wird ein Vergleich mit einem Referenzgerät vorgenommen um die Messdaten zu überprüfen.

7.1 Kalibrierung des Temperatursensors

Da der Sensor auch im Außenbereich angebracht werden soll, muss der Temperatureinfluss im gesamten Arbeitsbereich kalibriert werden. Dies entspricht einem Temperaturbereich von -20 °C bis +60°C. Aus den Messwerten kann dann eine entsprechende Kalibrierfunktion berechnet werden. Diese Funktion wird dann im weiteren Verlauf direkt an die Rohtemperaturwerte angebracht, bevor Messwerte gespeichert werden.

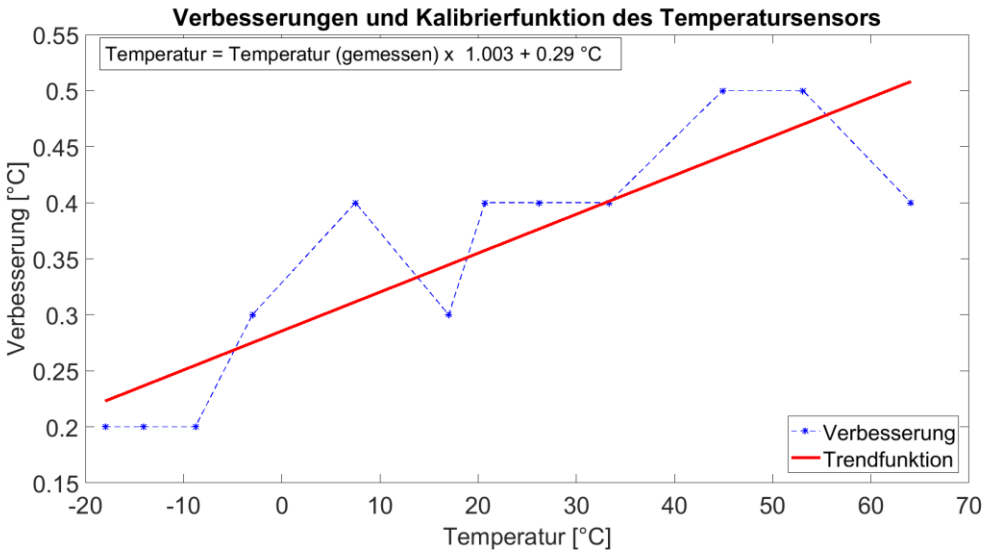


Abbildung 7.1: Kalibrierung des Temperatursensors

Der Prototyp wurde zusammen mit einem Referenzthermometer auf eine minimale Temperatur von ca. -19 °C abgekühlt. Dann sind schrittweise Erhöhungen der Temperatur bis ca. +63 °C erfolgt. Aus den Messwerten des Referenzthermometers und den Messwerten des Temperatursensors sind die jeweiligen Verbesserungen berechnet und in das Diagramm eingetragen worden. Aus den Verbesserungswerten sind ein Offset und eine Trendfunktion ersichtlich, die sich durch eine lineare Regression bestimmen lassen. Die Parameter der Regression sind in Abbildung 7.1 ersichtlich.

7.2 Kalibrierung des Temperaturabhängigkeit

Die Temperaturabhängigkeit spielt die größte Rolle. Wird diese nicht beachtet, kann bei z.B. täglichen Temperaturschwankungen eine Schwingung vorgetäuscht werden, die gar keine ist, sondern nur aus dem Tagesgang der Temperatur besteht. Daher muss eine Funktion des Zusammenhanges von Temperatur und Beschleunigungsmesswerten erstellt werden. Bei den verwendeten Beschleunigungssensoren *ADXL354* und dem favorisierten *ADXL355* ist direkt im Sensorgehäuse ein Temperatursensor installiert. Innerhalb des Sensors wird vom Hersteller aus die Temperatur an die Messwerte angebracht. Daher ist der Sensor werksseitig auf die Temperaturabhängigkeit kalibriert. In einem Testversuch ist bei Temperaturen von -19 °C bis +63 °C keine signifikante Änderung der Beschleunigungswerte festgestellt worden. Der Temperatursensor hat eine maximale Abweichung von der Solltemperatur von +0,5 °C. Daher sind die als Kalibrierung angebrachten Rohtemperaturwerte ausreichend genau. Im Rahmen dieser Masterarbeit und der Untersuchung von Schwingungen, die sich im Bereich von ca. einem Hertz bewegen, ist auf eine weitere Kalibrierung abgesehen worden. Soll der Sensor für Langzeitschwingungsanalysen mit Periodendauern von Tageslängen eingesetzt werden, ist eine weiterführende Analyse des Temperaturverhaltens erforderlich.

7.3 Kalibrierung der Abhängigkeit von der Eingangsspannung

Bei den ersten Tests mit dem *Raspberry Pi* als Auswerterechner ist aufgefallen, dass je nachdem welches Stromkabel zum Betrieb des *Raspberry Pi* verwendet wurde, die

Messwerte des *ADXL354* in Kombination mit dem *ADS1115* Analog-Digital-Wandler sehr unterschiedlich ausfielen. Die verwendeten Ladegeräte weisen unterschiedliche Ladespannungen auf, dies konnte in direkter linearer Abhängigkeit genauso in den Messwerten gesehen werden. Hier ist eine Festspannungsversorgung des Ad-Wandlers erforderlich. Auf diese Umrüstung ist aber verzichtet worden, da im weiteren Verlauf der Untersuchungen die analogen Sensoren nicht weiter verwendet wurden. Bei dem digitalen Beschleunigungssensor *ADXL355* mit dem integrierten AD-Wandler waren Abhängigkeiten von der Eingangsspannung im entsprechen Arbeitsbereich, von 2,5 V bis 5V, nicht festzustellen.

7.4 Kalibrierung der Ausgangswerte auf Normalschwere

Bei den Testmessungen im Kapitel 6.1 ist festgestellt worden, dass die herstellerseitig vorgeschlagenen Berechnungswerte der Beschleunigung um ca. den Faktor 1,12 abweichen. Die Beschleunigungsmesswerte sollen nun auf Normalschwere kalibriert werden. Die Untersuchungen finden innerhalb der Hochschule Neubrandenburg statt. Daher sollen die Messwerte auf den Normalschwerewert von Neubrandenburg korrigiert werden.

$$\gamma_0 = 9{,}78032677 * (1 + 0{,}0052790414 * sin^2\,\varphi + 0{,}0000232718 * sin^4\,\varphi + 0{,}0000001262 * sin^6\,\varphi \left[\frac{m}{s^2}\right] \quad (20)$$

$$\gamma(h) = \gamma_0 + \left(\frac{\partial\gamma}{\partial h}\right)_0 * h + \left(\frac{\partial^2\gamma}{\partial h^2}\right)_0 * h^2$$

$$mit \left(\frac{\partial\gamma}{\partial h}\right)_0 = -(0{,}30877 - 0{,}00043 * sin^2\,\varphi) * 10^{-5} \left[\frac{1}{s^2}\right] \quad (21)$$

$$und\ mit \left(\frac{\partial^2\gamma}{\partial h^2}\right)_0 = 1{,}44 * 10^{-12} \left[\frac{1}{m * s^2}\right]$$

In Formel (20) und (21) ist die Berechnung des Normalschwerewertes für das GRS80 angegeben. Die berechnete Normalschwere für die Hochschule Neubrandenburg mit einer ellipsoidischen Breite von 53° 33‘ und einer Höhe über dem GRS80 Ellipsoid von 53 Meter ergibt einen Normalschwerewert von 9,8136 m/s². Eine Berechnung des Schwerewertes mit dem Onlinerechner des Bundesamtes für Kartografie und Geodäsie ergibt eine Abweichung von 0,1 mm/s². Die Standardabweichungen einer Einzelmessung des

Beschleunigungssensors liegen bei ca. 5 mm/s² bei einer Abtastrate von ca. 500 Hertz. Daher ist eine Anpassung der Messwerte auf eine Beschleunigung bei Ruhe von 9.8136 m/s² ausreichend. Die Herstellerangaben der Umrechnungsfaktoren von den Rohmessdaten in Beschleunigungen des Beschleunigungssensors müssen um ca. 12 % verändert werden um die Normalschwere zu erreichen.

7.5 Vergleich mit einem Referenzgerät

Die Beschleunigungsdaten sind im Bezug zu einem Referenzmessgerät geprüft worden. Hier ist neben dem Rauschen der Einzelmessung, siehe Kapitel 6.2.2, die Abweichung zu einem Referenzgerät ermittelt worden.

Tabelle 7.1: Genauigkeiten des Beschleunigungssensors ADXL355 im Vergleich zum SBG-IG500E

	Achse X in [m/s²]	Achse Y in[m/s²]	Achse Z in [m/s²]
Rauschen des Prüflings (100 Hz)	0,0025	0,0023	0,0033
Rauschen der Referenz (100 Hz)	0,0020	0,0020	0,0030
Beschl. des Prüflings	0,211	9,800	-0,315
Beschl. der Referenz	0,212	9,803	-0,317
Abweichung	0,001	0,003	0,002

Hier wird deutlich dass die Genauigkeiten des Prüflings (*ADXL355*) nur gering unterhalb des Referenzgerätes (*SBG- IG500E*) liegen. Somit ist dieses Board für den Einsatz im geodätischen Monitoring bezüglich der Genauigkeit geeignet.

8 Messung an der Marienkirche als Referenzobjekt

Im Rahmen dieser Masterarbeit ist der Entwurf, die Konstruktion und der Einsatz eines auf Mikrosystemen basierenden Low-Cost-Sensors realisiert worden. Um die Einsatzfähigkeit des Sensors zu validieren, müssen die Ergebnisse des Sensors mit den Ergebnissen von anderen Messsystemen verglichen werden. So sind zeitgleich zu den Messungen mit dem neuentwickelten Sensor ebenfalls Messungen mit geodätischen Messgeräten, wie GNSS und Tachymeter, auch Neigungsmesser und Beschleunigungsmesser ausgeführt worden. Die Auswertung erfolgte mittels einer Schwingungsanalyse durch eine Fourier-Transformation, um von einer Zeitdarstellung in die Frequenzdarstellung zu gelangen. Zwei Mess- oder Rechengrößen sind in diesem Rahmen besonders wichtig. Die Frequenzen und Amplituden der Schwingung sollen mit allen Sensoren erfasst werden. Größte Beachtung sollte den Amplituden der Schwingungen gewidmet werden, da die Standsicherheit des Turmes in besonderem Maße von der Größe der Auslenkung abhängt. So muss aus der Frequenzdarstellung einwandfrei auf die Amplituden geschlossen werden können. Hierbei erweisen sich die tachymetrischen Messungen und die Messungen mittels GNSS am einfachsten, da die Messgrößen hier schon in Längenmaßen vorliegen. Die Beschleunigungssensoren hingegen liefern Daten der linearen Beschleunigung. Diese müssen mittels doppelter Integration in Längenmaße umgewandelt werden. Besonders schwierig erweist sich die Berechnung der Auslenkung bei den Neigungssensoren. Da hierbei Messwerte der Neigung, also Winkelwerte gewonnen werden, müssen diese ebenfalls in Längenmaße umgewandelt werden. Üblicherweise wird die Bogenformel verwendet, dabei wird unter der Voraussetzung gearbeitet, dass die Verformungslinie des Objektes eine Gerade ist. Hier müsste erst ein Modell zur Geometrie der Verformungslinie aufgestellt werden. Dies ist im Rahmen dieser Untersuchung nicht vorgesehen. So können die Amplitudenwerte der Neigungssensoren immerhin mit den Amplituden der Messungen der Beschleunigungsmesser verglichen werden, da die Sensoren jeweils auf gleicher Höhe angebracht wurden.

8.1 Das Messobjekt

Die Marienkirche Neubrandenburg gilt mit ihrem Ostgiebel als herausragendes Beispiel für

die norddeutsche Backsteingotik. Die Marienkirche war bis zu ihrer Kriegszerstörung eine neunjochige und dreischiffige Hallenkirche. Zum Ende des 2. Weltkrieges fielen rund 80 % der Innenstadt von Neubrandenburg einem Brand zum Opfer und auch die Marienkirche wurde stark zerstört. Im Jahre 1983 wurde der Kirchturm wiederhergestellt und erst im Jahre 2007 ist die Installation eines neuen Glockenstuhls abgeschlossen worden. Nach der Inbetriebnahme der fünf neuen Glocken wurden große Gebäudeschwingungen während des Glockenspiels durch das Betriebspersonals wahrgenommen. Diese Schwingungen machten eine Untersuchung der Standsicherheit des Kirchturmes erforderlich. So wurde im Rahmen eines Kooperationsprojektes zwischen der Hochschule Neubrandenburg und der Stadt Neubrandenburg die Überwachung des Turmes begonnen. So standen zur Validierung des neuentwickelten Sensors ein geeignetes Messobjekt und verschiedene Vergleichssensoren zur Verfügung.

8.2 Messaufbau

Die Messinstrumente sind über den Kirchturm verteilt angebracht, siehe Abbildung 8.1.

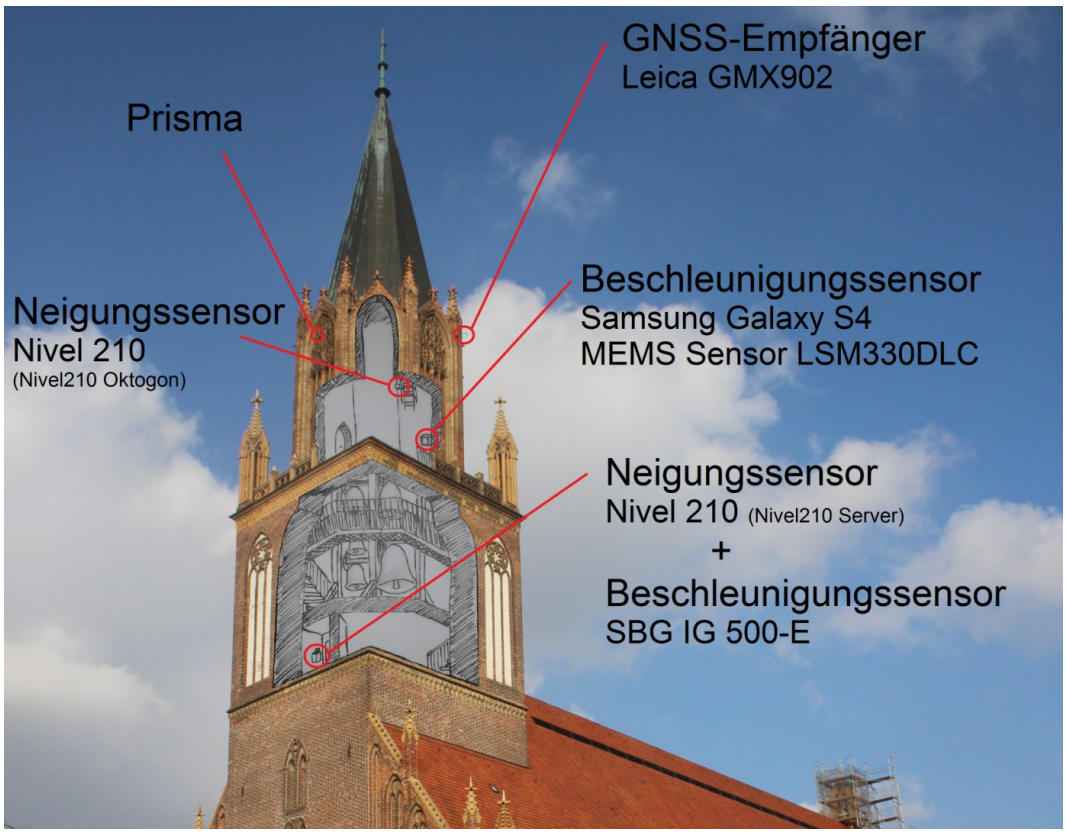


Abbildung 8.1: Messaufbau

Zur Erfassung der Turmschwingungen mittels GNSS sind zwei Zweifrequenzempfänger Leica GMX902 mit dazugehöriger Antenne verwendet worden. Mit der Software Leica Spider konnte eine Koordinatenberechnung in Echtzeit erfolgen. Die Empfänger konnten mit einer Abtastfrequenz von 20 Hz arbeiten. Mittels zwei Tachymetern (Leica TS30 und Leica TCRP1203) sind Reflektoren mit einer Abtastrate von ca. fünf Messungen pro Sekunde verfolgt worden. Ein Tachymeter konnte die Kirchturmschwingung in Längsrichtung und das andere Tachymeter in Querrichtung verfolgen. Mit zwei Präzisionsneigungsmessern Leica Nivel 210 wurde an verschiedenen Orten mit einer Abtastrate von 3 Hz die Neigung des Turmes gemessen. Die Auswertung ergab die rauschärmsten Ergebnisse, da die Leica Nivel 210 sehr hochauflösende Daten liefern. Die so erhaltenen Neigungen konnten aber nicht in eine metrische Auslenkung umgerechnet werden, da die Geometrie der Biegelinie des Turmes unbekannt ist. Aus dem Consumer-Bereich wurde ein Smartphone als Beschleunigungsmesser verwendet. Mit einer Abtastrate von 50 Hz sind Beschleunigungen in drei Achsen aufgezeichnet worden. Als zweiter Beschleunigungssensor ist ein IG-500E der Firma SBG Systems verwendet worden. Dieser wurde unterhalb des Glockenstuhls angebracht. Die Datenaufzeichnungsrate lag bei 50 Hz. Ebenfalls unterhalb des Glockenstuhls ist der Einbauort des neuentwickelten Sensors. Die Aufzeichnungsrate lag bei 400 Hz. Das Leica Nivel 210, der Beschleunigungssensor der Firma SBG Systems und der neuentwickelte Sensorzeichneten die Messdaten auf einer Ebene auf, wie in Abbildung 8.2 ersichtlich ist.

Abbildung 8.2: Sensoraufbau unterhalb des Glockenstuhls

8.3 Auswertung und Vergleich mit den anderen Messsystemen

Zuerst soll eine Übersicht aller Glockenfrequenzen und aller Gebäudefrequenzen, die gemessen wurden, in Tabellenform gegeben werden. Zur besseren Übersicht sind die gemessenen Frequenzen mit Nummern, z.B. 1 versehen worden. Diese Nummerierung ist auch in den folgenden Abbildungen zu finden.

Tabelle 8.1: Zusammenstallung gemessener Glocken- und Gebäudefrequenzen

Glocke	Glockenfrequenz [Hz]	Gebäudefrequenz [Hz]	Faktor
Marienglocke	0,38 6	1,14 1	3
Johannesglocke	0,42 7	1,26 2	3
Gebetsglocke	0,46 8	1,40 3	3
Sakramentsglocke	0,47 9	1,42 4	3
Lob- und Dankglocke	0,50 10	1,50 5	3

Mit allen Messmethoden konnten Schwingungen am Kirchturm nachgewiesen werden.

Die Messung mit dem Leica GMX 902 ergaben verschiedene Frequenzen und Amplituden, wie in Abbildung 8.3 ersichtlich ist.

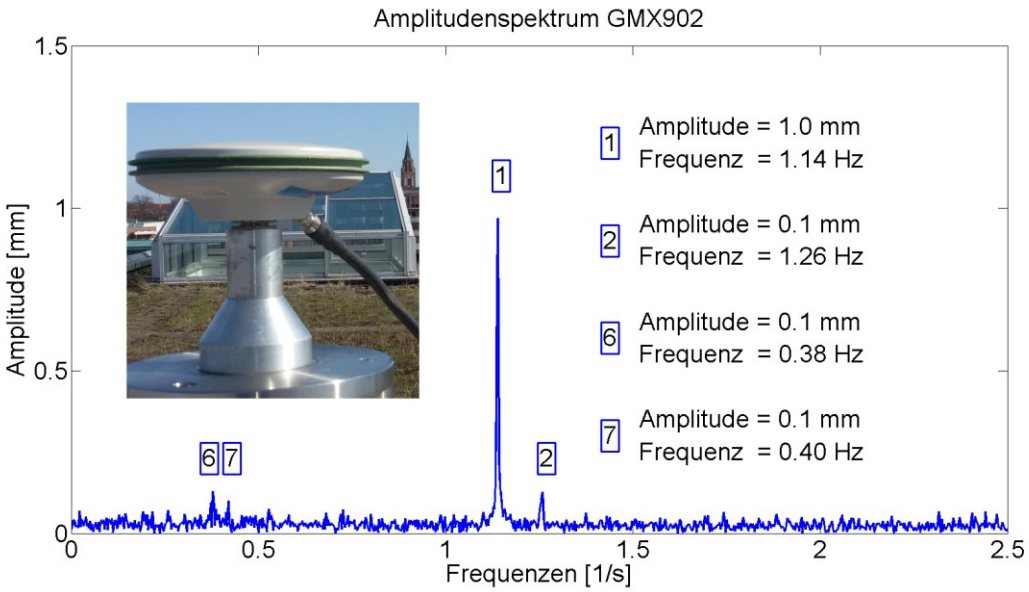


Abbildung 8.3: Frequenzanalyse GNSS

Mit dem quer zur Bewegungsrichtung angebrachten Tachymeter Leica TS30 konnten ebenfalls verschiedene Frequenzen und Amplituden gemessen werden, siehe Abbildung 8.4 .

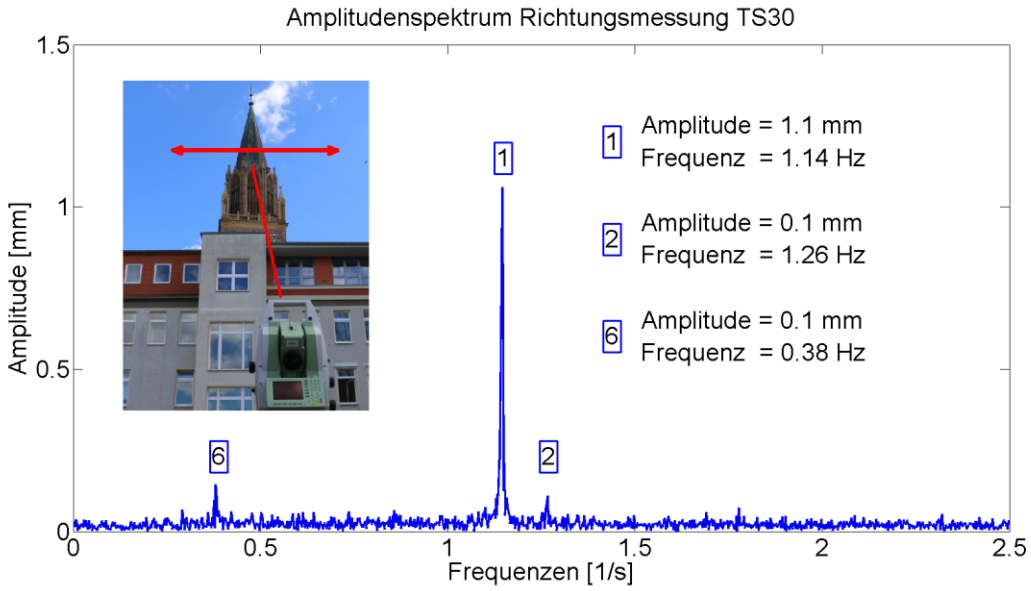


Abbildung 8.4: Frequenzanalyse Tachymeter Leica TS30

Mit dem Leica Nivel 210 auf der Glockenstuhlebene sind folgende Frequenzen und Amplituden gemessen worden, siehe Abbildung 8.5 .

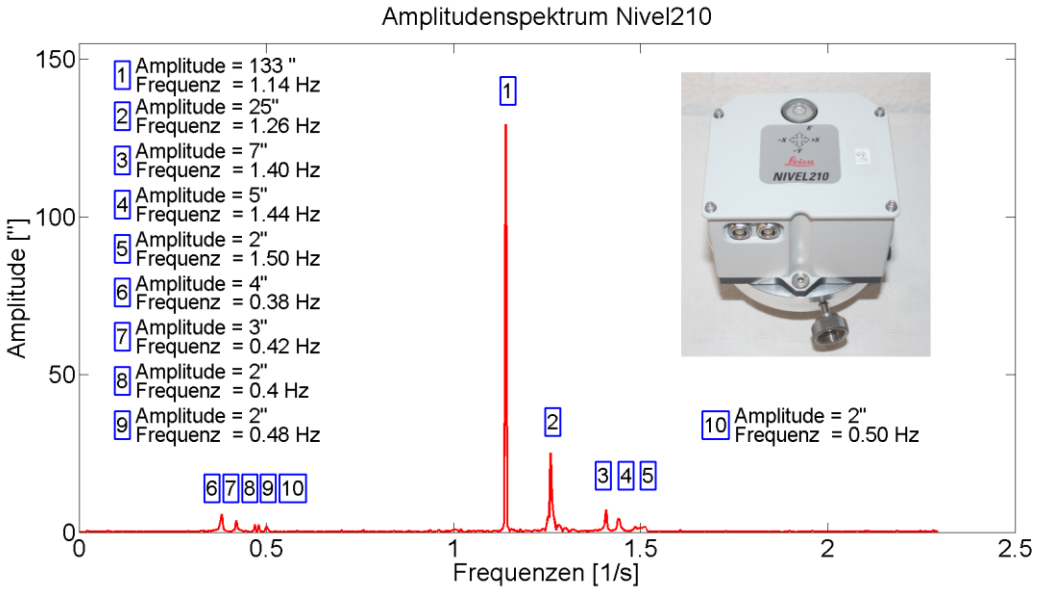


Abbildung 8.5: Frequenzanalyse Neigungssensor Leica Nivel210

Das Nivel liefert die am besten aufgelösten Messdaten. In Abbildung 8.5 können alle zehn Frequenzen, bestehend aus fünf Glockenfrequenzen und fünf Gebäudefrequenzen abgelesen werden.

Der Beschleunigungssensor der Firma SBG Systems ist direkt neben dem Leica Nivel 210 angebracht worden. Dieser liefert folgende Ergebnisse, die in Abbildung 8.6 ersichtlich sind.

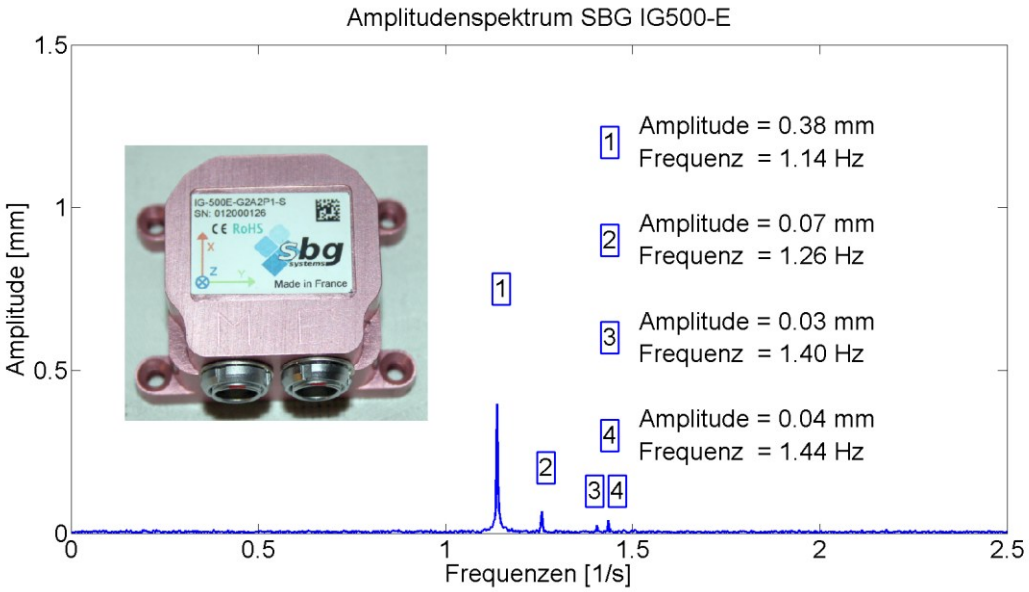


Abbildung 8.6: Frequenzanalyse SBG

Der Auswertung des neuentwickelten Sensors ergab sehr ähnliche Daten, siehe Abbildung 8.7 .

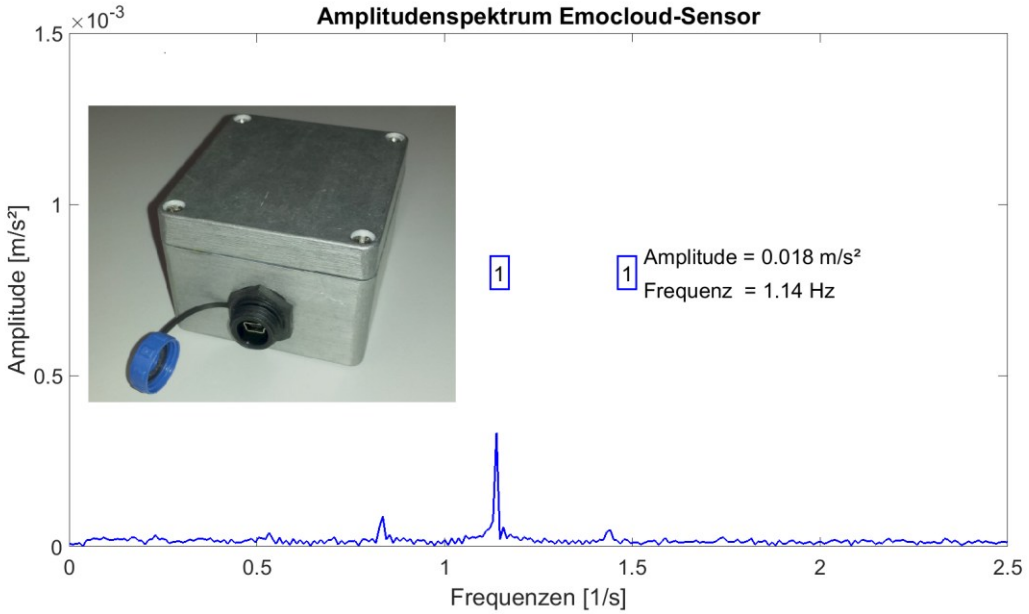


Abbildung 8.7: Frequenzanalyse des neuentwickelten Sensors beim Läuten der Marienglocke

Diese Messung erfolgte allerdings mit einer läutenden Glocke, der Marienglocke. Die Frequenz (1,14 Hz) und die Amplitude (0.3 mm) konnten festgestellt werden. Exemplarisch wird noch eine zweite Messung gezeigt. Bei dieser Messung läutete wieder nur eine einzelne

Glocke, hier die Gebetsglocke.

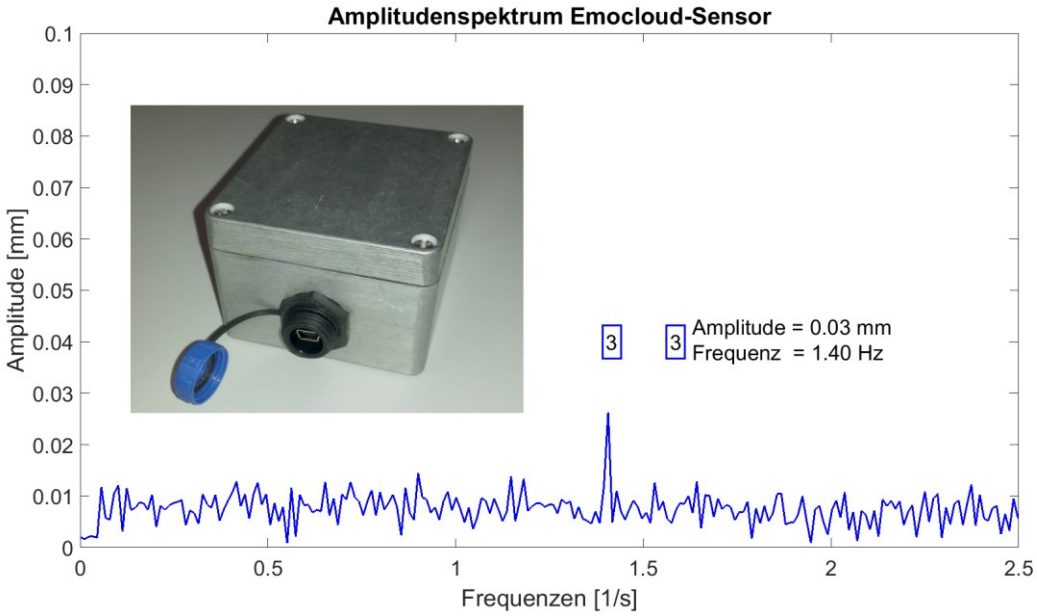


Abbildung 8.8: Amplitudenspektrum der Gebetsglocke

Auch die Einwirkung der Gebetsglocke auf das Mauerwerk mit einer Frequenz von 1,4 Hertz und einer Amplitude von 0.03 Millimeter wurden festgestellt. Damit konnte die Einsatzfähigkeit des neuentwickelten Sensors validiert werden.

9 Treppenüberwachung in der Hochschule Neubrandenburg

Als Testobjekt des neuentwickelten Sensors ist eine Treppe im Eingangsbereich des Hauses 2 der Hochschule Neubrandenburg ausgewählt worden. An diesem Messobjekt soll zuerst eine Dauermessung des Sensors erfolgen. Da sich das zu überwachende Objekt im Eingangsbereich der Hochschule befindet, können anfallende Modifizierungen des Sensors und der Auswertegeräte schnell und problemlos erfolgen. Die Dauermessung ist zur Bestimmung der Zuverlässigkeit des Sensors notwendig, da eventuelle Ausfallzeiten des Sensors oder Datenlücken in der Messaufzeichnung möglich sind. Diese Datenlücken sind im späteren Einsatz des Sensors zu eliminieren oder möglichst klein zu halten. Bei der Dauermessung soll auch auf auftretende Ausreißer in den Messelementen geachtet werden, da diese eine spätere Auswertung erschweren. Bei Ausreißern und bei Datenlücken können Ursachen ermittelt und behoben werden.

9.1 Das Messobjekt

Bei dem zu überwachenden Objekt handelt es sich um eine freitragende Treppe, die ca. acht Meter lang und ca. 4 Meter hoch ist.

Abbildung 9.1: Treppe im Eingangsbereich des Hauses 2 der Hochschule Neubrandenburg

Die Treppe im Eingangsbereich des Hauses 2 hat im oberen Bereich keine mechanische

Verbindung zum Fußboden des ersten Stockes. Durch die freitragende Konstruktionseigenschaft kann die Treppe bei Belastung durch Personen, welche die Treppe benutzen, in Schwingung geraten. Das Schwingverhalten der Treppe kann mit dem neuentwickelten Sensor auf einfache Weise überwacht werden.

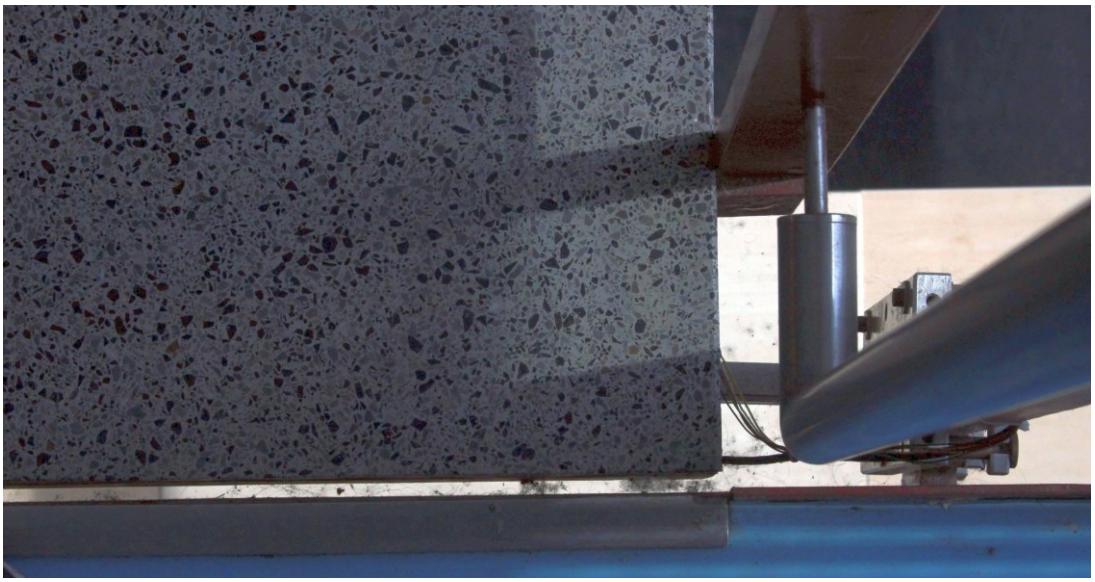


Abbildung 9.2: freitragende Konstruktion der Treppe

9.2 Messaufbau

Der Sensor ist direkt an der Treppe am obersten Stufenelement angebracht worden. In dieser Position kann der Sensor die Schwingung der Treppe in vollem Umfang mit ausführen.

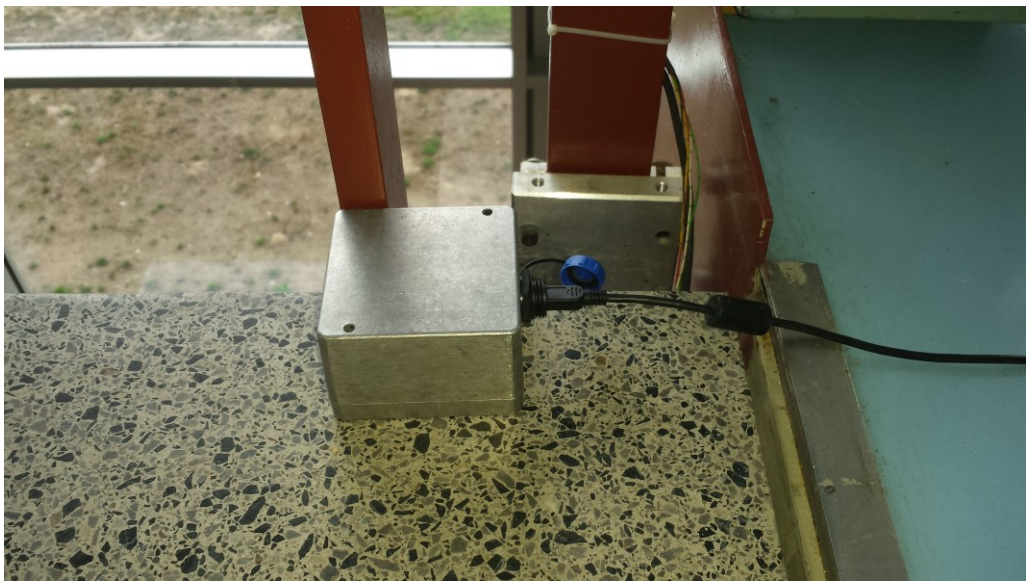


Abbildung 9.3: Anbringung des Sensors am obersten Stufenelement

Über einen Zeitraum von 7 Tagen ist die Treppe mit dem Sensor überwacht worden.

9.3 Auswertung

Aus der siebentägigen Messung sind hier einige Zeitabschnitte ausgewählt worden. In Abbildung 9.4 ist ein Zeitraum von ca. 10 Stunden, beginnend am Sonntag, den 19.08.2018 um fünf Uhr morgens, dargestellt.

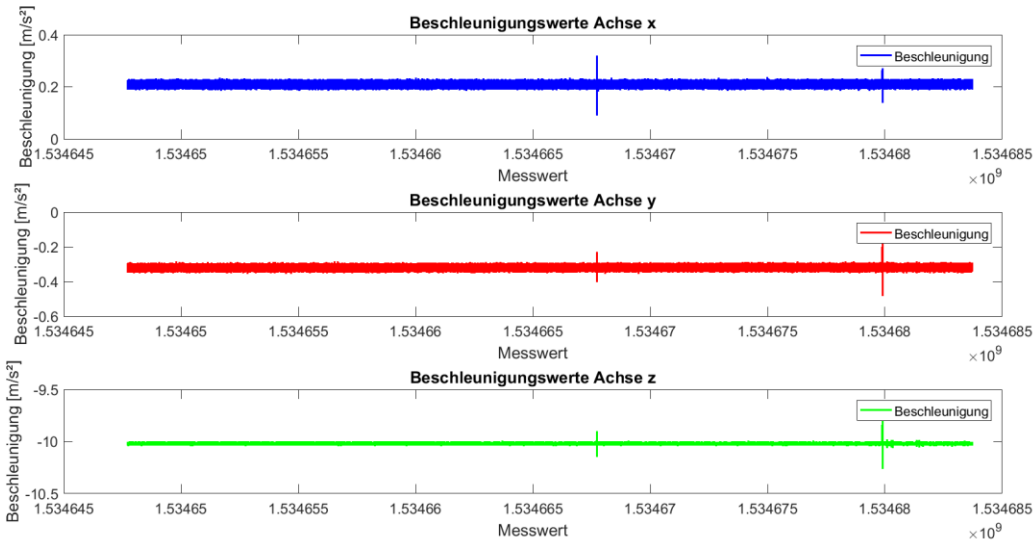


Abbildung 9.4: Messdaten der Treppe vom Sonntag, den 19.08.2018

Über einen Zeitraum von ca. 10 Stunden sind nur zwei Ereignisse zu erkennen. Um 12:10 Uhr und um 13:30 Uhr fanden Bewegungen auf der Treppe statt. Die übrige Zeit ist mit dem Grundrauschen versehen.

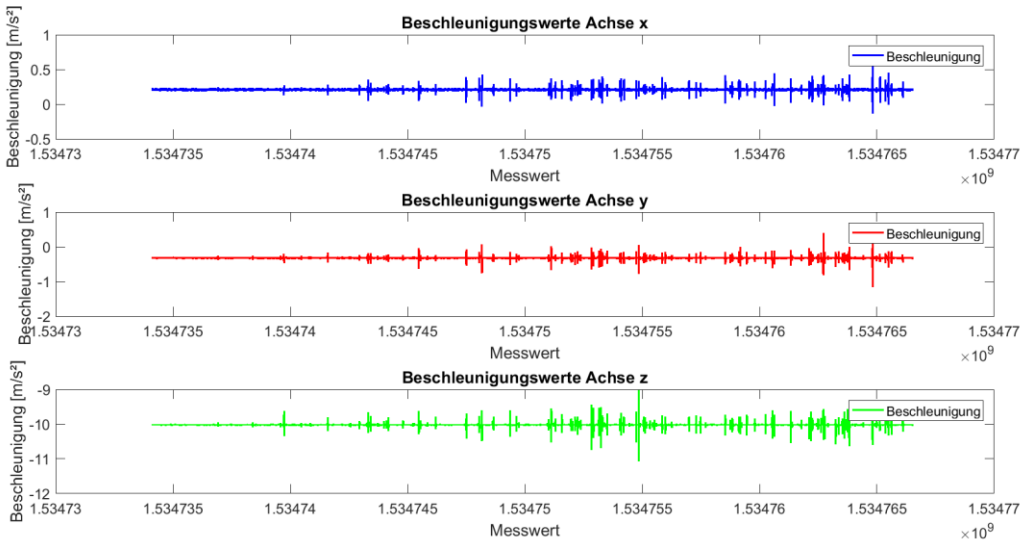


Abbildung 9.5: Messwerte der Treppe am Montag, den 20.08.2018

In Abbildung 9.5 sind insgesamt 89 Ereignisse im Unterschied zu den zwei Ereignissen in Abbildung 9.4 zu erkennen. Der zweite Auswertetag ist Montag, der 20.08.2018, an dem die Hochschule wieder geöffnet ist. Der Temperaturverlauf während der Messung ist in Abbildung 9.6 zu sehen. Interessant ist die Konstanz der Temperatur in den ersten drei Stunden. Erst um ca. acht Uhr morgens werden die Fenster und Türen im Eingangsbereich des Hauses 2 geöffnet. Gut zu sehen ist, dass der Temperaturverlauf nach der Kalibrierung keinen Einfluss auf die Messwerte hat.

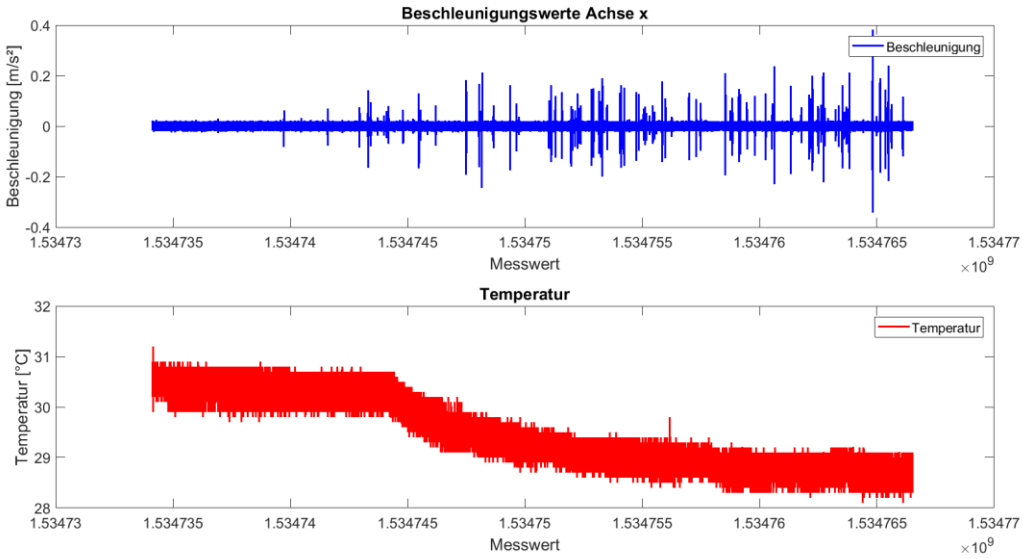


Abbildung 9.6: Messwerte einer Achse und Temperaturverlauf für den 20.08.2018

Nachdem die Messwerte über mehrere Stunden und Tage ausgewertet sind sollen nun einzelne Ereignisse näher untersucht werden. Das in der x-Achse dargestellte Ereignis mit der größten Amplitude soll weiter untersucht werden.

Hierbei handelt es sich um eine Schwingung der Treppe, die durch Erschütterung ausgelöst wurde. Da es sich um eine freitragende Treppe handelt, gerät sie vor allem durch Personenbewegungen im oberen Bereich in Schwingung. Diese Ereignisse lassen sich anhand Abbildung 9.6 erkennen. Am Montag, den 20. August 2018, sind im zehnstündigen Auswertezeitraum 89 Ereignisse beobachtet worden. Ein Ereignis besteht aus einer ca. 15 Sekunden andauernden Phase einer gedämpften Schwingung. In Abbildung 9.7 ist zu sehen, dass die Treppe von oben her betreten wurde, da die Schwingung mit der höchsten Amplitude beginnt. Bei einer Betretung von unten würden die Messwerte eine spiegelverkehrte Darstellung ergeben. Die Kurzzeitfouriertransformation zeigt einen

Frequenzverlauf ab der elften Sekunde. Ein Peak ist bei ca. zwei Hertz und ein weiterer ist zwischen fünf Hertz und vier Hertz. Nach ca. sechs Sekunden beginnt die Schwingung abzuklingen.

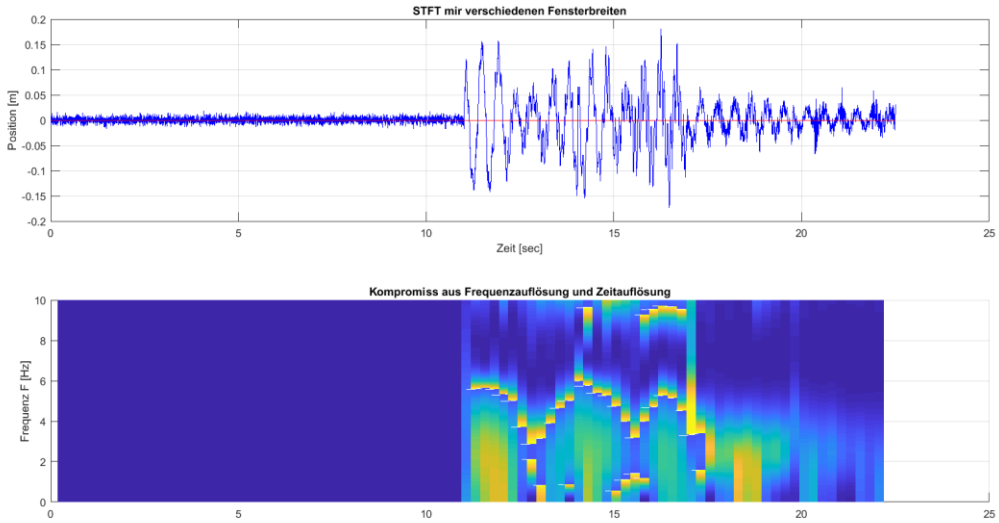


Abbildung 9.7: Erschütterung der Treppe und Kurzzeitfouriertransformation

Gut erkennbar ist, dass der Startzeitpunkt der Schwingung mit der Kurzzeitfouriertransformation gut festgestellt werden kann und das sich die Frequenzen je nach dem Ort der Belastung auf der Treppe ändern.

Bei diesem Messobjekt liefert der neuentwickelte Sensor sehr gute Ergebnisse. Die Untersuchung der Treppe ist ein Erfolg.

10 Bauwerksmonitoring in Waren/Müritz

In Zusammenarbeit mit dem ÖbVI Herrmann in Waren (Müritz) ist mit dem neuentwickelten Sensor eine Überwachungsmessung des Vermessungsbüros durchgeführt worden. Hier wurde der Sensor in einem realen Umfeld unter Arbeitsbedingungen eingesetzt, um Erkenntnisse über das Bewegungsverhalten des Gebäudes, in dem sich das Vermessungsbüro befindet, zu erhalten. Dies ist ein typisches Anwendungsgebiet für das Monitoring, da hier ohne personellen Aufwand ein autarkes Überwachungssystem installiert werden kann.

10.1 Das Messobjekt

Das Messobjekt ist ein Mehrzweckgebäude, gebaut als Mehrfamilienhaus im Jahre 1900. Es hat vier Vollgeschosse, in denen sich eine Arztpraxis, ein Vermessungsbüro und mehrere Wohnungen befinden.

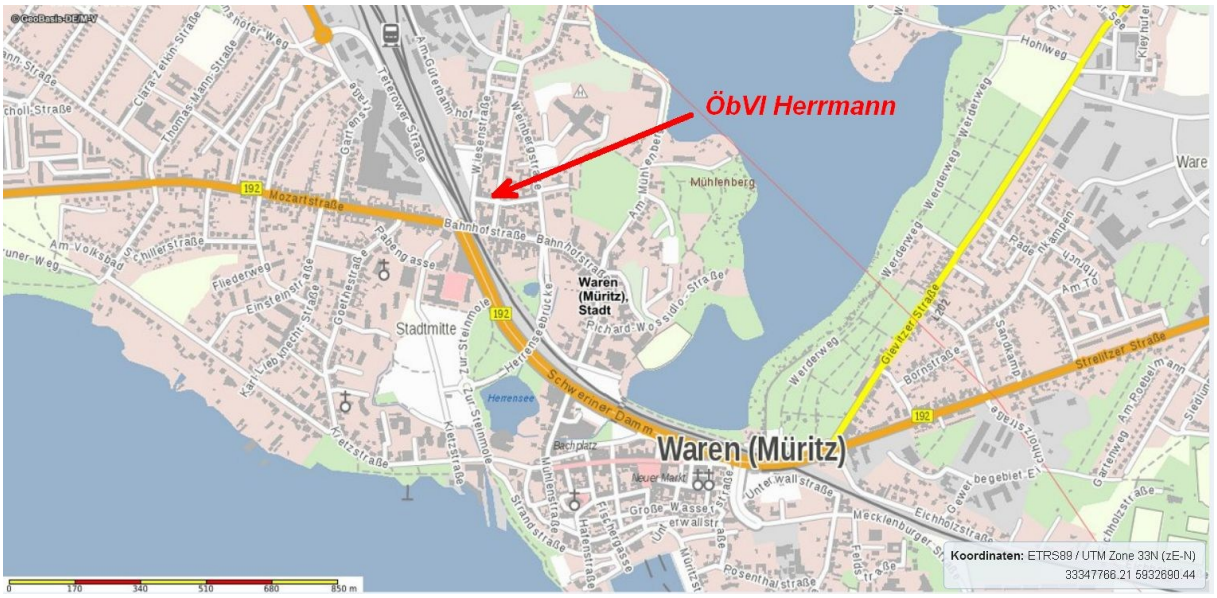


Abbildung 10.1: Lage des Vermessungsbüros in Waren (Müritz)

Das Gebäude des Vermessungsbüros liegt ca. 100 Meter östlich der Bahnstrecke Neustrelitz-Warnemünde. Diese Bahnstrecke, auch Lloydbahn genannt, ist eine zweigleisige Hauptbahn von ca. 130 Kilometer Streckenlänge. Auf dieser Bahnstrecke wird neben dem Personenverkehr ein Großteil des Güterverkehrs vom Hafen Rostock aus transportiert. Dementsprechend ist viel Güterverkehr auf der Strecke unterwegs. In den 1970er Jahren ist die Streckenführung in Waren von Südosten bis zum Bahnhof geändert worden. So konnte

der Engpass Waren mit vielen Bahnübergängen und kurvenreicher Streckenführung beseitigt werden.

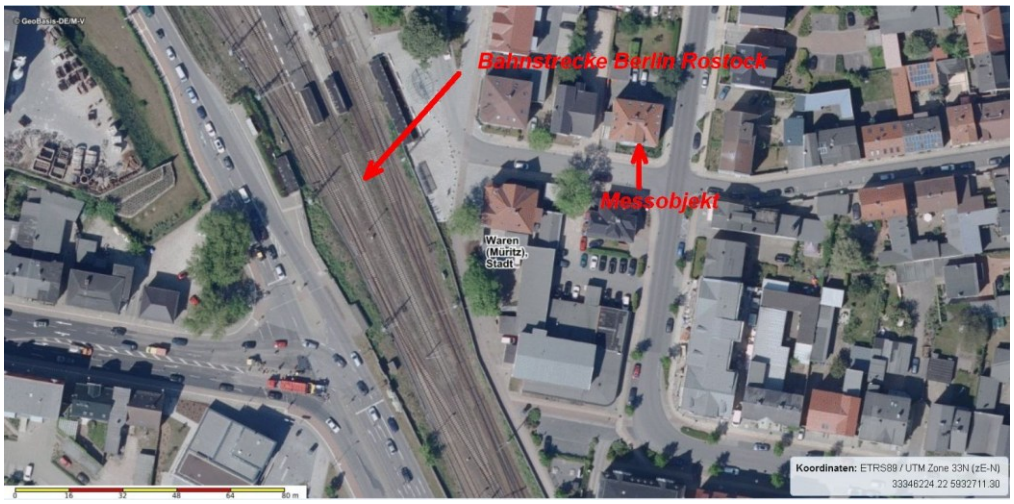


Abbildung 10.2: Übersicht Messobjekt und Bahnstrecke

Vom Jahre 2005 bis voraussichtlich 2019 wird die Eisenbahnstrecke ausgebaut. So soll die Geschwindigkeit auf 160 km/h und die maximale Achslast auf 25 Tonnen erhöht werden. [Deutsche Bahn AG, 2018] Dann können neben den Personenzügen auch die Güterzüge mit höherem Gewicht die Eisenbahnstrecke schneller befahren. Dadurch werden auch mehr Energien in Form von Schwingungen in den Untergrund abgeleitet.

Diese Schwingungen sind im umliegenden Bereich der Eisenbahnstrecke festzustellen. Im Vermessungsbüro des ÖbVI Herrmann wirken sich diese Schwingungen sehr nachteilig aus. Das Gebäude wird bei Durchfahrt eines Eisenbahnzuges in Schwingung versetzt. Daher besteht das Interesse, die Amplituden und die Frequenzen der Bauwerksschwingungen zu ermitteln.

10.2 Messaufbau

Zur Anwendung ist ein Sensor mit einem *Raspberry Pi* Einplatinencomputer als Auswerterechner gekommen. Die Messdaten konnten auf einer 64 Gigabyte Mikro-SD Karte gespeichert werden.

Abbildung 10.3: Sensor mit Raspberry Pi als Ausleserechner

Bei einer Abtastrate von ca. 400 Hz wird innerhalb einer Stunde ein Datenvolumen von 70 Megabyte erreicht. Pro Tag ergibt das ein Messdatenvolumen von 1,7 Gigabyte. Der nutzbare Speicherplatz von ca. 55 Gigabyte reicht für ca. 32 Tage. So muss spätestens nach einem Monat ein Auslesen der Messdaten erfolgen.

Abbildung 10.4: Außenansicht des Gebäudes des Vermessungsbüros des ÖbVI Herrmann mit dem Einbauort des Sensors

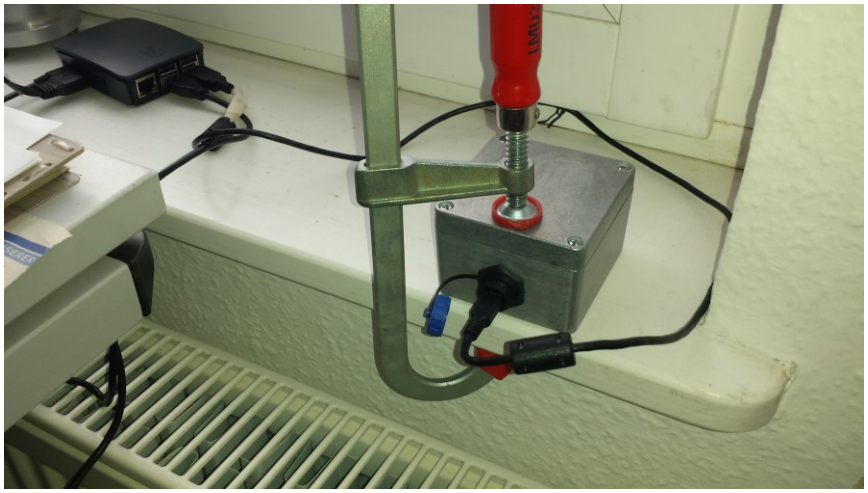


Abbildung 10.5: Sensoranbringung und Auswerterechner auf dem Fensterbrett

10.3 Auswertung

Über einen Zeitraum von 28 Tagen wurden Messdaten aufgezeichnet. Aus diesen Messdaten sind dann einzelne Tage herausgesucht und ausgewertet worden. Die komplette Messdatenmenge betrug rund 50 Gigabyte. Es waren innerhalb dieses Zeitraumes keine Ausfälle und Ausreißer zu erkennen. Ein Plot der Beschleunigungswerte der Achsen gibt dies wieder, siehe Abbildung 10.6

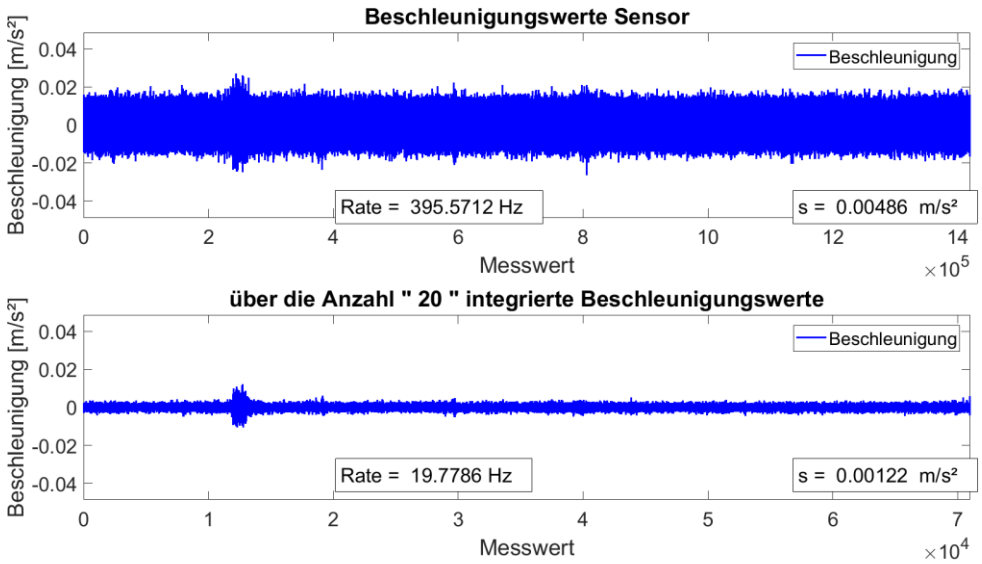


Abbildung 10.6: Plot der Beschleunigungswerte

In Abbildung 10.7 ist das Amplitudenspektrum über einen Zeitraum von einer Stunde dargestellt. Hier fällt auf, dass keine signifikanten Frequenzspitzen zu sehen sind.

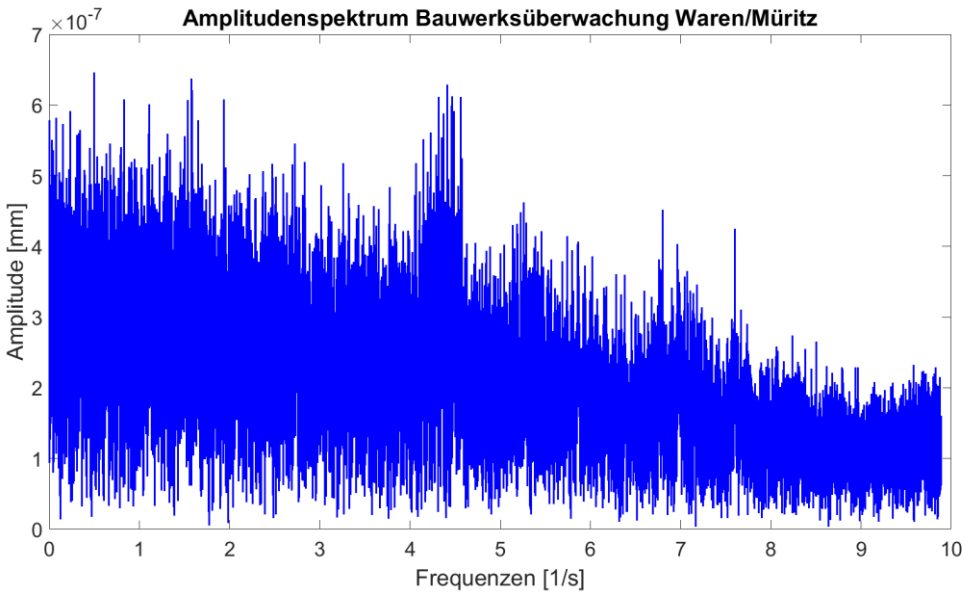


Abbildung 10.7: Amplitudenspektrum über eine Stunde

Die „fehlenden“ Frequenzspitzen lassen sich auf den zu langen Auswertezeitraum zurückführen. Dieses Beispiel soll exemplarisch die Schwierigkeiten der *Fouriertransformation* darstellen. Hier kann mit der normalen *Fouriertransformation* keine Aussage über eventuelle Schwingungen getroffen werden. Das Problem ist die Zeitdauer der Schwingungen. Eine Zugdurchfahrt dauert maximal 60 Sekunden. Dieser verhältnismäßig kurze Ereigniszeitraum im Verhältnis zum Auswertezeitraum von 60 Minuten verkleinert die Schwingungsspitzen derart, dass diese nicht mehr zu erkennen sind. Bei der Anwendung der *Fouriertransformation* müssen die einzelnen Ereignisse immer manuell selektiert und ausgewertet werden. Das ist mit Schwierigkeiten verbunden, wenn die die Amplituden der Schwingungen innerhalb der Messgenauigkeit des Sensors liegen. Hier ist nun die *Short-Time-Fourier-Transformation* klar im Vorteil, da nicht manuell selektiert werden muss, wie in Abbildung 10.8 zu erkennen ist.

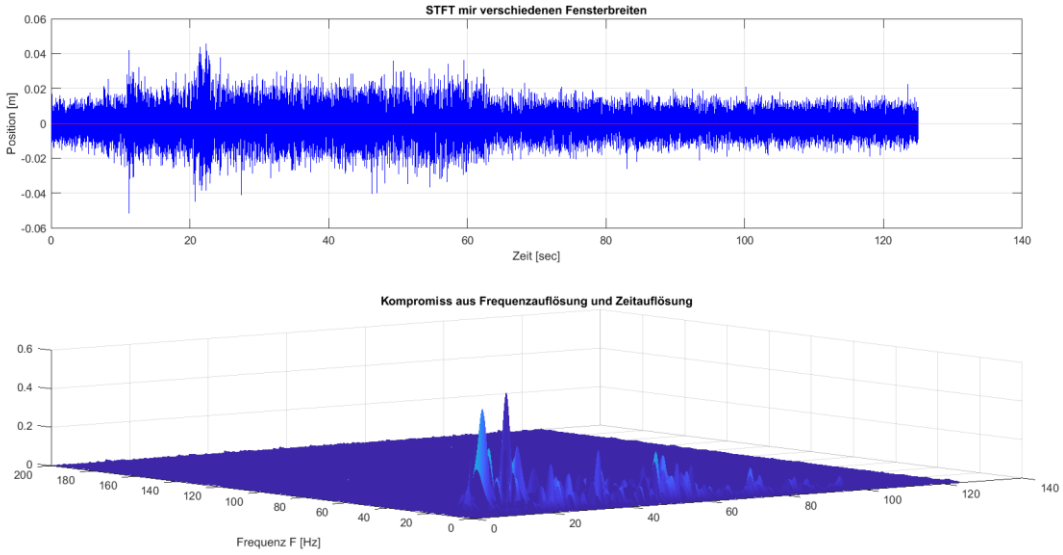


Abbildung 10.8: STFT über einen Zeitraum von 120 Sekunden

Über den gesamten Zeitraum hinweg können nun Amplituden und Frequenzen abgelesen werden. So ist es möglich die verschieden Ereignisse voneinander zu trennen. Es lassen sich die einzelnen Eisenbahnzüge aus dieser Darstellung erkennen. Dies ist ein großer Vorteil gegenüber den herkömmlichen Auswertemethoden, da die Zeitreihe nun im Zeitbereich und im Frequenzbereich innerhalb einer Darstellung ausgewertet werden kann.

Der Sensor misst Beschleunigungen in m/s^2, diese Angabe der Beschleunigungswerte ist aber nicht sehr praktisch. Interessanter sind hingegen Angaben zu den zurückgelegten Weglängen. Hier wird wie in Kapitel 6.6 beschrieben verfahren. Aus den Beschleunigungswerten werden durch doppelte Integration nach der Zeit die Wege bestimmt. Der Proportionalitätsfaktor zwischen den Beschleunigungen und den zurückgelegten Weglängen ist $1/(2*\pi*f)^2$. Nach Anbringen dieses Faktors werden die Amplituden der Schwingungen in Millimeter erhalten. Eine Zusammenfassung erfolgt in der Tabelle 10.1.

Tabelle 10.1: Zusammenstellung der Zeitpunkte des Auftretens, der Dauer, der Amplituden und der Frequenzen

Uhrzeit	Dauer [s]	Amplitude [mm]	Frequenz [Hz]
10:52	45	0,19	4,5
11:35	30	0,11	4,8
11:53	45	0,16	4,7
12:28	30	0,09	4,6

Die Tabelle 10.1 ist sehr aufschlussreich. Aus den angegebenen Uhrzeiten lässt sich durch Kenntnis des Fahrplanes der Deutschen Bahn AG eine Übereinstimmung mit den in diesem Zeitraum durchfahrenden Eisenbahnzügen feststellen. Damit kann die Ursache der Schwingungen des Gebäudes auf die Durchfahrten der Eisenbahnzüge zurückgeführt werden. Auch ist zu sehen, dass die Personenzüge um 11:35 Uhr und 12:28 Uhr eine deutlich geringere Amplitude als die Güterzüge zu den Zeiten 10:52 Uhr und 11:53 Uhr aufweisen. Dies ist durch das geringere Gewicht der Personenzüge zu erklären. Auch die höhere Geschwindigkeit der Güterzüge, die keinen Halt am Bahnhof einlegen, ist eine Ursache der geringeren Amplitude.

11 Monitoring des Turmes der St. Peter Kirche in Altentreptow

Das Monitoring der Hochschule Neubrandenburg am Turm der Marienkirche in Neubrandenburg blieb nicht ohne Außenwirkung. So ist auch die Kirchengemeine Altentreptow an einer Überwachung des Turmes der St. Peter Kirche in Altentreptow interessiert.

Abbildung 11.1: St. Peter Kirche in Altentreptow

Die St. Peter Kirche in Altentreptow ist in den letzten Jahrzehnten umfassend restauriert worden. Nachdem zuletzt der Dachstuhl des Kirchenschiffes erneuert wurde, beginnt jetzt die Sanierung des Kirchturms. Dabei sollen vor allem die Holzverkleidungen des Turmes ausgetauscht werden. Bei Arbeiten am Kirchturm sind auch hier Schwingungen bedingt durch das Glockengeläut aufgefallen. Eine Bestimmung von Amplitude und Frequenz dieser Schwingungen wird von Seite der Kirchengemeinde begrüßt. Für die Erprobung des neuentwickelten Sensors ist dies eine gute Gelegenheit, da mit der St. Peter Kirche in Altentreptow ein weiteres Messobjekt unter realen Bedingungen zur Verfügung steht. Interessant sind auch eventuelle Gemeinsamkeiten und Unterschiede zum Schwingverhalten der Marienkirche in Neubrandenburg.

11.1 Das Messobjekt

Die St. Peter Kirche in Altentreptow ist eine dreischiffige evangelische Stadtkirche. Sie stammt als Backsteinbau im gotischen Stil aus dem 14. Jahrhundert. Mit ihren Dimensionen von ca. 65 Meter Länge, 25 Meter Breite und 65 Meter Höhe ist diese Kirche das prägende Bauwerk in Altentreptow.

Abbildung 11.2: Innenstadt von Altentreptow mit der St. Peter Kirche

Der massige Kirchturm an der Westseite ist quadratisch errichtet worden. Ursprünglich besaß der Kirchturm eine Höhe von 90 Meter und war mit einem Spitzhelm abgeschlossen. Dieser ist bei einem Stadtbrand zerstört und durch einen 65 Meter hohen kleinen achteckigen Helm ersetzt worden, welcher im Jahre 1773 errichtet wurde.

Besonders markant ist auch das Glockenspiel in Altentreptow. Es besteht aus drei Glocken, mit insgesamt fast 8 Tonnen Gewicht. Die 4500 Kilogramm schwere „Große Glocke“ entstand im Jahre 1431 in der Gießerei Monkenhagen, die ihren Sitz in der Nähe von Rostock hatte. Die Glocke mit dem Schlagton cis^1 hat mit ihrer enormen Klangfülle eine beeindruckende Wirkung auf den Zuhörer. Die zweitgrößte Glocke, genannt Apostelglocke, stammt aus dem Jahre 1455. Sie ist 2500 Kilogramm schwer und hat den Schlagton d^1. Die kleinste Glocke ist die Sonntagsglocke mit dem Schlagton f^1, welche ursprünglich auch im Jahre 1431 entstand und ein Gewicht von 816 Kilogramm hatte. Diese wurde im Jahre 1929 abgehängt und an einer Kapelle weiterverwendet. Leider zersprang diese Glocke dabei und wurde im Jahre 1958 zu einer 770 Kilogramm schweren fis^1 Glocke umgegossen. Zu dieser

Zeit wurden alle drei Glocken an verkröpfte Stahljoche umgehängt, was das Klangbild minderte und die Sprunggefahr erhöhte. In den 1990er Jahren erfolgte eine Rückhängung aller Glocken an gerade Holzjoche. So befinden sich die Glocken heute in ihrem originalen Glockenstuhl.

Tabelle 11.1: Bestimmung der Glockenfrequenzen

Glocke	Gewicht	Frequenz [Hz]
Große Glocke	4500 kg	0,374
Apostelglocke	2500 kg	0,408
Sonntagsglocke	770 kg	0,462

Abbildung 11.3: Die drei Glocken der St. Peter Kirche in Altentreptow

11.2 Messaufbau

Zur Testmessung der Glockenbewegungen in der St. Peter Kirche Altentreptow wurden drei Sensoren verwendet. Zusätzlich ist ein Leica Nivel210 auf der Ebene des Glockenstuhls angebracht worden, welches zur Verifizierung der Messdaten dient. Die drei Sensoren befinden sich in unterschiedlichen Höhen. Sensor 1 wurde unterhalb der Glocken, Sensor 2 und das Leica Nivel210 ca. 2 Meter über den Jochen der Glocken und Sensor 3 ist ca. 15 Meter oberhalb des Glockenstuhls befestigt. So ist es möglich in verschieden Ebenen zu messen und die Auswirkungen der Deformationen des Glockenläutens auf den Kirchturm zu

bestimmen. Dies ist das Minimum um eine mögliche Biegeline zu bestimmen. Die Auswirkungen der Glockenschwingungen werden vergleichbar mit den Auswirkungen der fünf Glocken der Marienkirche in Neubrandenburg sein. So müsste mit zunehmender Höhe des Kirchturmes die Auslenkung des Kirchturmes zunehmen. Um dieses Verhalten nachzuweisen, sind mindestens drei Sensoren, die verteilt am Kirchturm angebracht werden, erforderlich.

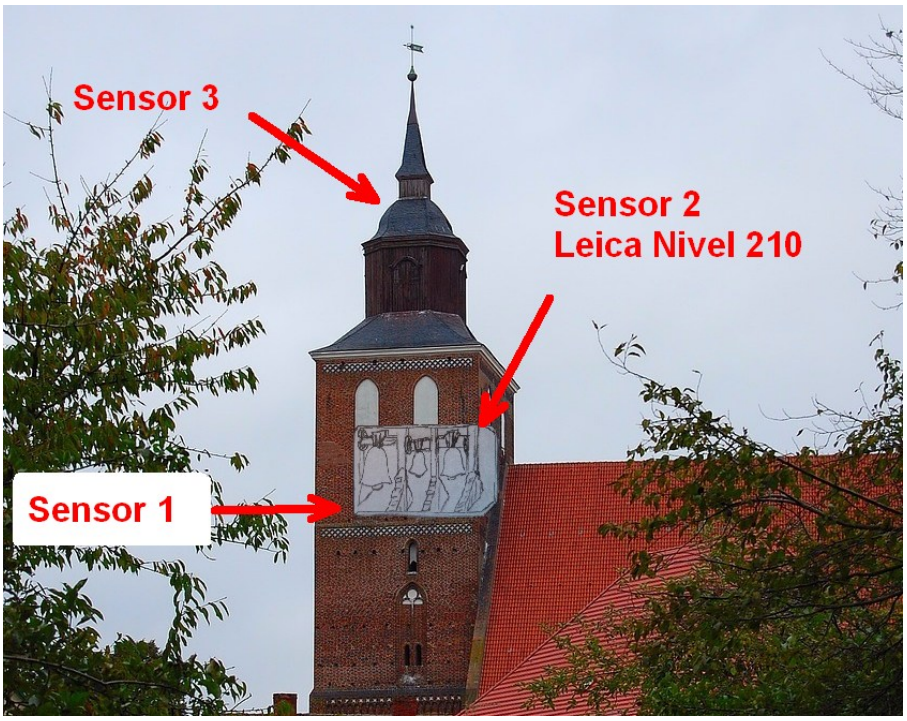


Abbildung 11.4: Sensorpositionen im Glockenturm der St. Peter Kirche

Die Sensoren sind entweder an Holzbalken des Kirchturmbodens oder in Mauernischen befestigt worden. Um die Stromversorgung sicherzustellen sind alle drei unteren Sensoren, also die zwei Testsensoren und das Leica Nivel210 durch das Stromnetz der Kirche versorgt worden. Der vierte Sensor ist autark über einen Akkumulator versorgt worden. Hier zeigte sich, dass eine autarke Stromversorgung von großem Vorteil ist, da eine Unabhängigkeit von konventionellen Systemen einen Ausfall des Sensors ausschließen kann.

11.3 Auswertung

Der Zeitraum der Messung an der St. Peter Kirche betrug eine Stunde. Dieser Zeitraum ist in ca. 45 Minuten Ruhemessung und ca. 15 Minuten Glockenspiel unterteilt. Die Glocken wurden jeweils fünf Minuten in gewichtsaufsteigender Reihenfolge geläutet. Innerhalb

dieses Zeitraumes waren keine Ausfälle und keine Ausreißer zu erkennen. Ein Plot der Beschleunigungswerte der y-Achsen, die in Schwingrichtung der Glocken ausgerichtet ist, spiegelt dies wieder, siehe Abbildung 11.5.

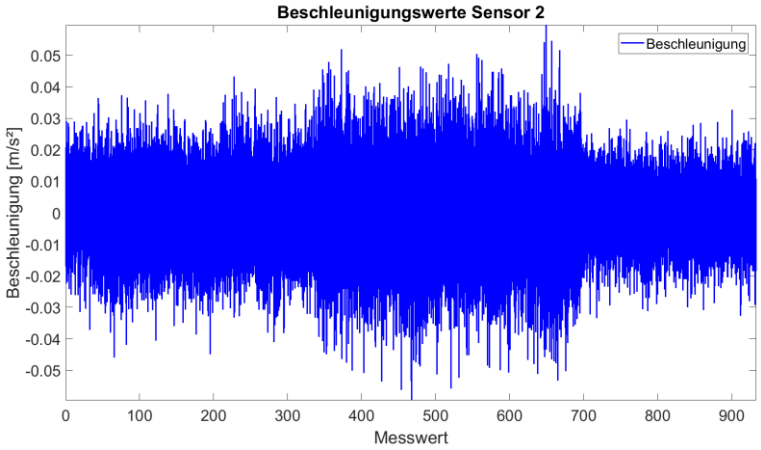


Abbildung 11.5: Plot der Beschleunigungswerte der y-Achse (15 Minuten)

In Abbildung 11.6 ist das Amplitudenspektrum über einen Zeitraum von 15 Minuten dargestellt. Die Glockenfrequenzen anhand der Tabelle 11.1 konnten teilweise ermittelt werden. Zusätzlich ist es auch möglich gewesen, die Bauwerksschwingungen, welche dreimal so hohe Frequenzen wie die Glockenfrequenzen aufweisen, zu ermitteln. Das Verhalten des Mauerwerkes im Bezug zu den Glockenfrequenzen ist vergleichbar mit dem Verhalten, das bei der Marienkirche Neubrandenburg ermittelt wurde. Das Mauerwerk „antwortet“ auch bei der St. Peter Kirche in Altentreptow mit der dreifachen Frequenz. Weiterführende Informationen hierfür sind bei [Köster, 2016] und [Engel, et al., 2017] zu finden.

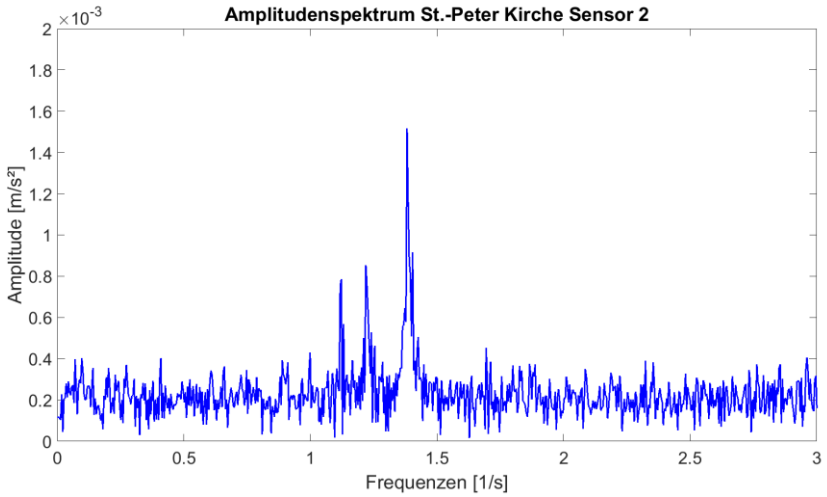


Abbildung 11.6: Amplitudenspektrum über eine Viertelstunde

Die Amplitude der Schwingung der Sonntagsglocke ist korrekt wiedergegeben worden, da diese über den kompletten Auswertezeitraum von 15 Minuten geläutet wurde. Bei der Apostelglocke (10 Minuten Läuten) und der Großen Glocke (5 Minuten Läuten) ist der Zeitraum der Schwingungen kürzer als der Auswertezeitraum. Daher sind diese Amplituden in Abbildung 11.6 nicht korrekt dargestellt. Hier ist immer darauf zu achten, dass der Auswertezeitraum mit dem Schwingungszeitraum übereinstimmt. Dies erfordert immer eine manuelle Bearbeitung der Auswahl des Auswertezeitraumes. Eleganter ist die Auswertung mittels *STFT*. Hier können Amplituden, Frequenzen und Zeiträume ermittelt werden, wie in Abbildung 11.7 zu erkennen ist.

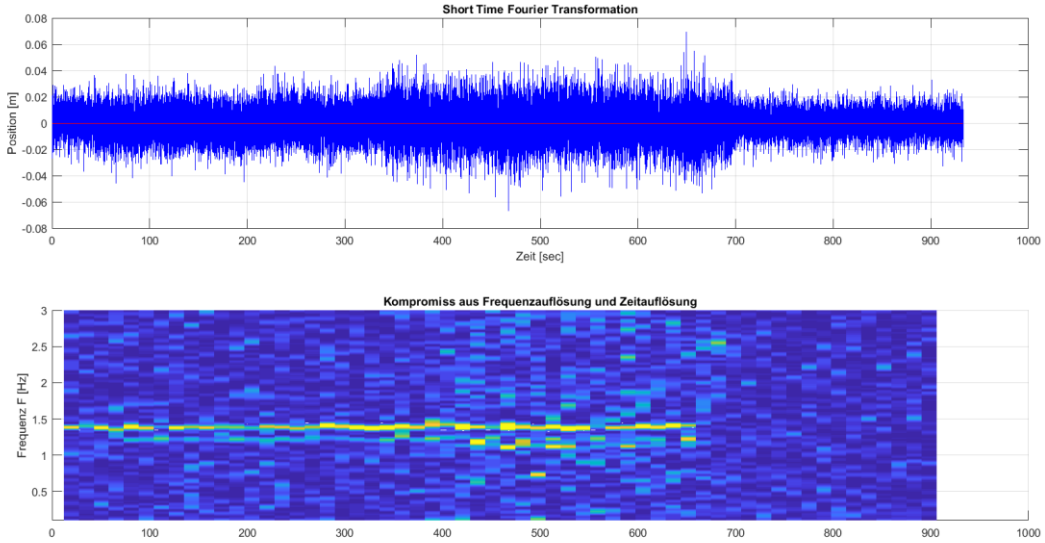


Abbildung 11.7: STFT der Viertelstundenmessung

Amplituden und Frequenzen können über den ganzen Zeitraum abgelesen werden. Eine Trennung der Ereignisse ist auch hier möglich. Die einzelnen Gebäudeschwingungen sind ablesbar. Die Schwingung mit einer Frequenz von 1,4 Hz ist über einen Zeitraum von Null bis 650 Sekunden aufgetreten, die Schwingung mit einer Frequenz von 1,22 Hz von 50 Sekunden bis 650 Sekunden und die Schwingung mit der Frequenz von 1,11 Hz von 400 bis 650 Sekunden.

Das hier wieder „nur“ Beschleunigungswerte in m/s² gemessen wurden, müssen diese in Auslenkungen Millimeter umgewandelt werden. Eine Zusammenfassung erfolgt in der Tabelle 10.1.

Tabelle 11.2: Zusammenstellung der Zeitpunkte des Auftretens, der Dauer, der Amplituden und der Frequenzen der Glockenschwingungen

Uhrzeit	Dauer [s]	Amplitude [mm]	Frequenz [Hz]
10:15	650	0,05	0,37
10:20	600	0,1	0,40
10:25	250	0,2	0,47

Aus den angegebenen Uhrzeiten lässt sich der Läuteplan der Glocken der Kirche in Altentreptow ablesen. Jeweils fünf Minuten lang wird jede Glocke einzeln geläutet. Bei der Erfassung der Glockenschwingungen sind hier die ersten 250 Sekunden der Sonntagsglocke nicht erfasst worden. Die dazugehörigen Mauerschwingungen sind in Tabelle 11.3 aufgeführt.

Tabelle 11.3: Zusammenstellung der Zeitpunkte des Auftretens, der Dauer, der Amplituden und der Frequenzen der Mauerschwingungen

Uhrzeit	Dauer [s]	Amplitude [mm]	Frequenz [Hz]
10:15	650	0,05	1,11
10:20	600	0,1	1,20
10:25	250	0,03	1,41

Die Tabellenwerte entsprechen jeweils dem am höchsten angebrachten Sensor. Die Messung war erfolgreich, da die Tauglichkeit des Sensors für Ermittlung von Bauwerksschwingungen konnte in vollem Umfang nachgewiesen werden konnte. Es stellte sich heraus, dass auch bei der St. Peter Kirche in Altentreptow die Bauwerksschwingungen den Faktor drei zu den Glockenfrequenzen aufweisen. Das entspricht den Werten der Messergebnissen, die an der Marienkirche in Neubrandenburg ermittelt wurden, siehe [Engel, et al., 2017].

12 Zusammenfassung und Ausblick

In diesem Kapitel wird ein Überblick der erreichten Ziele der Masterarbeit gegeben, sowie ein Ausblick auf mögliche weitere Anwendungen. Das primäre Ziel der Masterarbeit ist die Konstruktion, der Kalibrierung und der Anwendung eines MEMS-basierten Sensors für geodätische Überwachungsmessungen. Als weiteres Ziel ist die Anwendung einer Kurzzeitfouriertransformation im Bereich des geodätischen Monitorings. Hierbei werden Vor- und Nachteile dieser Auswertestrategie vorgestellt werden.

12.1 Sensorkonstruktion und Anwendung

Aus der Sensorkonstruktion sind verschiedene Prototypen entstanden. Grundsätzlich sind hier der MEMS-Sensor und das Auslesesystem, der Mikrocontroller, bei allen Prototypen in ähnlicher Form vorhanden, siehe Abbildung 12.1.

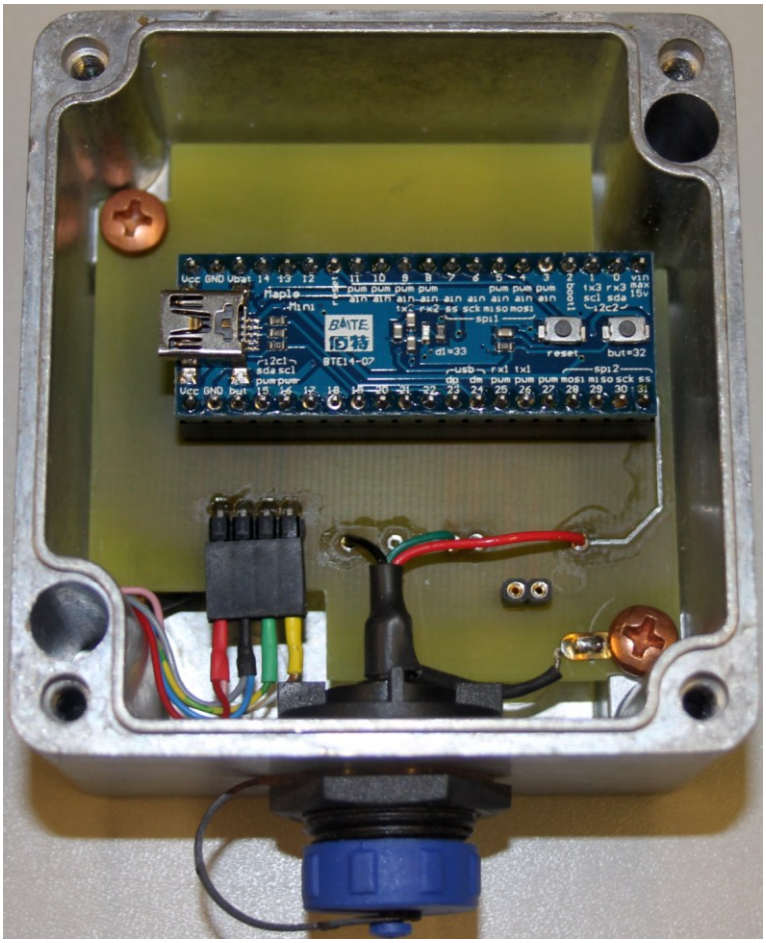


Abbildung 12.1: Sensorgehäuse mit Mikrocontroller, Anschlüssen und MEMS-Sensor (verdeckt)

Die Variante für das Monitoringsystem Dabamos ist mit diesen Komponenten fast komplett.

Bei dieser Version des Sensors werden die Messdaten über die serielle Schnittstelle ausgegeben. Die Messdaten werden dann mittels eines seriellen Geräteservers, der die seriellen Daten in ein Netzwerk einbindet, an die Monitoringsoftware übertragen. Anwendung fand diese Variante bei der Marienkirche Neubrandenburg, wo sie in das bestehende Monitoringsystem eingebunden ist und das Kirchengebäude überwacht.

Eine zweite Variante ist direkt als Einzelplatzlösung vorgesehen. Hier übernimmt ein Windows basierter Rechner die Auslese- und Auswerteaufgabe. Der Sensor überträgt seine Daten über den USB-Ausgang, der am Gehäuse angebracht ist, an den Rechner. Bei dem Prototyp ist der Rechner als Tablet ausgeführt, siehe Abbildung 12.2 .

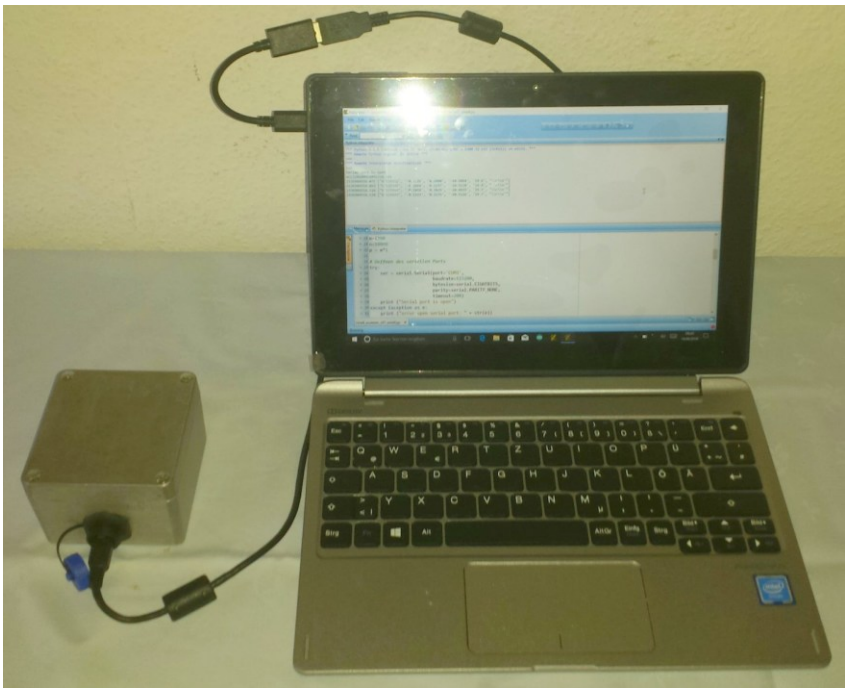


Abbildung 12.2: Variante mit Tablet als Auswerterechner

Bei der dritten Variante ist der Ausleserechner ein Einpaltinencomputer *Raspberry Pi*. Bei dieser Variante konnte die Sensor- und Auslesekombination verkleinert werden. Ebenfalls ist durch den Einsatz des *Raspberry Pi* als Rechner der Kostenfaktor erheblich verringert worden, da die Kosten des *Raspberry Pi* mit Gehäuse und Display ca. 80 € betragen. Die Kosten für ein preiswertes Tablet inklusive Betriebssystem betragen ca. 170 €. Diese Variante ist in Abbildung 12.3 dargestellt.

Abbildung 12.3: Variante mit Raspberry Pi und 7 Zoll Display

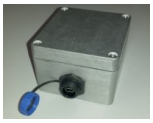
Die Kompaktheit hat bei der dritten Variante nicht vollständig überzeugt. Daher ist eine vierte Variante verwendet worden. Hierbei ist das große 7 Zoll Display durch ein kleines Display mit 3 Zoll, das direkt im Gehäuse integriert ist, ersetzt worden. Auch dieses Display hat eine Bedienmöglichkeit über Touch. Die Kosten für die Verwendung dieser Variante des *Raspberry Pi* betragen ca. 60 €, was noch einmal eine weitere Kosteneinsparung ergibt. Da das Display direkt im Gehäuse integriert ist, hat der Ausleserechner eine Größe von 95 Millimeter x 65 Millimeter. Diese Maße sind zusammen mit den Maßen des Sensors, 79 Millimeter x 74 Millimeter, höchst und ermöglichen den Einbau in beengte Plätze, siehe Abbildung 12.4.

Abbildung 12.4: Variante mit 3 Zoll Display, hier mit einem Akkumulator als Spannungsversorgung.

Zusätzlich ist die Verwendung eines Akkumulators getestet worden, da hier eine Unabhängigkeit vom Stromnetz besteht. Dies erwies sich vor allem in exponierten Einbauorten, wie im Kirchturm der St, Peter Kirche in Altentreptow, als sehr vorteilhaft. Hier hätten sonst mindestens 30 Meter Kabel für die Stromversorgung durch den Kirchturm verlegt werden müssen. So konnte die Zeit für den Auswand der Messvorbereitung erheblich verkürzt werden. Da potentielle Messobjekte nicht immer am Stromnetz angebunden sind, z.B. Brücken, ist diese Variante in Kombination mit Ausladeмöglichkeiten wie einem Solarpanel eine interessante Variante.

In Tabelle 12.1 ist ein Überblick der verschiedenen Varianten gegeben.

Tabelle 12.1: Variantenvergleich

Varianten	1	2	3	4
Ansicht				
Datenübertragung	Seriell	USB	USB	USB
System	Netzwerk	Einzel	Einzel	Einzel
Abmessungen	79 x 74 mm	79 x 74 mm Sensor 250 x 190 mm Rechner	79 x 74 mm Sensor 185 x 130 mm Display 65 x 64 mm Rechner	79 x 74 mm Sensor 95 x 65 mm Rechner
Kosten	100€ Sensor	100 € Sensor 170 € Rechner	100 € Sensor 50 € Rechner 30 € Display	100 € Sensor 50 € Rechner 10 € Display

Alle vier Varianten lieferten bei den Testmessungen verwertbare Ergebnisse. Neben der Testmessung an der Marienkirche Neubrandenburg zur Validierung der Sensordaten sind Daten dreier weiterer Objekte gemessen und ausgewertet worden. An der St. Peter Kirche in Altentreptow wurde ähnlich der Messung an der Marienkirche in Neubrandenburg ein historisches Gebäude überwacht. Im Gebäude 2 der Hochschule Neubrandenburg fand ein Monitoring einer freitragenden Treppe statt und in Waren/Müritz wurden Informationen zum Bewegungsverhalten eines Mehrfamilienhauses in direkter Nähe zu einer Eisenbahnstrecke ermittelt.

Tabelle 12.2: Messobjekte und Auswertemethoden

Messobjekt	Marienkirche Neubrandenburg	St. Peter Kirche Altentreptow	Treppe Hochschule Neubrandenburg	Mehrfamilienhaus Waren/Müritz
Ansicht				
System	Netzwerk	Einzel	Einzel	Einzel
Messdauer	Dauermessung	30 Minuten während des Glockenläutens	eine Woche	vier Wochen
Auswertung	Ereignisse in den Messdaten Fouriertransfor-mation Kurzzeitfourier-transformation	Ereignisse in den Messdaten Fouriertransfor-mation Kurzzeitfourier-transformation	Ereignisse in den Messdaten Fouriertransformation Kurzzeitfouriertrans-formation	Ereignisse in den Messdaten Fouriertransfor-mation Kurzzeitfouriertrans-formation

12.2 Anwendung der Kurzzeitfouriertransformation

Mit den hier verwendeten Datenanalysemethoden aus der Signaltheorie kann der Auswertungsspielraum der Ingenieurgeodäsie erheblich erweitert werden. Bisher werden zur Frequenzanalyse meist nur die "normale" Fouriertransformation in diesem Ausgabenbereich verwendet. Die Vorteile der Wavelettransformation und insbesondere der Kurzeitfouriertransformation sind nicht von der Hand zu weisen. Jetzt sind Werkzeuge vorhanden, mit denen auch die zeitlichen Veränderungen der Schwingungsfrequenzen aufgezeigt werden können, siehe. Bei den Messdaten des Kirchturmes der Marienkirche in Neubrandenburg sind die zeitlichen Änderungen bekannt, d.h. die Dauer und die Zeitpunkte des Glockenläutens. Deshalb wurde untersucht ob diese zeitlichen Änderungen auch nachgewiesen werden können. Diese zeitlichen Änderungen konnten nachgewiesen werden. Die Messdaten des Leica Nivel 210 sowie der Testmessung mit dem neuentwickelten Sensor konnten in vollem Umfang ausgewertet werden. Hierbei bestätigten sich die Zeitpunkte der zeitlichen Änderungen. Die Abbildung 12.5 zeigt zeitliche Änderungen der Schwingungsfrequenzen.

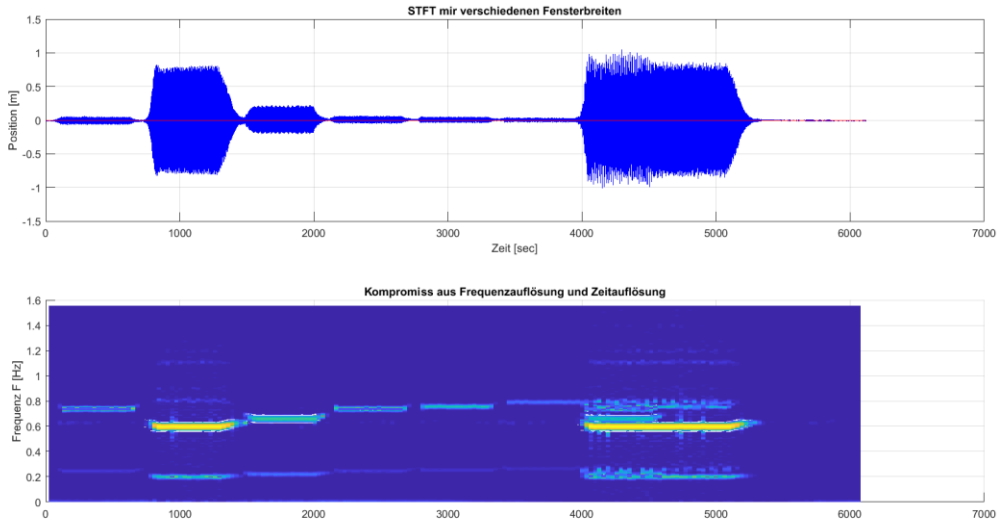


Abbildung 12.5: Kurzzeitfouriertransformation des Glockenspiels der Marienkirche Neubrandenburg

Wie in Abbildung 12.5 zu erkennen ist, konnten die Glockenfrequenzen und der zeitliche Beginn und Ende der Schwingungen eindeutig sichtbar gemacht werden.

Bei der St. Peter Kirche Altentreptow ist das Verhalten des Turmes und der Läuteplan vorab nicht bekannt gewesen. Hier erweist sich der neuentwickelte Sensor in Verbindung mit der Kurzzeitfouriertransformation als besonders effektiv. Nach nur einer Messung mit einer Dauer von ca. 30 Minuten kann ein Schwingverhalten sowohl zeitlich als auch in der Amplitude bestimmt werden. In Abbildung 12.6 sind die zeitlichen Verläufe und die Amplituden der Turmüberwachung dargestellt worden.

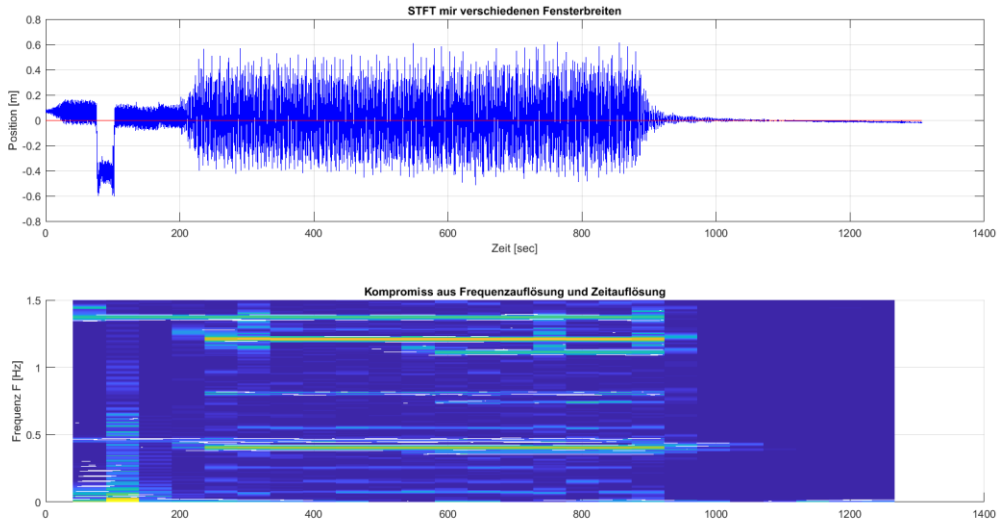


Abbildung 12.6: Turmüberwachung der St. Peter Kirche Altentreptow mit Fokus auf der Frequenzauflösung

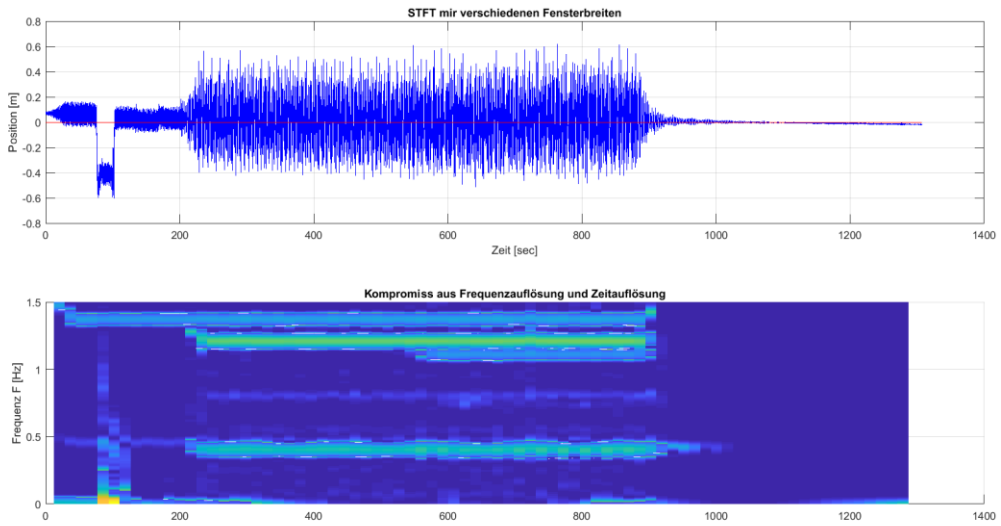


Abbildung 12.7: Turmüberwachung der St. Peter Kirche Altentreptow mit Fokus auf der Zeitauflösung

Bei der Kurzzeitfouriertransformation ist immer ein Kompromiss zwischen der Zeitauslösung und der Frequenzauflösung erforderlich, da beide Auflösungen können nicht beliebig gesteigert werden können. Eine nähere Erläuterung findet sich hierzu in Kapitel 5.3.1. In Abbildung 12.6 ist der Fokus auf die Frequenzauflösung gelegt worden. Dadurch sind aber die Zeitabschnitte recht grob gewählt worden. Eine exakte Zuordnung der Zeitpunkte der einzelnen Schwingungsereignisse ist nur bedingt möglich. Es können so die einzelnen Frequenzen sehr gut voneinander getrennt werden. In Abbildung 12.7 ist der Fokus auf die

Zeitauflösung gelegt worden. Dabei ist erkennbar, dass die Zeitabschnitte sehr kurz sind, also eine gute zeitliche Zuordnung zu den Ereignissen gegeben ist. Die Frequenzen hingegen werden nur sehr grob abgebildet. Hier ist eine Verschmelzung der drei Frequenzen von 0,37 Hz, 0,40 Hz und 0,47 Hz zu einer Frequenz bei ca. 0,4 Hz zu sehen. So muss immer ein Kompromiss zwischen der Zeitauflösung und Frequenzauflösung gewählt werden um optimale Ergebnisse zu erhalten.

Die Anwendung der Kurzzeitfouriertransformation lässt ein erweitertes Anwendungsspektrum zu, da bei dem ingenieurgeodätischem Monitoring üblicherweise nicht klar ist, wann Schwingungseinwirkungen beginnen und wie lange diese anhalten. Somit kann nach der Ursache der Schwingungseinwirkung geforscht und diese in einen kausalen Zusammenhang mit der Schwingung gebracht werden. Dies kann in vielfältiger Weise zur Standsicherheit von Gebäuden beitragen.

12.3 Ausblick

Die Untersuchungen haben gezeigt, dass der in dieser Masterarbeit entwickelte Sensorblock vergleichbare Genauigkeit wie professionelle Sensoren aufweist und bei den untersuchten Messobjekten signifikante Ergebnisse liefern konnte. Die Idee, einen sehr kostengünstigen Sensor zu entwickeln ist umgesetzt worden.

Um diesen Sensorblock im realen Umfeld testen zu können, ist unter anderem die Integration in das Monitoring-System der Marienkirche Neubrandenburg erfolgt. Neben diesem Vorhaben ist der Sensor an weiteren Messobjekten arbeiten. Im Rahmen dieser Masterarbeit sind drei verschiedene Objekte überwacht worden. Einem Einsatz des Sensors an Brücken oder anderen Ingenieurbauwerken ist möglich und sinnvoll.

Das Programm sollte sicherlich in Zukunft erweitert werden. Hier werden die späteren Anwendungsszenarien weitere Verfeinerungen des Programmes erforderlich machen. So sollte zum Beispiel auch eine grafische Benutzeroberfläche implementiert werden. Das würde die Arbeit mit dem Programm vereinfachen.

Eine weitere Zukunftsidee ist die Ansteuerung weiterer Sensoren. So könnte die Ansteuerung eines kombinierten Beschleunigungs- und Drehratensensor *(6DOF)*

vorgenommen werden. Dabei könnten die Drehratensensoren ausgelesen und eine Erweiterung des Einsatzbereiches kann auf kinematische Anwendungen erfolgen. Hierbei müssten zusätzlich zu den drei Achsen der Beschleunigung auch die drei Achsen der Winkelbeschleunigung ausgewertet werden.

Die Anwendungsmöglichkeiten der Kurzzeitfouriertransformation und der Wavelettransformation können ebenfalls ausführlicher untersucht werden, da hierbei ein hohes Anwendungspotential vorliegt.

13 Literaturverzeichnis

Analog Devices. 2018. *EVAL-ADXL354/EVAL-ADXL355 User Guide.* [Dokument] Norwood : Analog Devices, 2018.

Clarke, Robert. 2018. Blue Pill STM32duino wiki. *STM32duino wiki.* [Online] 16. 05 2018. [Zitat vom: 16. 07 2018.] http://wiki.stm32duino.com/index.php?title=Blue_Pill.

Deutsche Bahn AG. 2018. *Deutsche Bahn AG.* [Online] Bauprojekt Berlin – Rostock | BauInfoPortal der Deutschen Bahn, 2018. [Zitat vom: 17. 08 2018.] https://bauprojekte.deutschebahn.com/p/berlin-rostock.

DIN. 2018. DIN ISO 17359:2018-05, Zustandsüberwachung und -diagnostik von Maschinen - Allgemeine Anleitungen (ISO 17359:2018). *2018.* s.l. : Deutsches Insttiut für Normung E.V., 2018.

DLR, Deutsches Zentrum für Luft- und Raumfajrt. 2016. Structural Health Monitoring ermöglicht Detektion von Schäden in Flugzeugbauteilen aus CFK. *DLR- DLR Portal.* [Online] 19. 02 2016. [Zitat vom: 16. 07 2018.] https://www.dlr.de/dlr/Portaldata/1/Resources/bilder/portal/braunschweig/SARISTU1.jpg.

DVW. 2004. *Interdisziplinäre Messaufgaben im Bauwesen – Weimar 2004: Beiträge zum 61. DVW-Fortbildungsseminar am 27. und 28. September 2004 an der Bauhaus-Universität Weimar.* Weimar : Wißner-Verlag, 2004.

Elektronik, Kompendium. 2016. www.elektronik-kompendium.de/sites/bau/1503041.htm. *elektronik.kompendium.de.* [Online] 27. 12 2016. [Zitat vom: 27. 12 2016.] http://www.elektronik-kompendium.de/sites/bau/1503041.htm.

Elektronik, Kompendium. 2017. www.elektronik-kompendium.de/sites/com/1810231.htm. *elektronik.kompendium.de.* [Online] 01. 01 2017. [Zitat vom: 04. 01 2017.] http://www.elektronik-kompendium.de/sites/com/1810231.htm.

Engel, Philipp, Foppe, Karl und Köster, Uwe. 2017. Ingenieurgeodätisches Monitoring an der Marienkirche Neubrandenburg. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement.* 2017, 4.

Faro. 2012. Faro Lasertracker Vantage Technische Spezifikationen. 2012.

Fraunhofer-Institut für Nauturwissenschaftlich-Technische Trendanalysen. 2015.

https://www.int.fraunhofer.de/content/dam/int/de/documents/EST/EST%200715%20Strutural%20Health%20Monitoring.pdf. *https://www.int.fraunhofer.de.* [Online] 01. 05 2015. [Zitat vom: 16. 07 2018.] https://www.int.fraunhofer.de/content/dam/int/de/documents/EST/EST%200715%20Strutural%20Health%20Monitoring.pdf.

Gräfe, Dirk. 2016. de.wikipedia.org/wiki/Drehtisch. *www.wikipedia.org.* [Online] 16. 06 2016. [Zitat vom: 04. 01 2017.] https://de.wikipedia.org/wiki/Drehtisch#/media/File:Rundtisch_W%C3%A4lzsto%C3%9Fmaschine.JPG.

HBM, Hottinger Baldwin Messtechnik GmbH. 2018. Structural Health Monitoring HBM. *HBM Messtecnik.* [Online] 16. 07 2018. [Zitat vom: 16. 07 2018.] https://www.hbm.com/fileadmin/mediapool/images/applications/structuraltesting/illu-shm-data-transfer.jpg.

Hering, Ekbert, Bressler, Klaus und Gutekunst, Jürgen. 2017. *Elektronik für Ingenieure und Naturwissenschaftler.* Heubach, Ditzingen, Nütringen : Springer Vieweg, 2017.

Huiskamp, Wim. 2018. Open-source RTOS for IoT development. *MapleMini_Hello - Using low cost LeafLab Maple Mini boards with mbed.* [Online] 01. 08 2018. [Zitat vom: 01. 09 2018.] https://os.mbed.com/users/hudakz/code/MapleMini_Hello/.

IDS Innomic GmbH. 2008. IDS Innomic GmbH Schwingungsmesstechnik. *Rauschen und Auflösung IDS Innomic GmbH Schwingungsmesstechnik.* [Online] 01. 01 2008. [Zitat vom: 01. 09 2018.] https://www.innomic.com/service/schulungen/schulungsmaterial/lernkurs_piezoelektrische_beschleunigungssensoren/kenndaten/rauschen_und_aufloesung.html.

Ingenieurbüro Heimann. 2018. Ingenieurbüro Heimann. *Ingenieurbüro Heimann >> C# Entwickler.* [Online] 01. 08 2018. [Zitat vom: 01. 09 2018.] https://www.heimann-online.com/c-sharp-entwickler/.

Ionescu, Horia. 2016. en.wikipedia.org/wiki/Six_degrees_of_freedom. *Wikipedia.* [Online] 26. 12 2016. [Zitat vom: 28. 12 2016.] https://en.wikipedia.org/wiki/Six_degrees_of_freedom#/media/File:6DOF_en.jpg.

K, Bob. 2005. Wikipedia. *Fensterfunktion - Wikipedia.* [Online] 15. 12 2005. [Zitat vom: 01. 09 2018.] https://de.wikipedia.org/wiki/Fensterfunktion.

Knickmeyer, Elfriede. 2014. *Skript Navigation.* Hochschule Neubrandeburg : s.n., 2014.

Knopf, P. 2013. www.karl-gotsch.de/Monatsbruecken/2013Feb.htm. *www.karl-gotsch.de.* [Online] 02 2013. [Zitat vom: 28. 12 2016.] http://www.karl-gotsch.de/Bilder/Ruegenbruecke10.jpg.

Köster, Uwe. 2016. *Multisensor-Monitoring zur Detektion hochfrequenter Schwingungen an der Marienkirche Neubrandenburg.* [Bachelorarbeit] Neubrandenburg : s.n., 2016.

Küveler, Gerd und Schwoch, Dietrich. 2017. *C/C++ für Studium und Beruf.* Glashütten : Springer Vieweg, 2017. ISBN 978-3-658-18580-0.

Larisch, Hans-Jürgen. 2013. *Skript Sensorik und spezielle Auswertemethoden.* Hochschule Neubrandenburg : s.n., 2013.

Lerch, Reinhard. 2016. *Elektrische Messtechnik.* Erlangen : Springer Vieweg, 2016. ISBN 978-3-662-46940-8.

Mertins, Alfred. 2013. *Signaltheorie - Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung.* Wiesbaden : Springer-Verlag, 2013. Bd. 3.

Möser, et al. 2012. *Handbuch Ingenieurgeodäsie, Grundlagen.* Dresden : Wichmann, 2012.

Möser, Hoffmeister, Müller, Schlemmer, Staiger, Wanninger. 4. Auflage 2012. *Handbuch Ingenieurgeodäsie, Grundlagen.* Dresden : Wichmann, 4. Auflage 2012.

Muffert, M, Eling, C. und Kuhlmann, H. 2009. Grundlagen der Zeitreihenanalyse im Frequenzbereich. *Schriftenreihe des DVW: Zeitabhängige Messgrößen - Verborgene Schätze in unseren Daten.* 2009, Bd. 59.

Neuner, H. und K., Foppe. 2009. Grundlagen der Zeitreihenanalyse im Zeitbereich. *Schriftenreihe des DVW: Zeitabhängige Messgrößen - Verborgene Schätze in unseren Daten.* 2009, Bd. 59.

Nitschke, Martin. 2012. *Vorlesungsmanuskript VBGI07 Mathematik 2 - Hochschule Neurbeandenburg.* 2012.

Nordkurier. 2015. www.nordkurier.de/mueritz/a-19-dicht-droht-der-verkehrskollaps-0316575608.html. *www.nordkurier.de.* [Online] 03. 08 2015. [Zitat vom: 28. 12 2016.] http://www.nordkurier.de/sites/default/files/styles/teaser_bild_423px/public/dcx/2015/08/03/doc6lnj5duzm29lqzn5lp2__file6ka1vb9alvl17h1bah9.jpg?itok=VrfPk5Pg.

n-tv, Nachrichtenfernsehen GmbH. 2011. www.n-tv.de/panorama/Gutachter-muss-erneut-vor-Gericht-article4311976.html. *www.n-tv.de.* [Online] 15. 09 2011. [Zitat vom: 28. 12 2016.] http://bilder1.n-tv.de/img/incoming/origs4311936/2302737340-w1000-h960/DI10298-20110914.jpg.

Python, Community. 2008. www.python.org/community/logos/. *www.python.org.* [Online] 06. 08 2008. [Zitat vom: 04. 01 2017.] https://upload.wikimedia.org/wikipedia/commons/f/f8/Python_logo_and_wordmark.svg.

Resnik, Boris, Friedrich, Andreas und Gerstenberg, Josef. 2013. Kalibrierung von MEMS-Beschleunigungssensoren fur Verwendung im geodatischen Bauwerksmonitoring. 2013.

Rinner, K. 1978. Entwicklungstendenzen in der Ingenieurgeodäsie. *Allgemeine Vermessungs-Nachrichten.* 1978, S. 161 - 166.

Schweiger, Volker und Sternberg, Harald. 2014. Multi-Sensor-Systeme in der Ingenieurgeodäsie - Grundlagen und Überblick. *Schriftenreihe des DVW: Multi-Sensor-Systeme - Bewegte Zukuntsfelder.* 2014, Bd. 75.

Staiger, Rudolf. 2009. Instumente uind Messsysteme zur Beobachtung zeitabhängiger Größen. *Schriftenreihe des DVW: Zeitabhängige Messgrößen - Verborgene Schätze in unseren Daten.* 2009, Bd. 59.

Stralsund, Hansestadt. 2015. www.hansestadt-stralsund.de/de/bildergalerien/ruegenbruecke. *www.hansestadt-stralsund.de.* [Online] 01. 06 2015. [Zitat vom: 04. 01 2017.] http://www.hansestadt-stralsund.de/var/hansestadt-stralsund/storage/images/2012/bilder/ruegenbruecke/ruegenbruecke/20180-1-ger-DE/ruegenbruecke_lightbox_2012.jpg.

Thuselt, Frank und Gennrich, Felix Paul. 2013. *Praktische Mathematik mit MATLAB, Scilab und Octave - für Ingenieure und Naturwissenschaftler.* Heidelberg : Springer-Verlag, 2013.

Weigend, Michael. 2015. *Raspberry Pi programmieren mit Python.* Heidelberg : Verlagsgruppe Hüthig Jehle Rehm GmbH, 2015.

Wild-Pfeiffer, Franziska und Schäfer, Bernhardt. 2011. MEMS-Sensoren, auch für die Geodäsie. *zfv – Zeitschrift für Geodäsie, Geoinformation und Landmanagement.* 2011.

Witte, Schmidt. 2000. *Vermessungskunde und Grundlagen der Statistik für das Bauwesen.*

Stuttgart : Verlag Konrad Wittwer, 2000.

14 Tabellenverzeichnis

Tabelle 1.1: Arten von Deformationen 4
Tabelle 1.2 Übersicht über die geodätischen Monitoring-Bereiche 7
Tabelle 1.3: Übersicht der Anforderungen an das Monitoring [Staiger, 2009] 8
Tabelle 1.4: Einteilung von Multi-Sensor-Systemen [Schweiger, et al., 2014] 10
Tabelle 2.1: Kostenzusammenstellung 24
Tabelle 3.1: Geschwindigkeiten von USB 28
Tabelle 3.2: AD-Wandlung und Temperaturauflösung 34
Tabelle 3.3: AD-Wandlung und Beschleunigungsauflösung 34
Tabelle 3.4: AD-Wandlung und Beschleunigungsauflösung inklusive ADXL355 35
Tabelle 4.1: Beispiel für die Ergebnisdatei des Beschleunigungssensors 42
Tabelle 4.2 Anforderungen an Sensoren in verschiedenen Bereichen des ingenieurgeodätischen Monitorings 45
Tabelle 6.1: Erste Temperaturmessungen 64
Tabelle 6.2: Erste gespeicherte Beschleunigungswerte 64
Tabelle 6.3: Genauigkeiten des Temperatursensors 66
Tabelle 6.4: Genauigkeiten der Beschleunigungssensoren 67
Tabelle 7.1: Genauigkeiten des Beschleunigungssensors ADXL355 im Vergleich zum SBG-IG500E 76
Tabelle 8.1: Zusammenstallung gemessener Glocken- und Gebäudefrequenzen 80
Tabelle 10.1: Zusammenstellung der Zeitpunkte des Auftretens, der Dauer, der Amplituden und der Frequenzen 95
Tabelle 11.1: Bestimmung der Glockenfrequenzen 98
Tabelle 11.2: Zusammenstellung der Zeitpunkte des Auftretens, der Dauer, der Amplituden und der Frequenzen der Glockenschwingungen 102
Tabelle 11.3: Zusammenstellung der Zeitpunkte des Auftretens, der Dauer, der Amplituden und der Frequenzen der Mauerschwingungen 102
Tabelle 12.1: Variantenvergleich 106
Tabelle 12.2: Messobjekte und Auswertemethoden 107

15 Abbildungsverzeichnis

Abbildung 1.1: Rügenbrücke [Knopf, 2013] 1
Abbildung 1.2: Eislaufhalle in Bad Reichenhall [n-tv, 2011] 2
Abbildung 1.3: Teilgebiete der Ingenieurgeodäsie 3
Abbildung 1.4: Messpunkte an einer Windenergieanlage 5
Abbildung 1.5: kontinuierliche Deformationsmessung [Witte, 2000] 6
Abbildung 1.6: Bestandteile und Eigenschaften eines Structural-Health-Monitoring-System [HBM, 2018] 9
Abbildung 1.7: Prinzip der Messung von sechs Freiheitsgraden [Ionescu, 2016] 11
Abbildung 1.8: Aufbau eines MEMS-Beschleunigungssensors [Elektronik, 2016] 11
Abbildung 1.9: Funktionsweise eines MEMS-Beschleunigungssensors [Elektronik, 2016] ... 12
Abbildung 2.1: Pinbelegung Linker Temperature Temperatursensor 13
Abbildung 2.2: Pinbelegung Linker 3-Axis-Accelerometer 14
Abbildung 2.3: EVAL-ADXL354 Evaluation-Board 15
Abbildung 2.4: Schaltbild EVAL-ADXL354 [Analog Devices, 2018] 15
Abbildung 2.5: EVAL-ADXL355 Evaluation-Board 16
Abbildung 2.6: Schaltbild EVAL-ADXL355 [Analog Devices, 2018] 16
Abbildung 2.7: Erweiterungsmodul Linker Kit Baseboard 17
Abbildung 2.8: Mikrocontroller Arduino Mega/ Uno/ Nano 18
Abbildung 2.9: Mikrocontroller Arduino Nano 19
Abbildung 2.10: Arduino Due 19
Abbildung 2.11: "Bluepill" Mikrocontroller 20
Abbildung 2.12: Bluepill, Maple Mini und Arduino Nano im Größenvergleich 21
Abbildung 2.13:Pinbelegung des Maple Mini Boards [Huiskamp, 2018] 21
Abbildung 2.14: Raspberry Pi mit und ohne Gehäuse 22
Abbildung 2.15: professioneller Beschleunigungssensor SBG IG-500E 23
Abbildung 3.1: D-Sub Stecker einer RS232 Schnittstelle [Hering, et al., 2017] 25
Abbildung 3.2: Spannungsverlauf einer Datenübertragung der RS232 Schnittstelle [Hering, et al., 2017] 26
Abbildung 3.3: 3 Kabel-Lösung zur Datenübertragung 27
Abbildung 3.4: USB Stecker [Hering, et al., 2017] 27
Abbildung 3.5: Beispiel einer I²C Bus Ansteuerung [Lerch, 2016] 28
Abbildung 3.6: Ablauf der Datenübertragung auf dem I²C Bus [Hering, et al., 2017] 29
Abbildung 3.7: Versuchsaufbau mit Linker Temperature und Arduino Nano 30
Abbildung 3.8: Versuchsaufbau mit Linker 3-Axis-Accelerometer und Arduino Nano 30

Abbildung 3.9: Sketch des Arduino Programmes zum Auslesen des Temperatursensors 31
Abbildung 3.10: ADXL354 Beschleunigungssensor und Arduino Due 32
Abbildung 3.11: ADXL354 Sensor mit ADS1115 und Arduino Nano 32
Abbildung 3.12: 16-Bit AD-Wandler ADS1115 33
Abbildung 4.1: Übersicht der Entwicklung der Programmiersprache C [Ingenieurbüro Heimann, 2018] 37
Abbildung 4.2: vorläufiger Programmablaufplan zum Auslesen von MEMS-Sensoren 40
Abbildung 4.3: PyScripter-Oberfläche und erster Programmentwurf 41
Abbildung 4.4: Syntax zum Öffnen des seriellen Portes 42
Abbildung 4.5: Probleme mit dem Start des Empfangs der seriellen Daten 43
Abbildung 4.6: überarbeiteter Programmablaufplan 44
Abbildung 4.7: wasserdichtes Gehäuse 46
Abbildung 4.8: USB Anschluss am Gehäuse 47
Abbildung 4.9: Stromversorgung über den Raspberry Pi 48
Abbildung 4.10: !2 Volt zu 5 Volt Konverter 48
Abbildung 4.11: fertiger Prototyp 49
Abbildung 5.1: Positionen eines GNSS Rovers, ein Beispiel für eine Zeitreihe 50
Abbildung 5.2: Beispiel für einen Ausreißer in einer Zeitreihe 52
Abbildung 5.3: Beispiel für einen Sprung in den Messdaten 52
Abbildung 5.4: Beispiel für Offset (grüne Gerade) und linearen Trend (rote Gerade) innerhalb einer Zeitreihe 53
Abbildung 5.5: Zeitreihe mit Datenlücken 53
Abbildung 5.6: FOURIER-Approximation eines Rechtecksignals 55
Abbildung 5.7: FFT eines Tonleiterwechsels 58
Abbildung 5.8: Auswahl von Fensterfunktionen [K, 2005] 59
Abbildung 5.9: STFT eines Tonleiterübergangs in zwei- und dreidimensionaler Darstellung 60
Abbildung 5.10: Skalierungs- und Translationsverhalten 'cmorl' [Thuselt, et al., 2013] 62
Abbildung 5.11: Wavelet Skalogramm 63
Abbildung 6.1: Erhöhung der Genauigkeit durch Mittelwertbildung (Integration über die Anzahl) 66
Abbildung 6.2: Standardabweichungen der drei Achsen des Beschleunigungssensors 67
Abbildung 6.3: Drift des Sensors innerhalb der ersten 150 Sekunden 68
Abbildung 6.4: zeitliche Äquidistanz des Beschleunigungssensors 69
Abbildung 6.5: Testvorrichtung zur Simulation von Bauwerksdeformationen 70
Abbildung 6.6: detektierte Schwingung 70
Abbildung 6.7: Autokorrelationsfunktion mit Konfidenzintervallen 71

Abbildung 6.8: Amplitudenspektrum der Schwingung 71
Abbildung 6.9: Amplituden- und Frequenzbestimmung mit dem Accel-Board 72
Abbildung 7.1: Kalibrierung des Temperatursensors 73
Abbildung 8.1: Messaufbau 78
Abbildung 8.2: Sensoraufbau unterhalb des Glockenstuhls 79
Abbildung 8.3: Frequenzanalyse GNSS 80
Abbildung 8.4: Frequenzanalyse Tachymeter Leica TS30 81
Abbildung 8.5: Frequenzanalyse Neigungssensor Leica Nivel210 81
Abbildung 8.6: Frequenzanalyse SBG 82
Abbildung 8.7: Frequenzanalyse des neuentwickelten Sensors beim Läuten der Marienglocke 82
Abbildung 8.8: Amplitudenspektrum der Gebetsglocke 83
Abbildung 9.1: Treppe im Eingangsbereich des Hauses 2 der Hochschule Neubrandenburg 84
Abbildung 9.2: freitragende Konstruktion der Treppe 85
Abbildung 9.3: Anbringung des Sensors am obersten Stufenelement 85
Abbildung 9.4: Messdaten der Treppe vom Sonntag, den 19.08.2018 86
Abbildung 9.5: Messwerte der Treppe am Montag, den 20.08.2018 86
Abbildung 9.6: Messwerte einer Achse und Temperaturverlauf für den 20.08.2018 87
Abbildung 9.7: Erschütterung der Treppe und Kurzzeitfouriertransformation 88
Abbildung 10.1: Lage des Vermessungsbüros in Waren (Müritz) 89
Abbildung 10.2: Übersicht Messobjekt und Bahnstrecke 90
Abbildung 10.3: Sensor mit Raspberry Pi als Ausleserechner 91
Abbildung 10.4: Außenansicht des Gebäudes des Vermessungsbüros des ÖbVI Herrmann mit dem Einbauort des Sensors 91
Abbildung 10.5: Sensoranbringung und Auswerterechner auf dem Fensterbrett 92
Abbildung 10.6: Plot der Beschleunigungswerte 92
Abbildung 10.7: Amplitudenspektrum über eine Stunde 93
Abbildung 10.8: STFT über einen Zeitraum von 120 Sekunden 94
Abbildung 11.1: St. Peter Kirche in Altentreptow 96
Abbildung 11.2: Innenstadt von Altentreptow mit der St. Peter Kirche 97
Abbildung 11.3: Die drei Glocken der St. Peter Kirche in Altentreptow 98
Abbildung 11.4: Sensorpositionen im Glockenturm der St. Peter Kirche 99
Abbildung 11.5: Plot der Beschleunigungswerte der y-Achse (15 Minuten) 100
Abbildung 11.6: Amplitudenspektrum über eine Viertelstunde 100
Abbildung 11.7: STFT der Viertelstundenmessung 101

Abbildung 12.1: Sensorgehäuse mit Mikrocontroller, Anschlüssen und MEMS-Sensor (verdeckt)..... 103
Abbildung 12.2: Variante mit Tablet als Auswerterechner 104
Abbildung 12.3: Variante mit Raspberry Pi und 7 Zoll Display 105
Abbildung 12.4: Variante mit 3 Zoll Display, hier mit einem Akkumulator als Spannungsversorgung. 105
Abbildung 12.5: Kurzzeitfouriertransformation des Glockenspiels der Marienkirche Neubrandenburg 108
Abbildung 12.6: Turmüberwachung der St. Peter Kirche Altentreptow mit Fokus auf der Frequenzauflösung..... 109
Abbildung 12.7: Turmüberwachung der St. Peter Kirche Altentreptow mit Fokus auf der Zeitauflösung 109