HAW HAMBURG

BACHELORTHESIS

Daniel Kraft

Data Analysis of the optical synchronization system

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Daniel Kraft

Data Analysis of the optical synchronization system

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Marina Tropmann-Frick
Zweitgutachter: Prof. Dr. Martin Schultz

Eingereicht am: 18. März 2021

Daniel Kraft

Thema der Arbeit

Data Analysis of the optical synchronization system

Stichworte

Data Analysis, Data Science, DESY, European XFEL

Kurzzusammenfassung

Die Arbeit umfasst die Analyse der Daten der Links des European XFEL. Diese Links sind für die Synchronisation der Subsysteme im XFEL zuständig. Die Datenanalyse umfasst das Sammeln der Daten, sowie das Filtern und die Analyse der daraus resultierenden Daten. Hierbei werden Korrelationen zwischen den Links untersucht und es werden auch die Klimadaten aus dem Beschleunigerteil des XFEL mit einbezogen. Als nächster Schritt wird die Clusteringanalyse auf den Links ausgeführt.

Daniel Kraft

Title of Thesis

Data Analysis of the optical synchronization system

Keywords

Data Analysis, Data Science, DESY, European XFEL

Abstract

This work contains the analysis of the data from the links of the European XFEL. These links are responsible for the synchronization of the subsystems in the European XFEL. The analysis includes the collecting of the data, the filtering of the relevant parts of the data and the analysis of these parts of the data. For the analysis correlations between the data from the links are used. Correlations between climate data and the data from the links are also used. The next step of the analysis is the clustering of the data.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die Arbeit wurde in Zusammenarbeit mit dem Deutschen Elektronen-Synchrotron, kurz DESY, erstellt. So wurde mir die Möglichkeit gegeben, dass ich die Daten von dem European XFEL nutzen kann, um sie genauer im Rahmen meiner Arbeit zu untersuchen. Zusätzlich wurden die Ergebnisse der Analyse und eventuell aufgetretene Probleme in wöchentlichen Meetings besprochen, um so auf eventuell Ursachen von Auffälligkeiten zu kommen und Probleme und Unklarheiten zu lösen.

1.1 Motivation

Im alltäglichen Leben, in Unternehmen und auch in der Forschung fallen sehr viele Daten an und Ziel der Data Analysis ist es diese Daten zu sammeln, filtern und zu untersuchen. Durch diese Daten und anhand der Analyse der Daten ist es uns möglich bessere Entscheidungen, basierend auf den Daten, zu treffen.

So kann durch die Data analysis im Bereich der Predictive Maintenance früher auf Probleme eingegangen werden. So zeigen sich Probleme bei Maschinen und Defekte schon vorher in den Daten, bevor diese Maschinen einen Totalausfall erleiden und eine komplette Reparatur benötigen. Wenn man dieses Phänomen schon vorher in den Daten erkennt, kann man eingreifen, bevor die Maschinen einen Totalausfall erleiden, und so ist es dem Unternehmen möglich Geld für die Reparatur zu sparen und die Ausfallzeiten der Maschinen geringer zu halten. Mit diesem Feld befasst sich der Bereich der Predictive Maintenance (Deutsch: Vorausschauende Wartung) [1].

Der European XFEL ist der aktuell größte, operierende lineare Teilchenbeschleuniger. Der Beschleuniger bietet Messungen, welche ein Timing mit einem Fehler in einem Bereich von Femtosekunden benötigen. Die Links dienen dabei als Startpunkt für die Datenanalyse, da sie das

Synchronisationssignal für die Subsysteme bereitstellen. Die Daten der Links zeigen langsame Drifts, welche hauptsächlich durch Klimaveränderungen entstehen.

1.2 Zielsetzung der Arbeit

Diese Arbeit umfasst die Datenanalyse der Daten, die die Links bereitstellen. Hierbei geht es darum die Daten zu sammeln, aufzubereiten und zu analysieren. Dies umfasst die explorative Analyse der Daten, sowie Clustering Versuche und auch das Untersuchen der Korrelationen. Die Korrelationen werden auch unter dem Einbeziehen der Klimadaten untersucht. Ebenso werden die Daten und die Ergebnisse der Analyse graphisch dargestellt.

1.3 Gliederung der Arbeit

Die Arbeit ist in drei Teile untergliedert. Im ersten Teil gebe ich einen kurzen Überblick über DESY, die Beschleuniger und auch einen Überblick über den European XFEL und den Aufbau des betrachteten Beschleunigers.

Im zweiten Teil der Arbeit gehe ich auf die durchgeführten Experimente ein. Dabei starte ich mit der explorativen Analyse. Danach gehe ich auf die verschiedenen Analysen ein, beginnend mit der Bildung von Korrelationen. Zusätzlich gebe ich noch einen kurzen Überblick über die Art der Daten, technische Informationen bezüglich der Implementierung und einen Überblick über die verwendeten Funktionen. Zum Schluss gehe ich noch auf die Probleme ein, welche mir im Laufe der Arbeit begegnet sind.

Im letzten Teil fasse ich meine erarbeiteten Ergebnisse noch einmal zusammen und gehe hierbei auch auf die wichtigsten Ergebnisse aus der Analyse ein. Hierbei gehe ich noch darauf ein, wie man weiter vorgehen könnte.

2 Grundlagen

2.1 Überblick zu DESY

Das Deutsche Elektronen-Synchrotron, kurz DESY, ist ein Forschungszentrum, welches sich mit der naturwissenschaftlichen Grundlagenforschung beschäftigt. DESY liegt in Hamburg Bahrenfeld. Die Forschung verteilt sich auf vier Schwerpunkte [2]:

1. Beschleuniger (Durch die Beschleuniger können Wissenschaftler die Struktur und die Funktion von Materie näher untersuchen)
2. Forschung mit Photonen (Durch das spezielle Röntgenlicht der Lichtquellen werden atomare Strukturen und Reaktionen im Nanokosmos sichtbar)
3. Teilchenphysik (Umfasst die Untersuchung der grundlegenden Bausteine und Kräfte im Universum)
4. Astroteilchenphysik (Gammastrahlung und Neutrinos werden dazu genutzt, um hochenergetische Prozesse in unserem Universum zu verstehen)

Die Datenanalyse im folgenden Teil meiner Arbeit beschäftigt sich hier mit den resultierenden Daten aus den Beschleunigern, in meinem Fall kommen die Daten von dem European XFEL, wobei XFEL für X-ray free-electron lasers steht. Die Beschleuniger lassen kleine, elektrisch geladene Teilchen auf fast Lichtgeschwindigkeit beschleunigen, was in verschiedenen Forschungsdisziplinen genutzt werden kann. So kann man diese Teilchen frontal aufeinanderprallen lassen, um so die Materie untersuchen zu können. Ebenso können Beschleuniger das hellste Röntgenlicht der Welt erzeugen, damit man so unterschiedliche Materialien genauer untersuchen kann. Ebenso können diese Teilchen auch zur Krebstherapie genutzt werden. DESY gehört weltweit zur Spitze von den Beschleunigerzentren. Verschiedene Beschleuniger gehören zu DESY [3]:

1. FLASH
2. European XFEL
3. ILC
4. PETRA III
5. PITZ
6. REGAE

2.2 Überblick zu European XFEL

Der European XFEL kann ultrakurze Laserlichtblitze im Röntgenbereich erzeugen. Dies geschieht 27.000-mal in der Sekunde und diese Lichtblitze sind milliardenfach intensiver als die der besten herkömmlichen Röntgenquelle. Der European XFEL ist weltweit einzigartig und eröffnet neue Möglichkeiten für die Forschung und die Industrie. Der XFEL befindet sich in unterirdischen Tunnelröhren und weist eine Gesamtlänge von circa 3,4 km auf. Er verläuft von dem DESY-Gelände bis nach Schnenefeld in Schleswig-Holstein. Dort befindet sich der Forschungscampus mit einer großen Experimentierhalle. Der XFEL funktioniert so, dass er Elektronen auf nahezu Lichtgeschwindigkeit beschleunigt. Diese Elektronen werden dann durch spezielle Magneten auf Slalombahnen gezwungen. Dadurch entsenden die Eletronen extrem kurze und starke Röntgenblitze. Diese Blitze haben Lasereigenschaften. Im Gegensatz zu ähnlich großen Röntgenlasern in den USA und Japan, schafft der XFEL viel mehr Röntgenblitze pro Sekunde durch supraleitende Materialien. Supraleitende Materialien sind Materialien, welche unterhalb einer bestimmten Temperatur keinen elektrischen Widerstand aufweisen. Diese hohe Anzahl an Röntgenblitzen kann für manche Experimente entscheidend sein. Der European XFEL ist seit 2017 im Betrieb und bietet eine unterirdische Experimentierhalle mit Platz für zehn Messplätze [4].

2.3 Aufbau des European XFEL

2.3.1 Injektor

Der Injektor befindet sich in einem Gebäude in dem westlichen Part von dem DESY Campus. Der Injektor liegt in einer Tiefe von 27m. Durch eine Photokathode werden die Elektronen in einem Laser rausgeschossen und in dem Injektor werden die Elektronen auf bis zu 120 MeV (Megaelektronenvolt) beschleunigt. Die Ruhenergie von den Elektronen liegt bei 0,511 MeV. Durch den Injektor wird auch eine notwendige Qualität sichergestellt, wie zum Beispiel eine niedrige Emittanz (Ergebnis des Produktes aus Winkeldivergenz und Querschnittsfläche). Dies ist sehr wichtig für das weitere Vorgehen bei den Experimenten im XFEL. Die Gesamtlänge des Injektors beträgt 66m. Anschließend werden die Elektronen in den linearen Beschleuniger weitergeleitet.

2.3.2 Linearer Beschleuniger

Der Beschleuniger befindet sich in einer Tunneleinheit im Untergrund. Diese Tunneleinheit schließt an den Injektor im Untergrund an. Der Tunnel hat einen Durchmesser von 5,2m und der Beschleuniger endet nach 1,6km Länge. 116 Beschleunigermodule machen den Großteil der Länge aus. Diese Beschleunigermodule dienen dazu die Elektronen auf eine Energie von bis zu 20 Gigaelektronenvolt zu bringen. Ein weiterer wichtiger Teil sind die zwei Bunchkompressor, welche durch Magneten die Länge des Elektronenbündel auf 55µm bringen. Dies entspricht einer Dauer von weniger als 200 Femtosekunden. Der erste Bunchkompressor liegt an dem Punkt, an dem die Energie von den Elektronen bei 0,5 GeV liegt, der Zweite liegt an dem Punkt, an dem die Energie der Elektronen bei 2 GeV liegt. Die letzten 0,4km des Beschleunigers haben die Aufgabe, dass der Strahl gut ausgerichtet und kollimiert ist für den weiteren Teil des XFEL.

2.3.3 Beam distribution system

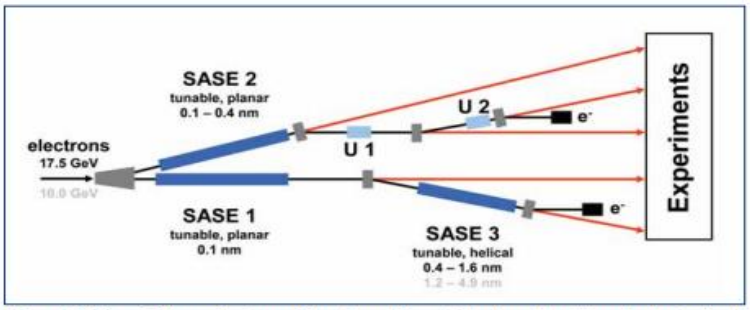

Figure 3.1.2 *Schematic view of the branching of electron (black) and photon (red) beamlines through the different SASE and spontaneous emission undulators. Electron beamlines terminate into the two beam dumps, photon beamlines into the experimental hall.*

Abbildung 1 Schematische Überischt über die Undulatoren und SESE im XFEL aus [5]

Dadurch dass die Einrichtung für parallele Experimente gedacht ist, müssen die Elektronenbündel auf die verschiedenen Stationen verteilt werden. Die Nutzer an den verschiedenen Stationen haben verschiedene Anforderungen an die Elektronenbündel, deswegen müssen diese auch an den jeweiligen Nutzer angepasst werden. Durch sogenannte „fast kicker“ werden die Elektronenbündel in die Undulatoren SASE1 oder SASE2 geleitet. Diese weisen eine Länge von 200m auf. Darauffolgend können die Elektronenbündel durch diese Transfertunnel SASE1 und SASE2 zu weiteren Undulatoren.

2.3.4 Undulatoren

Die Undulatoren sind dazu da, um die Brillanz der Strahlung zu erhöhen. Als Brillanz der Strahlung versteht man die Bündelung eines Strahles. Die XFEL Undulatoren unterscheiden sich darin, dass sie länger sind als die konventionellen Undulatoren SASE1 und SASE2 sind. Um die sehr strengen Toleranzen für Ausrichtung und Uniformität zu erfüllen bestehen die Undulatoren aus 5m langen Modulen mit 1,1m langen Überschneidungen. In den letzten Modulen ist Equipment für Diagnostik und Korrektionen für die Elektronenflugbahn.

2.3.5 Photon beamlines

Nach dem die Photonen die Elektronenbündel verlassen am Ende der Undulatoren, müssen sie immer noch die restliche Strecke zu den Experimentierhallen zurücklegen. Dies geschieht in Tunneln mit einem Durchmesser von 4,5m. Diese Tunnel leiten die Photonen dann zu der Experimentierhalle. Auf dem Weg zur Halle wird der Strahl noch nach Belieben formt und kollimiert.

2.3.6 Experimentierstationen

Man kann das Equipment für die Stationen nicht zusammenfassen, da sie je nach wissenschaftlichem Nutzen variieren. Die Stationen für die Experimente befinden sich im Untergrund in einer Halle mit einer Fläche von 50 x 90m² und einer Höhe von 14m. Für jede Station ist eine Fläche von 15 x 42m² vorgesehen.

2.3.7 Schenefeld Campus

Durch die erfolgreichen Experimente von dem European XFEL wird das wissenschaftliche Interesse an dem Projekt und an den Experimenten größer. Die dadurch teilnehmenden Wissenschaftler können dann auch an dem Campus an Workshops, Konferenzen besuchen und sich einfach nur mit wissenschaftlichen Kollegen austauschen. Ebenso ist es auch interessant für Exkursionen von Schulen und Universitäten [5].

2.4 Methoden der Datenanalyse

2.4.1 Korrelation und Autokorrelation

In vielen Studien werden Korrelationen zur Datenanalyse benutzt. Bei der Korrelation geht es um das Untersuchen der Abhängigkeit zwischen zwei Variablen. Der Korrelationkoeffizient bewegt sich im Wesentlichen hierbei zwischen -1 und +1, wobei sich +1 auf eine perfekte positive Korrelation bezieht, -1 auf eine perfekte negative Korrelation und 0 bezieht sich hierbei auf gar keine Korrelation zwischen zwei Variablen, wobei dies sehr selten der Fall ist, dass die Korrelation zwischen zwei Variablen 0 ist [6]. Korrelationen zwischen zwei Variablen können einen weiteren Anhaltspunkt für weitere Untersuchungen bilden. Die Autokorrelation

beschreibt hier einen Sonderfall, da es hierbei um die Korrelation einer Variablen mit sich selbst geht.

2.4.2 Bereinigung des Mittelwerts

Die Bereinigung des Mittelwerts war bei den Daten der TIMING Kanäle notwendig, da der Offset dieser Daten sehr hoch war.

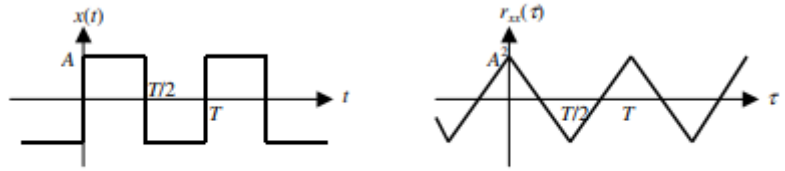

Bild 8.8 Rechteckfunktion und ihre Autokorrelationsfunktion

Abbildung 2 Korrelation der Rechtecksfunktion aus [7]

Der Screenshot zeigt die Notwendigkeit der Mittelwertsbereinigung. Dadurch dass der Offset der TIMING Daten hoch ist (vierstelliger Bereich), ähnelt der Ausschnitt einer Woche aus den TIMING Daten einer Rechtseckfunktion, weswegen das Ergebnis der Korrelation eine Dreiecksfunktions darstellt. Dies konnte man beheben, indem man den Durchschnitt der Daten über eine Woche von jedem einzelnen Datenpunkt abzieht, um den Offset der Daten rauszurechnen.

2.4.3 Fast Fourier transform

Die Methode correlate von scipy [16] stellt die performante Methode zur Bestimmung der Korrelationen über die Fast Fourier transform (FFT) zur Verfügung. Diese Methode funktioniert nur bei numerischen Arrays. Die Fast Fourier transform stellt eine effiziente Möglichkeit zur Berechnung der diskreten Fourier Transformation (DFT) einer Zeitreihe dar. Durch die DFT können die Korrelationen auf eine performantere Weise berechnet werden, solange es sich bei den Daten um numerische Arrays handelt [8].

2.4.4 Clustering

Das Clustering dient dazu Gruppen anhand von verschiedenen Features in Daten zu finden. Clustering ist ein wichtiger Bestandteil in der Datenanalyse, um Muster in Daten erkennen zu

können. Der verwendete Algorithmus zur Datenanalyse in der folgenden Arbeit ist der K-Means-Algorithmus. Bei dem K-Means-Algorithmus handelt es sich um ein partitioniertes Clusteringverfahren. Die Besonderheit an partitionierten Clusteringverfahren ist, dass die Anzahl der Cluster festgelegt werden muss vor der Ausführung. Aufgrund von betrachteten Erwartungswerten und in der gemeinsamen Besprechung wurde sich für die Verwendung von 2, 3 und 4 Clustern entschieden. Der K-Means-Algorithmus dient dazu den Abstand der Daten zu ihrem Clustermittelpunkt zu minimieren [9].

3 Experimente

3.1 Beschreibung der Experimente

Bei den Experimenten geht es darum, sich die Daten von den Links des XFEL (X-Ray Free Electron Laser) anzugucken. Diese Links erstrecken sich über die Gesamtlänge des Tunnels des XFEL und variieren in einer Länge von 43 Metern bis zu 3.515 Metern. Diese Links lassen sich in drei Kategorien unterteilen: 1. Die Links, die im Injektor enden, welche sich zwischen einer Länge von 43 Metern bis 91 Metern bewegen. 2. Die Links, die im Tunnel enden, welche auch bei der späteren Betrachtung der Klimadaten die wichtigste Rolle spielen, da die aussagekräftigen Klimadaten aus dem Tunnel gemessen wurden. Diese Links bewegen sich zwischen einer Länge von 160 Metern bis 1.950 Metern. 3. Die Links, die bis nach Schenefeld laufen und eine Länge von 3.502 und 3.515 Metern aufweisen. Der Unterschied der beiden langen Links liegt daran, dass sie bis Osdorfer Born parallel verlaufen und sich ab dem Punkt verschiedene Wege nehmen. Nach der Rücksprache mit DESY wurde sich darauf geeinigt meine Datenrecherche auf die AMC Controller festzulegen. Daten werden auf einer Frequenz von 10 Hz ausgelesen. Der Regler arbeitet in einer Geschwindigkeit von ungefähr 3 MHz, ausgelesen werden davon aber nur 2^{15} Samples und davon wird dann der Mittelwert alle 10 Hz gebildet. Mit diesem Mittelwert habe ich mich dann im Laufe meiner Arbeit hauptsächlich beschäftigt. Die Experimente haben damit gestartet, dass ich mich damit beschäftigt habe, mir erstmal einen Überblick über die Daten zu verschaffen. Diese Daten kann ich mit der Methode getData über das PythonDAQClientInterface [10] auslesen.

Ebenso habe ich mich mit der graphischen Betrachtung der Plots beschäftigt. In der allgemeinen Betrachtung der Daten wurden vorerst keine Auffälligkeiten gefunden. Darauf habe ich mich entschieden die Daten gezielter zu betrachten und mir wurde vorerst von DESY geraten mir vier besondere Links anzuschauen. Zwei von den Links sind besonders lang (3.502m und

3.515m) und zwei sind besonders kurz (43m und 84m). Dabei habe ich mir den Zeitraum von dem 03.10.2020 bis zum 11.10.2020 angeschaut und dabei hat sich herausgestellt, dass es einen Zeitraum gibt, in dem es keine Daten gibt. Dies ist vermutlich darauf zurückzuführen, dass es einen Ausfall in dem DAQ gab und in dem Zeitraum keine Daten übermittelt wurden.

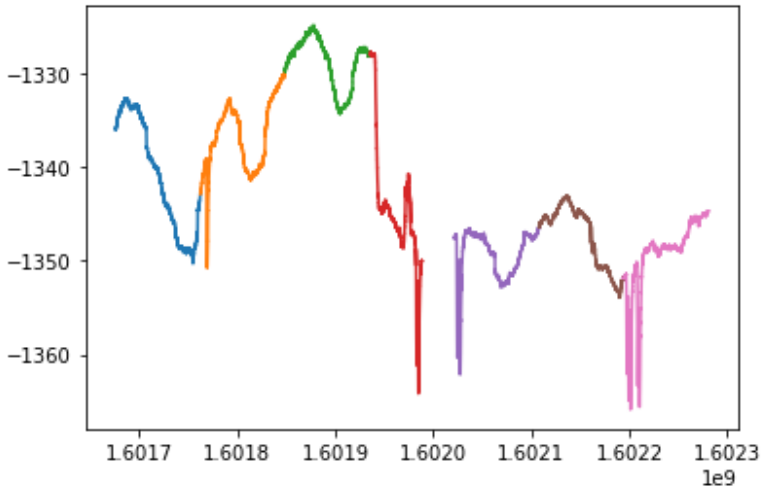

Abbildung 3 Datenloch innerhalb der Daten

Auf dem Screenshot sieht man das oben angesprochene Datenloch, weswegen sich gegen die Betrachtung dieses Zeitraums entschieden wurde und ich mich für einen anderen Zeitraum entschieden habe. Aus dem Grund habe ich mich in meiner Recherche vorerst für die Woche vom 02.11.2020 bis zum 08.11.2020 entschieden.

Dadurch, dass ich mit so vielen Daten arbeite, musste ich mir Gedanken über die langzeitige Speicherung von den Daten mache, da es mittlerweile zu langwierig ist, die Daten jedes Mal auszulesen, bevor ich dann weiterarbeiten kann. Dabei habe ich mich für h5py [11] entschieden, um die Daten in einem Backup zu speichern. Zur Auswahl standen noch hdfStore [12] und hickle [13]. Die Probleme bei den beiden Möglichkeiten waren, dass hdfStore nicht funktionsfähig war in dem Notebook und hickle war nicht performant genug, wodurch sich diese Möglichkeit nicht gelohnt hat. Durch h5py hat sich die Möglichkeit ergeben die Daten innerhalb von Sekunden zu speichern und aus dem Backup wieder auszulesen.

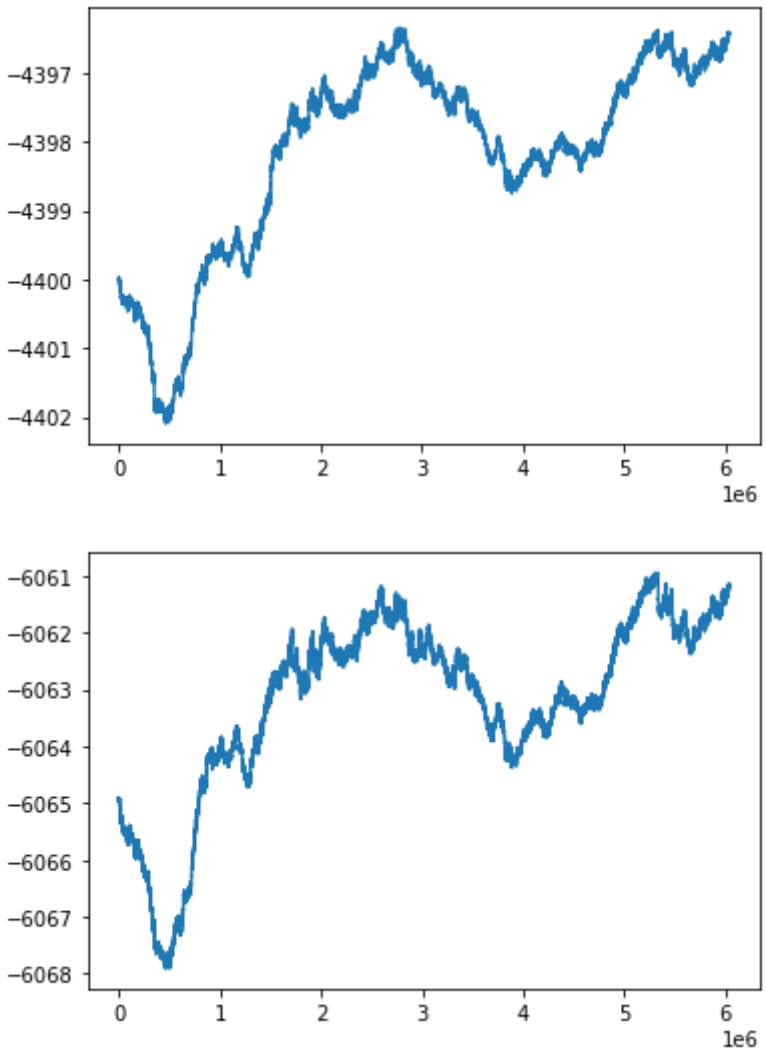

Abbildung 4 Vergleich zwischen den Links 10.0 und 9.3

Für diese Woche hatte ich mich vorerst auch für die vier oben beschriebenen Links entschieden. Hierbei hat sich herausgestellt, dass sich die beiden kurzen Links sehr ähnlich verhalten, da diese parallel zueinander verlaufen und dadurch, dass sie sehr kurz sind, sind sie allgemein weniger anfällig für Störungen.

Bei der Betrachtung der Graphen fällt aus, dass hier noch die Linuxtimestamps die x-Achse bilden. Diese habe ich im weiteren Verlauf der Arbeit zu normalen lesbaren Timestamps geändert. Neben den CTRL OUT Kanälen, mit denen ich mich beschäftigt hatte, habe ich mich auch mit den TIMING Kanälen beschäftigt. An den TIMING Kanälen ist der Vorteil, dass die

Course Tuning Schritte rausgerechnet werden. Die CTRL OUT Kanäle sind auf einen gewissen Grenzbereich festgelegt. Wird ein bestimmter Wert erreicht, fängt der Motor an, dagegen an zu regeln, was sich auch an den Daten bemerkbar macht.

Nun ergibt sich der nächste Schritt in der Analyse, in dem es darum geht, Korrelationen zwischen den verschiedenen CTRL OUT Kanälen und den verschiedenen TIMING Kanälen durchzuführen. Hierbei wurde vorerst die Methode correlate von numpy [14] benutzt. Aus Performancegründen habe ich mich dazu entschieden bei der Korrelation von den Daten nur jedes 100. Element in die Berechnung mit einzubeziehen, da sich im Vergleich am Endergebnis nicht ändert und es zeitlich nicht lohnend ist, wenn nachher Korrelationen über alle Links berechnet werden.

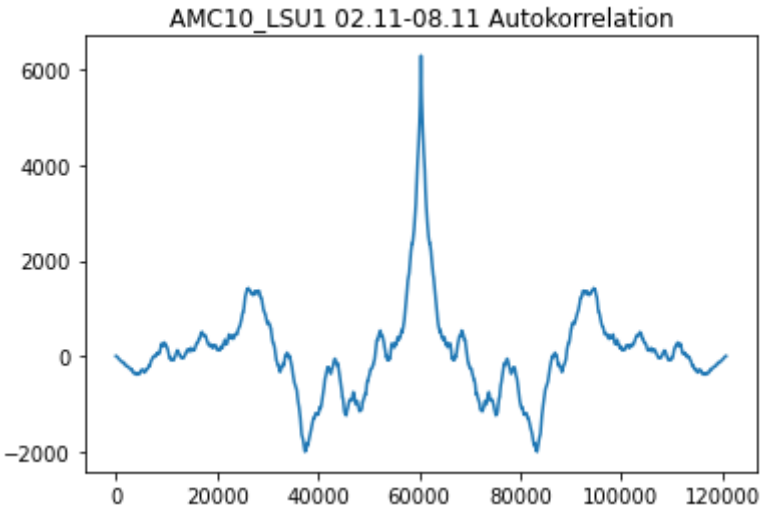

Abbildung 5 Autokorrelation für den Link 10.1

Hier hat sich ein weiteres Problem herausgestellt. Bei den Korrelationen mit den TIMING Kanälen hat sich bei allen Korrelationen ein pyramidenartiges Bild ergeben und man so kein vernünftiges Bild für die Korrelationen ablesen konnte. Wie schon oben beschrieben, ergibt sich bei der Korrelation der TIMING Daten eine Dreiecksfunktion. Das Problem war, dass der Offset von den Timingdaten zu groß ist und so die TIMING Daten einer Rechtecksfunktion ähneln. Dieses Problem ließ sich durch eine Bereinigung des Mittelwertes beheben. Den Mittelwert habe ich bereinigt, indem ich wochenweise den Mean über die Daten gebildet habe und dann

von jedem einzelnen Datenwert den Mean abgezogen habe. Auf dem Screenshot sieht man beispielsweise die Autokorrelation für den Link 10.1 ohne die Mittelwertsbereinigung, da hier der Offset durch die Daten nicht so groß war und mir dieses Problem nicht direkt in die Augen gesprungen ist, weswegen mir die Korrelationen für die CTRLOUT Kanäle noch normal erschienen. Als Autokorrelation versteht man hier die Korrelation der Daten mit sich selbst. Später folgen noch die Kreuzkorrelationen, wobei ich mich dabei auf die Korrelationen der verschiedenen untereinander beziehe.

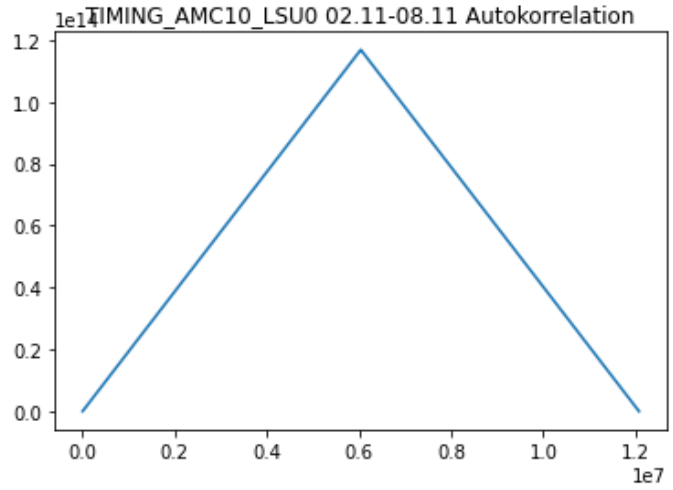

Abbildung 6 Autokorrelation vor Mittelwertsbereinigung

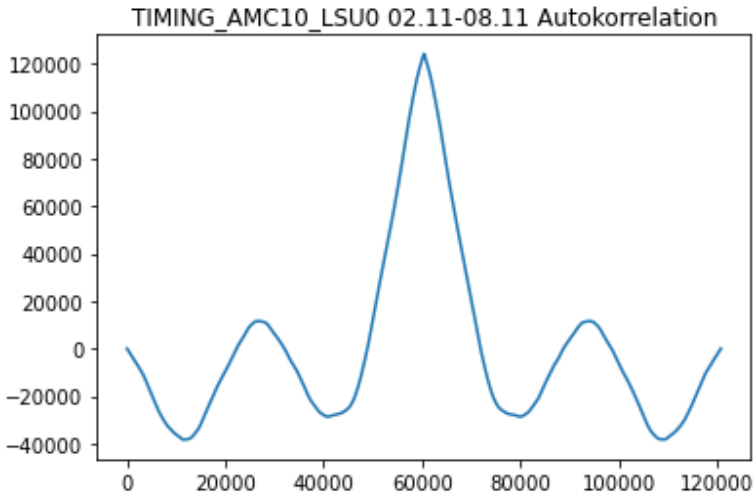

Abbildung 7 Autokorrelation nach Mittelwertsbereinigung

Hier sieht man im Vergleich einmal die Autokorrelation ohne die Mittelwertsbereinigung (erster Screenshot) und nach der Mittelwertsbereinigung (zweiter Screenshot). Im Gegensatz zu der Korrelation mit dem CTRLOUT Kanal ist hier der Offset in den Daten sehr viel größer, weswegen sich nur bei den TIMING Kanälen dieses Bild ergeben hat. Hieraus hat sich auch ein weiteres Problem ergeben, dass die Methode correlate von numpy [14] keine gute Performance aufgewiesen hat bei der Menge an Daten, die ich als Input genommen habe. So hatte ich Stunden auf Ergebnisse gewartet, die ich dann im Endeffekt nicht nutzen konnte und es war auch nicht direkt meine erste Intention, dass das Problem durch die Mittelwertsbereinigung behoben werden kann.

Der nächste Schritt war nun das Einbeziehen der Daten von allen Links (18 Links insgesamt). Der Zeitraum blieb vorerst noch der Gleiche. Beim Betrachten der Rohdaten dieser Links ist aufgefallen, dass einige der Daten Oszillationen in einem 30 Minuten Intervall aufweisen.

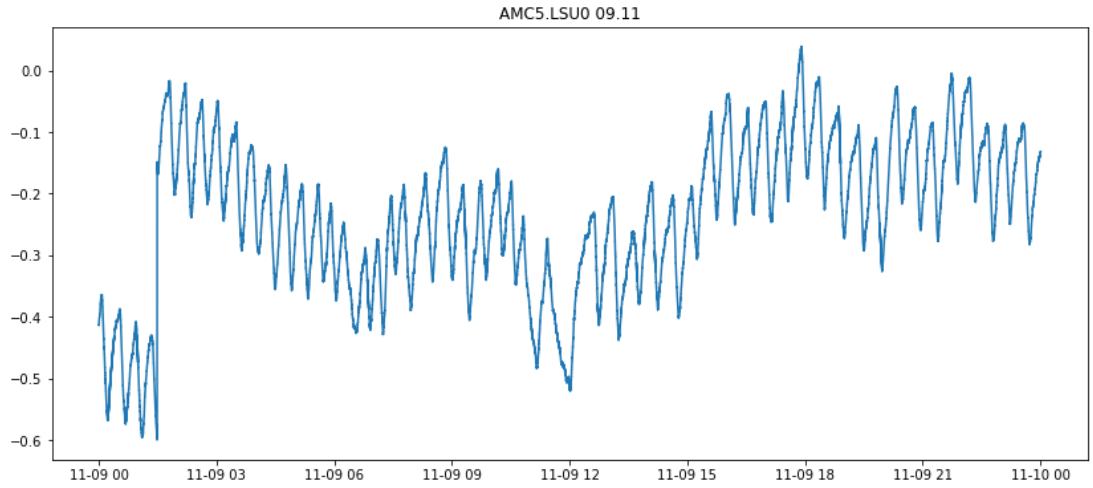

Abbildung 8 Oszillationen in 30-minütiger Periode

Eine Ursache für die Oszillationen in einer 30-minütigen Periode wurde auch in der gemeinsamen Besprechung nicht gefunden. Zur besseren Veranschaulichung der Zusammenhänge zwischen den Links habe ich mich für eine Heatmap entschieden, um die Korrelationen zwischen den 18 Links abzubilden. Diese Heatmap habe ich als Dreieck dargestellt, da die Heatmap symmetrisch ist und deswegen die eine Hälfte redundant ist. Bevor ich die Heatmaps erstellen konnte, musste ich meine Daten vorher in ein Dataframe umgewandelt. Hierbei ist zu beachten, dass die Datenlisten von allen Links gleich lang sein müssen, weswegen eine Interpolation

nötig ist. Zum Erstellen der Heatmaps habe ich die Methode heatmap von seaborn [15] verwendet

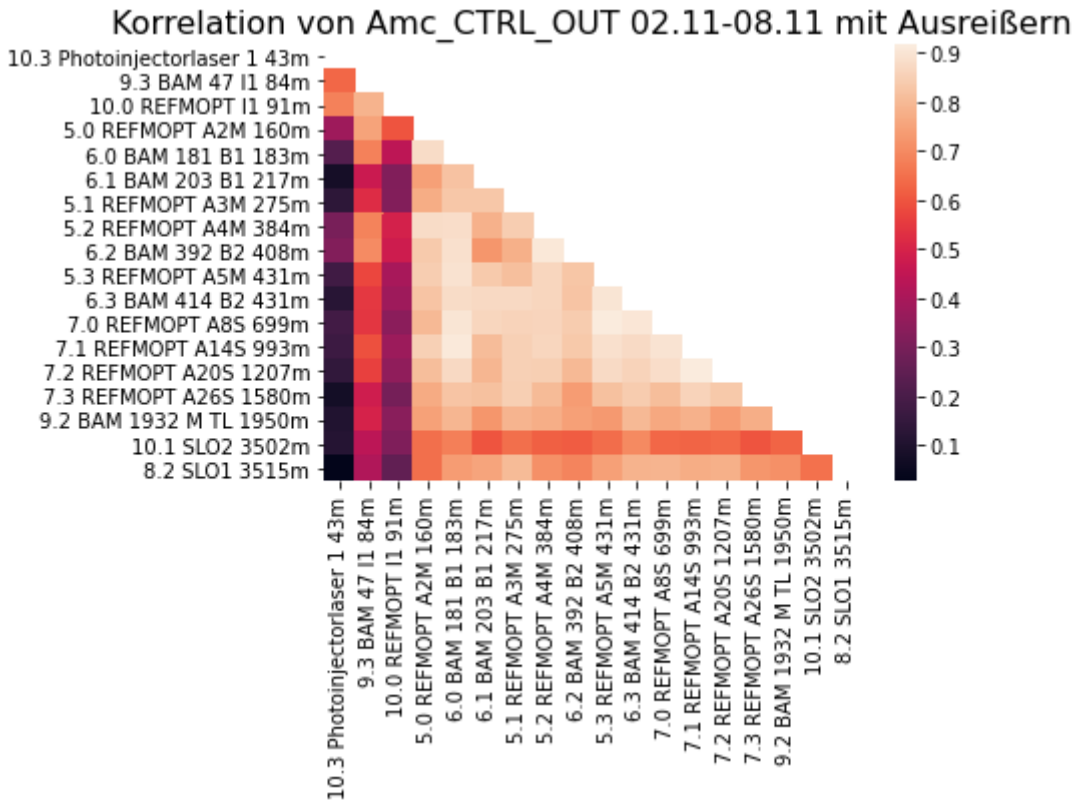

Abbildung 9 Heatmap vom 02.11 bis 08.11 für die CTRL OUT Kanälen

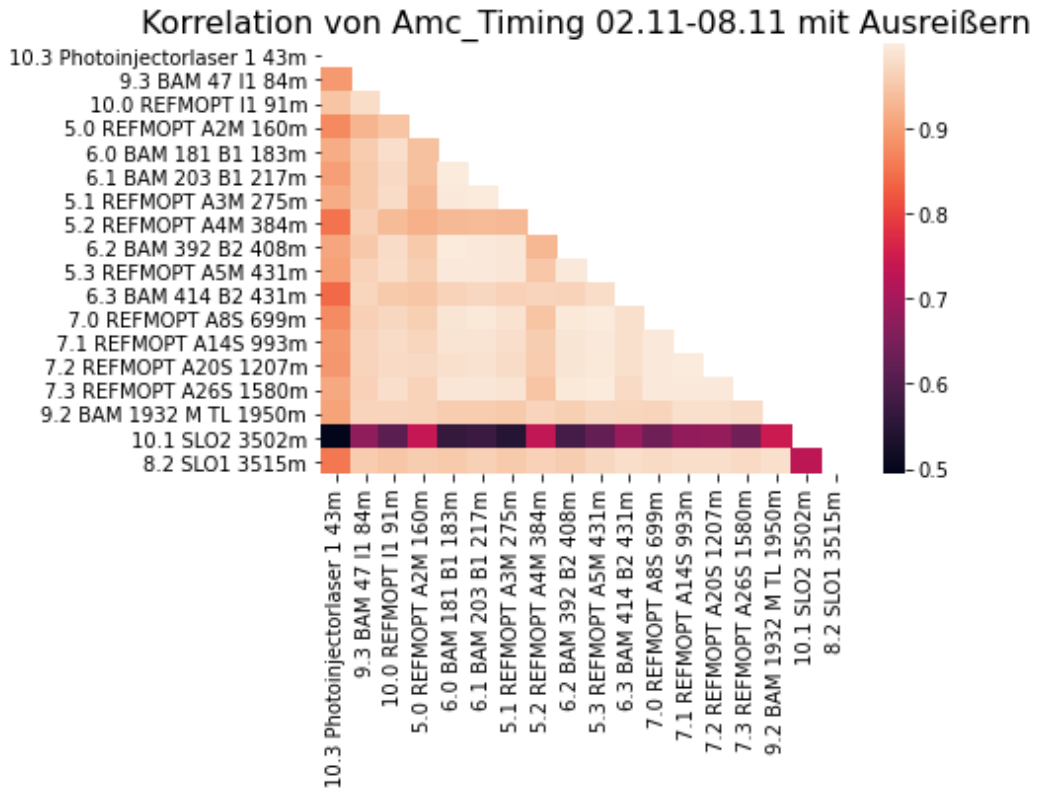

Abbildung 10 Heatmap vom 02.11 bis 08.11 für die TIMING Kanälen

Bei den Heatmaps für die Woche vom 02.11 bis zum 08.11 lässt sich direkt ein Bild erkennen, was man besonders bei der Heatmap der TIMING Kanälen sieht. Hier sieht man, dass der Link 10.1 SLO2, welcher eine Länge von 3.502m aufweist und in Schenefeld endet, sehr stark heraussticht. Das Besondere an diesem Link ist, dass er nicht parallel zu den anderen Links verläuft und sich durch den großen Längenunterschied auch von den anderen Links unterscheidet. Hier hat der nächste Link nur eine Länge von 1.950m. Sonst ergibt sich kein auffälliges Bild für die anderen Links, was zu erwarten ist, da sie sich nicht so stark von der Länge unterscheiden und auch parallel verlaufen. Ich habe hier für die TIMING Kanäle und für die CTRLOUT Kanäle jeweils eine Heatmap gebildet, wobei die Heatmap für die TIMING Kanäle aussagekräftiger ist, da hier die Course Tuning Schritte rausgerechnet werden und so die Daten durch abrupte Sprünge nicht verfälscht werden. Die Heatmap für die CTRLOUT Kanäle unterscheidet sich darin, dass hier auch die Links 10.3, 9.3 und 10.0 herausstechen. Dies sind Links, die alle im Injektorgebäude enden. Diese Abweichung lässt sich vermutlich dadurch erklären, dass die Course Tuning Schritte nicht rausgerechnet wurden im Gegensatz zu den TIMING Kanälen

und sich das Ergebnis dadurch verändert. Durch diese Course Tuning Schritte verfälschen die Ergebnisse zum Teil, weswegen ich mich für die späteren Analyseschritte dazu entschieden habe, mich mehr auf die TIMING Kanäle zu fokussieren, da hier die Course Tuning Schritte rausgerechnet werden.

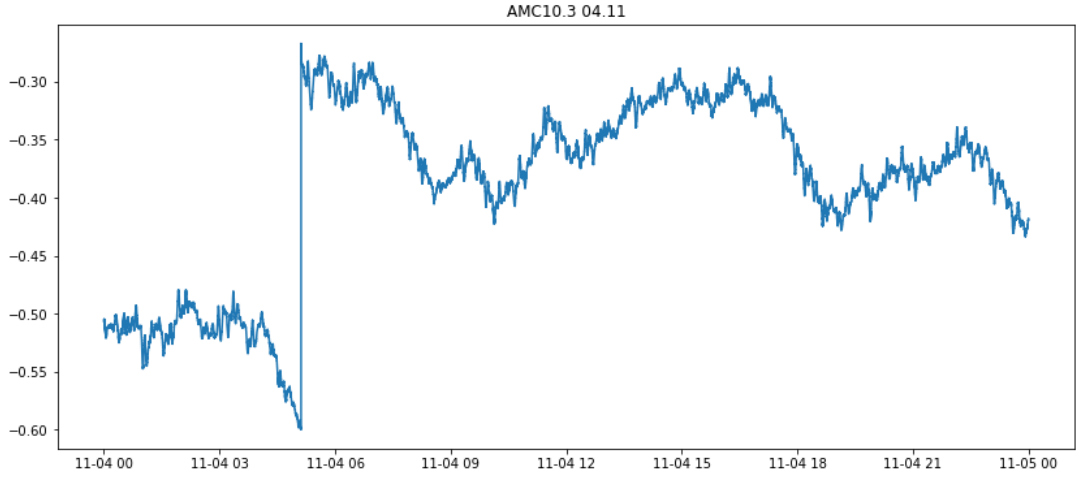

Abbildung 11 Beispiel Course Tuning Schritt in den Rohdaten

Auf dem Screenshot sieht man kurz vor 6 Uhr am 4.11 einen Course Tuning Schritt.. Diese finden statt, wenn ein bestimmter Grenzwert für die Steuerspannung bei den Daten erreicht wird, was sich meisten auf einen Bereich von -0,6 bis +0,6 begrenzt. Als Ausreißer werden hier die Links bezeichnet, die nicht im Tunnel enden, also die Links: 10.3 Photoinjectorlaser 1, 9.3 BAM 47 I1, 10.0 REFMOPT I1, 10.1 SLO2 und 8.2 SLO1. Zusätzlich habe ich die jeweiligen Korrelationen als einzelne Graphen dargestellt. Hier war es hilfreich anstatt der vorher benutzten correlate-Funktion von numpy [14], die Methode correlate von scipy [16] zu benutzen. Bei dieser Methode kann ich angeben, auf welche Art die Korrelationen berechnet werden. Hier habe ich mich, durch Tipps von DESY, für die „fft“-Variante entschieden. Dies bietet eine sehr viel performantere Lösung, um die Korrelationen zu berechnen, was eine erhebliche Zeitersparnis mit sich bringt. Die Graphen für die jeweiligen Korrelationen sind im Notebook zu finden.

Der nächste Schritt innerhalb meiner Experimente war es, den Zeitraum zu erweitern, um so Auffälligkeiten erkennen zu können. Also habe ich zuerst damit beschäftigt, die Daten für den

Zeitraum von dem 09.11.2020 bis zum einschließlich 22.11.2020 auszulesen und zu speichern. Bei der Betrachtung der Rohdaten für die CTRL OUT Kanäle ist mir aufgefallen, dass in der Woche vom 09.11 bis zum 16.11 die oben angesprochenen Oszillationen immer noch vorhanden waren und dann in der darauffolgenden Woche abgeklungen sind. Ebenso habe ich für diese beiden Wochen für die CTRL OUT Kanäle und für die TIMING Kanäle Heatmaps gebildet und die einzelnen Korrelationsgraphen erstellt. Zu den Heatmaps habe ich noch die Unterscheidung gemacht, dass ich einmal eine Heatmap mit allen Links gebildet habe und eine, in der ich nur die Links berücksichtigt habe, die in dem Tunnel enden. Somit werden eventuell Ergebnisse für die Links in dem Tunnel nicht durch die anderen Links verfälscht, sodass sie in der Betrachtung nicht auffallen. Im Anhang findet man noch die zusätzlichen Heatmaps, in denen ich die Ausreißer für die Berechnung der Heatmap rausgenommen habe. Diese Heatmaps sind mit dem Zusatz „ohne Ausreißer“ im Titel zu finden.

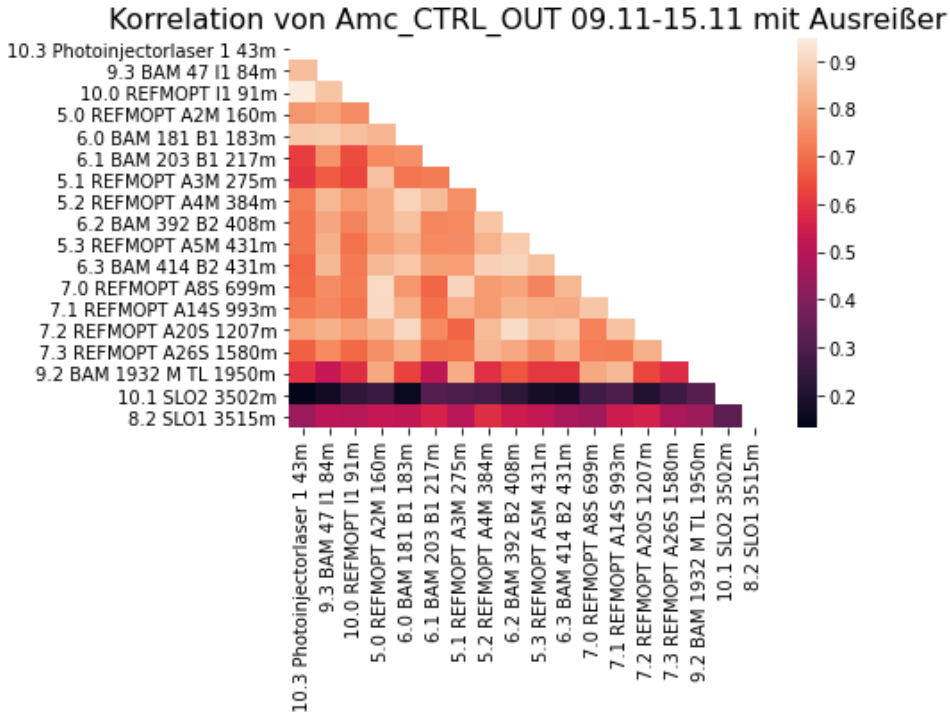

Abbildung 12 Heatmap vom 09.11 bis 15.11 für die CTRL OUT Kanälen

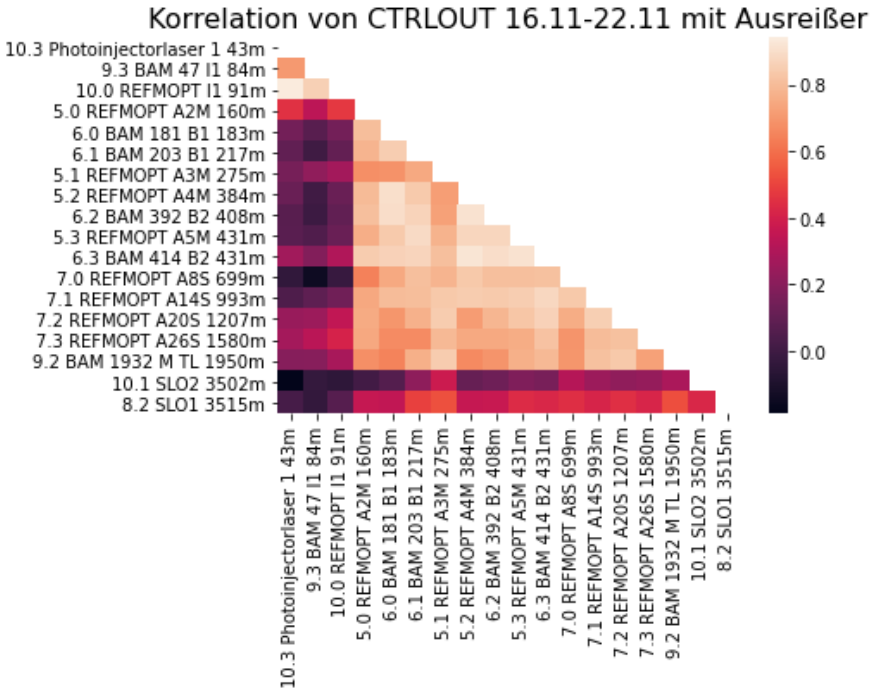

Abbildung 13 Heatmap vom 16.11 bis 22.11 für die CTRL OUT Kanälen

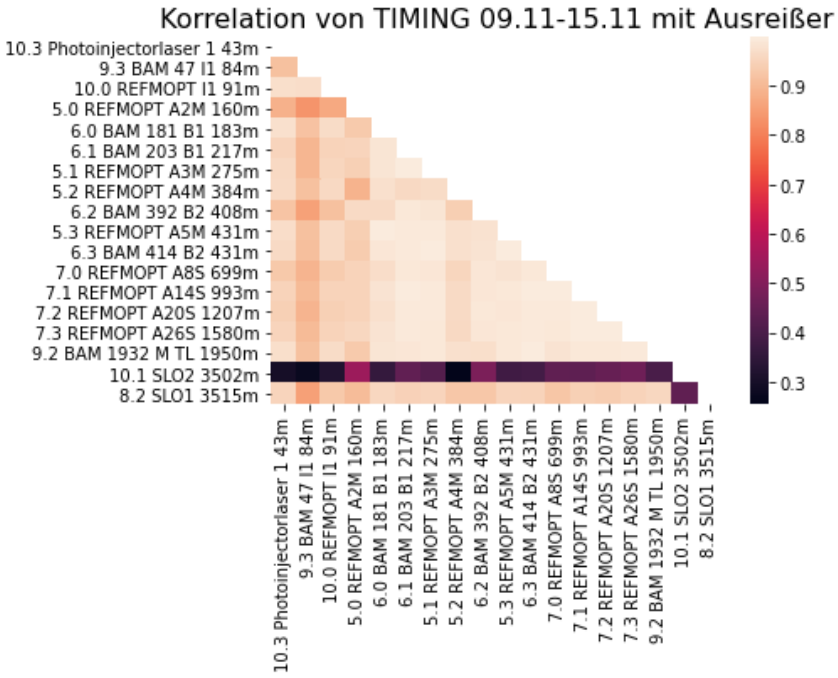

Abbildung 14 Heatmap vom 09.11 bis 15.11 für die TIMING Kanälen

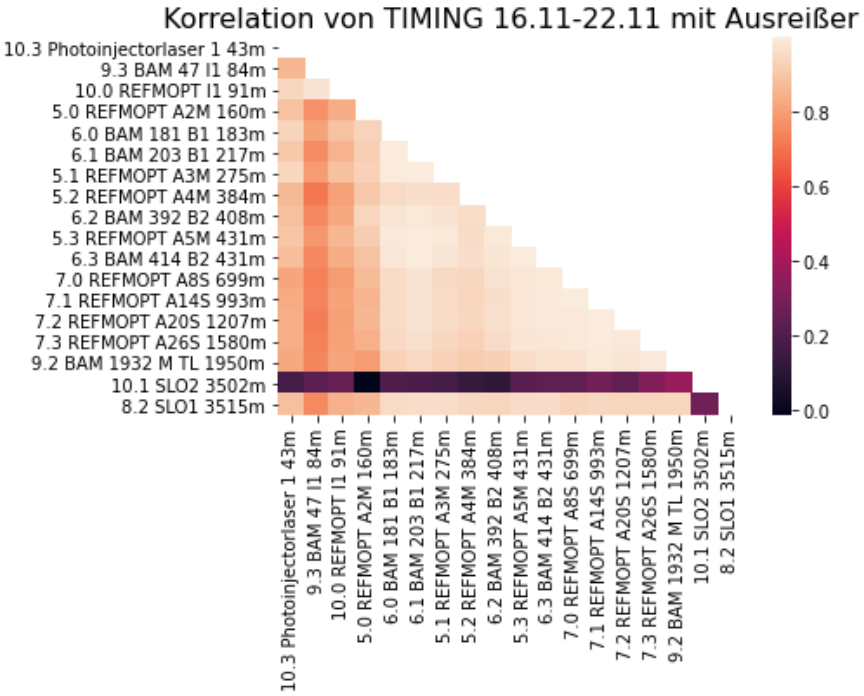

Abbildung 15 Heatmap vom 16.11 bis 22.11 für TIMING Kanälen

Bei den Heatmaps für die Woche vom 09.11 bis zum 15.11 und für die Woche vom 16.11 bis zum 22.11 lässt sich wieder ein ähnliches Bild erkennen wie in der Woche davor, was zu erwarten war. Auch hier sieht man, dass der 10.1 Link bei den TIMING Kanälen in der Heatmap wieder rausfällt. Auch lässt sich hier bei den TIMING Kanälen erkennen, dass die drei Links, die im Injektorgebäude enden, leicht herausstechen. Dies ist bei der Woche vom 16.11 bis zum 22.11 besser ersichtlich. Was sich hier aber ändert ist, dass in der Woche vom 09.11 bis zum 15.11 bei den CTRLOUT Kanälen keine Links, bis auf den 10.1 Link herausstechen, aber bei der Woche vom 16.11 bis zum 22.11 bildet sich wieder ein ähnliches Bild ab wie in der Vorwoche vom 02.11 bis zum 08.11. Hier stechen auch die Links raus, die im Injektorgebäude enden, sowie die beiden langen Links, welche bis Schenefeld laufen. Bei der Bearbeitung von diesen Daten, kamen wir in der gemeinsamen Besprechung der Ergebnisse, auf die Idee, die Daten vor der Korrelation zu skalieren. Endgültig haben wir uns dafür entschieden die Daten über den MinMaxScaler von Sklearn [17] zu skalieren. Hierbei ging es darum, die Daten über alle Features der Matrix zu skalieren, aber dadurch das der MinMaxScaler die Daten für jedes

Features individuell skaliert, was nach der Besprechung im Meeting nicht gewünscht war, musste ich meine Matrix bevor ich sie skaliere in ein eindimensionales Array umgewandelt, damit der Scaler die Daten über alle Features skaliert und dann konnte ich die Daten wieder in die ursprüngliche Form zurückbringen, um weiter mit den Daten zu arbeiten. Dabei ist mir bei den Korrelationen aufgefallen, dass meine Plots wieder eine pyramidenartige Form haben. Dies habe ich dadurch behoben, dass ich nach der MinMaxskalierung wieder eine Mittelwertsbereinigung durchgeführt habe. Zusätzlich zu den Korrelationen der einzelnen Wochen untereinander, habe ich noch eine Korrelation über die drei gesamten Wochen durchgeführt.

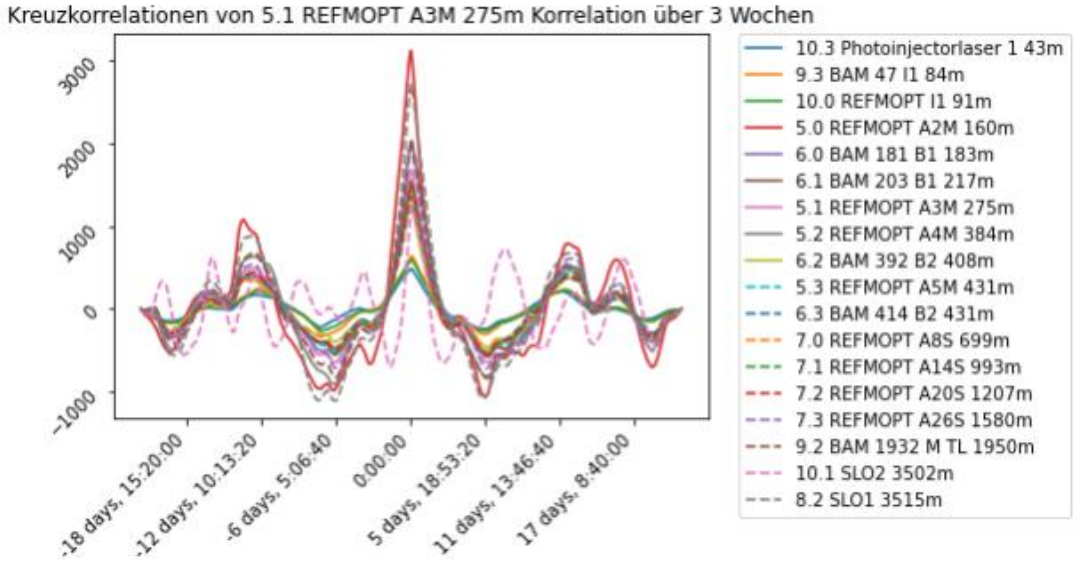

Abbildung 16 Kreuzkorrelationen über 3 Wochen für die TIMING Kanäle

Der Graph beschreibt hier die Korrelationen des 5.1 mit allen anderen Links einschließlich sich selbst. Hier zeigt sich, dass der 5.0 Link ein starkes Korrelationsverhalten aufweist und ziemlich heraussticht. Ebenso sticht der 10.1 Link heraus, was keine Überraschung ist, da er schon in den Heatmaps ein ziemlich auffälliges Verhalten gezeigt hat und dies nur durch diese Korrelationen weiter gestützt wird. Dieses Verhalten der beiden Links zeigt sich so auch in den weiteren Analysen. Ebenso interessant ist, dass die drei Links, die im Injektorgebäude enden (10.3, 9.3 und 10.0) ein schwaches Korrelationsverhalten aufweisen. Dieses schwache Korrelationsverhalten lässt sich darauf zurückführen, dass die Links kurz sind und nicht bis in den

Tunnel gehen, weswegen sie weniger anfällig für Störungen und Klimabedingungen sind. Hier sticht auch heraus, dass alle Links bis auf den 10.1 drei Maxima über den kompletten Korrelationsgraphen aufweisen und sich dadurch keine Periode über die drei Wochen erkennen lässt. Nur der 10.1 weist mehr Extrema auf und lässt eine Periode erkennen. Die Ursache für diese Periode konnte aber in der gemeinsamen Besprechung nicht gefunden werden. Hier wäre es interessant die Korrelation über einen größeren Zeitraum zu erstrecken, was aber im Rahmen der Bachelorarbeit zu zeitaufwändig wäre, da das Daten auslesen einen sehr großen Teil der Zeit in Anspruch genommen hat.

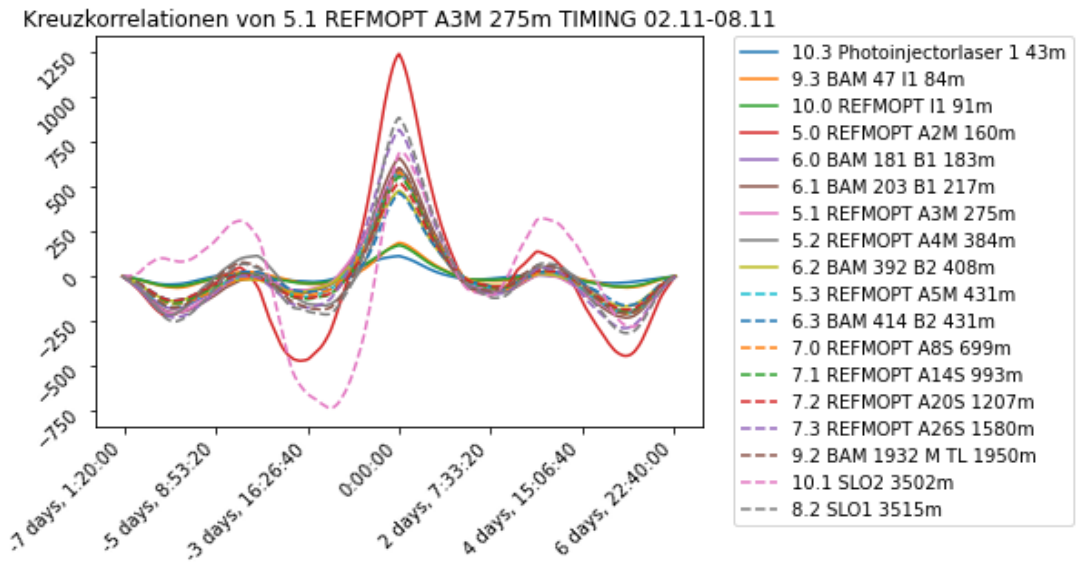

Abbildung 17 Korrelationen für den Link 5.1 vom 02.11 - 08.11

Hier sieht man im Vergleich die Kreuzkorrelationen für eine Woche, in diesem Fall die Woche vom 02.11 bis zum 08.11. Bei diesem Screenshot zeigt sich ebenso ein auffälliges Verhalten der beiden Links 10.1 und 5.0. Nur weisen hier alle Links die gleiche Anzahl an Extrema auf, wobei sie in unterschiedlicher Ausprägung auftreten. Auch hier sieht man das schwache Korrelationsverhalten der ersten drei Links.

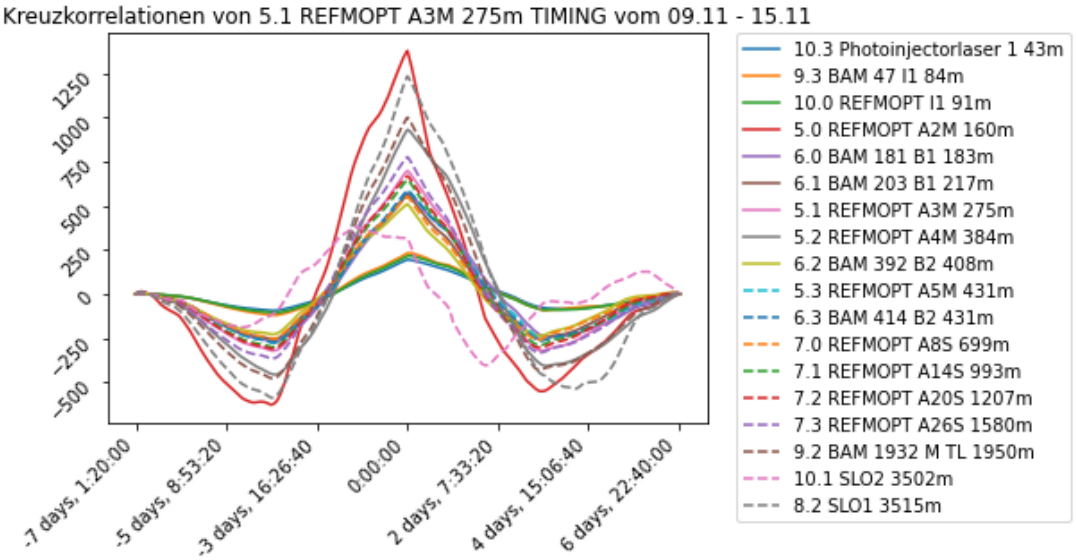

Abbildung 18 Korrelationen für den Link 5.1 vom 09.11 - 15.11

Für die zweite Woche der Kreuzkorrelation für den 5.1 Link ergibt sich ein anderes Bild. Die Tendenzen, dass sich die ersten drei Links gleich verhalten, sowie dass der 10.1 Link in seinem Verhalten abweicht und der 5.0 Link ein starkes Korrelationsverhalten zeigt, bleiben gleich.

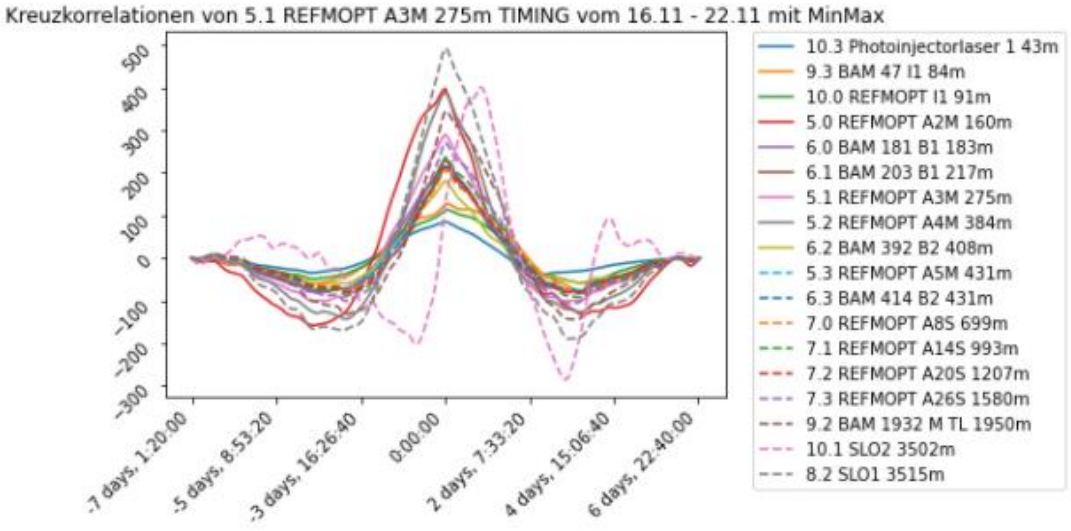

Abbildung 19 Korrelationen für den Link 5.1 vom 16.11 - 22.11

Für die dritte Woche habe ich vor der Korrelation die Daten noch Minmax skaliert. Dies hatte aber keine Auswirkungen auf die Korrelationen, weswegen ich bei den anderen beiden Wochen bei den alten Korrelationsgraphen geblieben bin ohne vorheriger Minmaxskalierung, wobei die Mittelwertsbereinigung bei allen Korrelationen vorher stattgefunden hat. Für diese Woche ergibt sich ebenso ein anderes Bild, wobei auch hier die Tendenzen gleich sind. Zusätzlich dazu zeigt hier diesmal auch der ein starkes Korrelationsverhalten, wobei er auch bei den ersten beiden Wochen ein starkes Korrelationsverhalten gezeigt hat, aber dort der 5.0 Link stärker herausgestochen ist. Für die Veranschaulichung der einzelnen Wochen und der Korrelation über die drei Wochen habe ich mich für den Link 5.1 entschieden, wobei es keine großen Auswirkungen hat, für welchen Link ich mich zur Veranschaulichung entscheide, weil sie alle ein sehr ähnliches Bild zeigen. Die Kreuzkorrelationen von den anderen Links sind auch im beigefügten Jupyter Notebook zu finden.

Als Nächstes ging es darum die Klimadaten mit in meine Analyse einzubeziehen. Das Problem was sich bei den Klimadaten herausgestellt hat war, dass die meisten Sensoren zu gering aufgelöst waren und ich bei manchen Graphen über eine Woche nur drei verschiedene Werte hatte. Klimadaten, die sich angeboten hatten, um mit ihnen weiterzuarbeiten, waren die Luftfeuchtigkeitsdaten von den verschiedenen Sensoren in den verschiedenen Tunnelsegmenten.

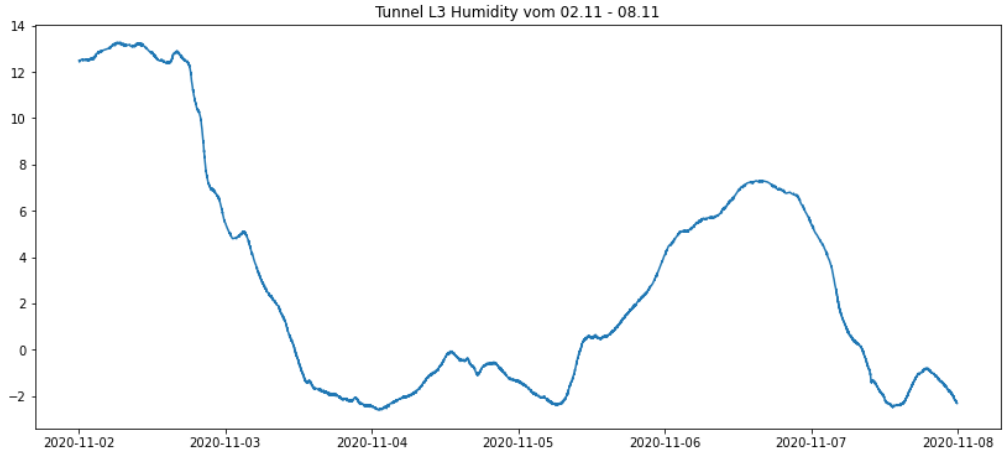

Abbildung 20 Hoch aufgelöste Klimadaten

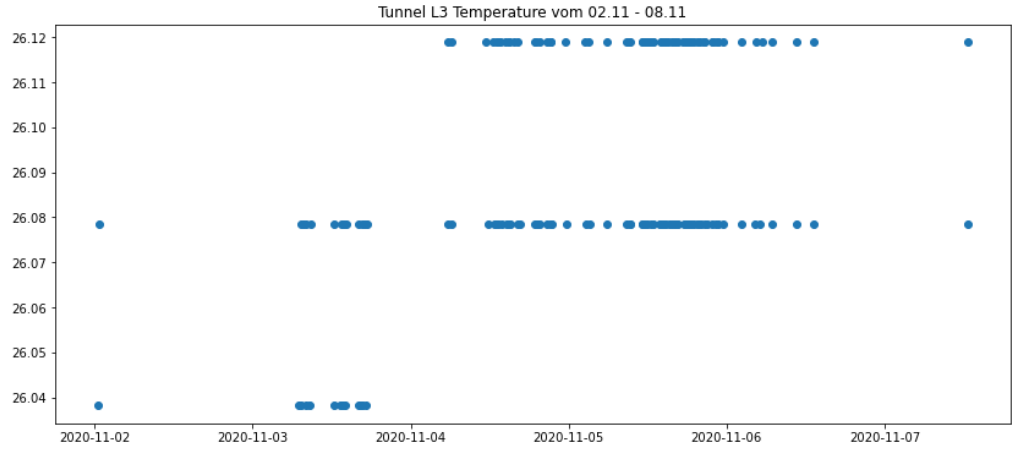

Abbildung 21 Niedrig aufgelöste Klimadaten

An den beiden Fotos erkennt man den Unterschied in der Auflösung der Daten. Im ersten Bild sieht man einem normalen Datenverlauf, wobei man bei den Temperaturen im zweiten Bild nur drei verschiedene Datenpunkte erkennt. Bei den Daten für die Feuchtigkeit muss man beachten, dass ich für die Berechnung der Korrelationen eine Mittelwertsbereinigung durchgeführt habe, weswegen die Werte zum Teil negativ sind. Hier hat sich ein Scatterplot für die Darstellung geeignet. Das liegt daran, dass die Temperatursensoren nur in 0,04er Schritten ihre Werte ändern, sodass man auf nur wenige verschiedene Werte kommt, wenn die Temperatur konstant ist. Auf dem Bild ist noch zu erkennen, dass der Sensor keine Daten sendet, wenn die Daten sich nicht verändern, das kann man besonders direkt am Anfang am 02.11 sehen, an dem sich die Daten über einen Tag lang nicht genug geändert haben. Es werden nur neue Werte gesendet, wenn sich der Datenwert auch geändert hat und über diesem Zeitraum hat sich wie schon beschrieben die Temperatur konstant gehalten. Der nächste Schritt war es, die Klimadaten mit den Daten von den TIMING Kanälen und den CTRL OUT Kanälen zu korrelieren. Die Korrelationen der Links mit den anderen aussagekräftigen Klimadaten und die dazugehörigen Rohdaten sind im Anhang zu finden. Dabei ist zu beachten, dass L1, L2, L3 und SASE1 für verschiedene Sensoren im Tunnel stehen.

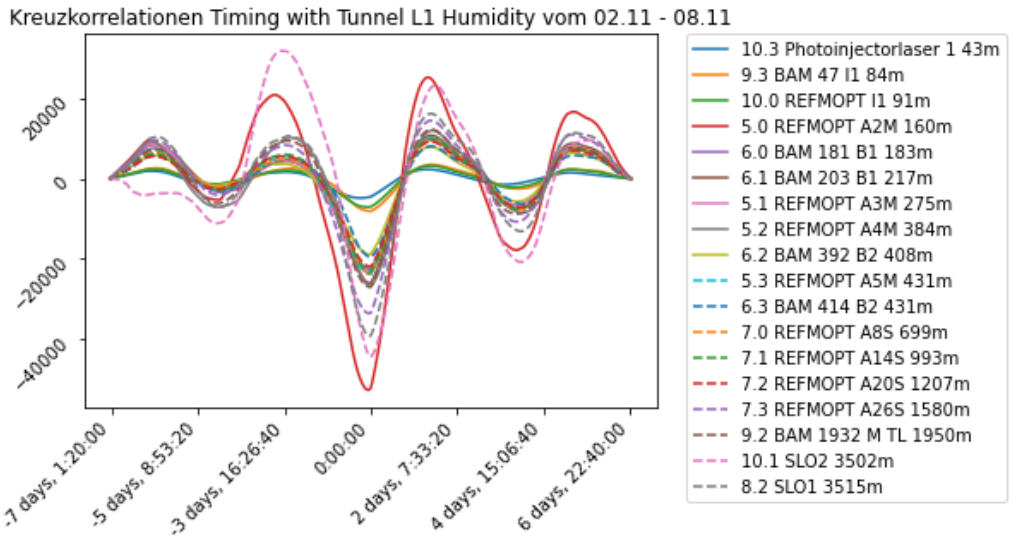

Abbildung 22 Kreuzkorrelation vom 02.11 - 08.11 für die Klimadaten

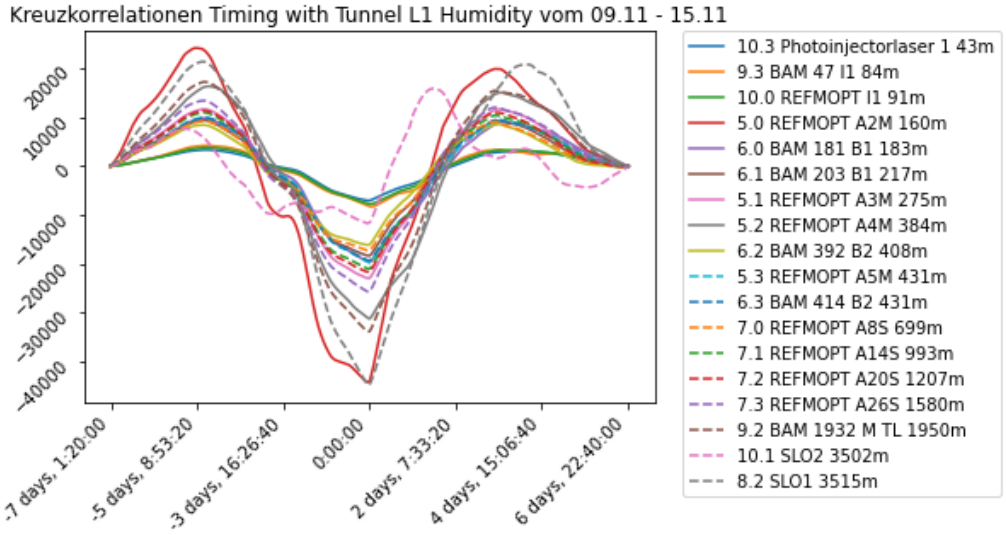

Abbildung 23 Kreuzkorrelation vom 09.11 - 15.11 für die Klimadaten

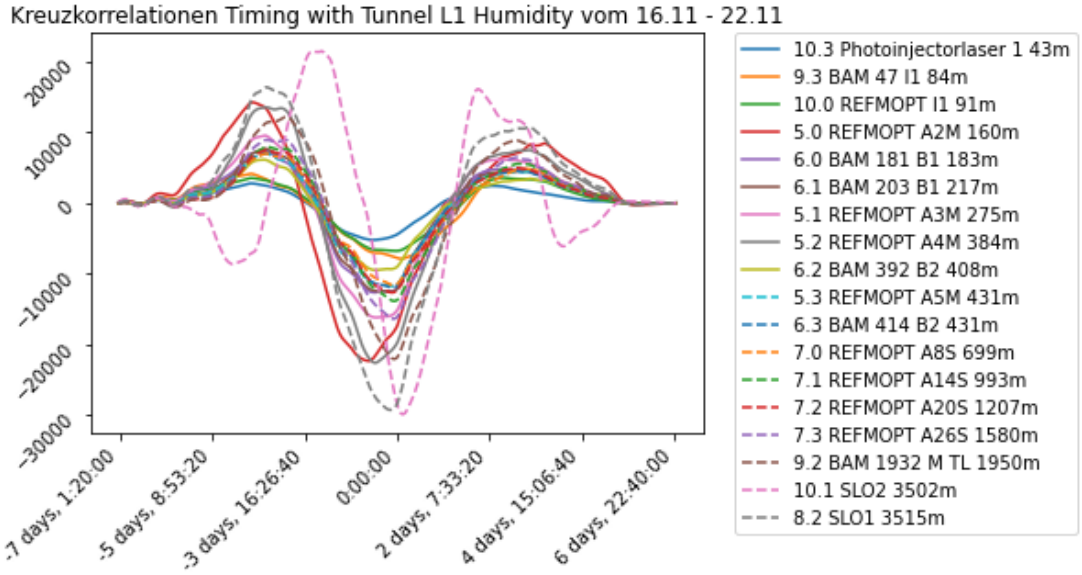

Abbildung 24 Kreuzkorrelation vom 16.11 - 22.11 für die Klimadaten

Für diese Klimadaten habe ich mich entschieden nur noch die Korrelationen mit den TIMING Kanälen durchzuführen, weil hier die Course Tuning Schritte rausgerechnet sind und so das Korrelationsverhalten nicht verfälschen können. Auf den Screenshots sieht man die Kreuzkorrelation der TIMING Kanäle mit den verschiedenen Wochen und mit dem Sensor in dem ersten Tunnelsegment. In dem Tunnel befinden sich vier Sensoren, welche sich über den Verlauf des Tunnels verteilen. Die Korrelationen mit den Daten von den Sensoren aus den anderen Tunnelsegmenten sehen im Verlauf ähnlich aus (siehe Anhang). An den Korrelationen mit den Klimadaten sieht man direkt zwei Links, die herausstechen. Wie schon in den oben abgebildeten Heatmaps sieht man, dass der 10.1 Link herausfällt. Dieser Link ist bisher in allen Korrelationsabbildungen herausgestochen. Ein Grund hierfür ist, dass er im Vergleich zu den anderen Links sehr lang ist und zum Teil einen anderen Weg als der 8.2 Link nimmt. Der Grund warum sich nur der 10.1 Link anders verhält im Korrelationsverhalten wurde auch in der gemeinsamen Besprechung nicht gefunden. Die Links 8.2 und 10.1 haben ab einen bestimmten Punkt (Osdorfer Born) einen unterschiedlichen Weg, aber dennoch weist der 8.2 Link ein ähnliches Korrelationsverhalten auf, wie die restlichen Links im Tunnel. Ebenfalls interessant ist,

dass der 5.0 Link (rote durchgezogene Linie) auch sehr stark in seinem Korrelationsverhalten heraussticht. Bei der Betrachtung der Rohdaten konnte ich feststellen, dass der 5.0 Link sehr große Veränderungen im Verlauf der Zeit macht, was vermutlich auch das starke Korrelationsverhalten erklärt. Was sich noch ergibt ist, dass die Links, die schon im Injektorgebäude enden (10.3, 9.3 und 10.0) ein eher schwachen Korrelationsverhalten aufweisen, was sich darauf zurückführen lässt, dass sie von den Klimaänderungen im Tunnel nicht betroffen sind, da sie schon vorher enden.

Als letzter Schritt in der Analyse ging es darum die Daten zu Clustern. Hierbei habe ich mich, mit Rücksprache aus den wöchentlichen Meetings, dazu entschieden den K-Means-Algorithmus [18] mit zwei, drei und vier Clustern anzuwenden. Zu erwarten war, dass sich drei Cluster bilden, nämlich einmal für die Links, die im Injektorgebäude enden, für die Links, die im Tunnel enden und für die beiden letzten Links, die bis nach Schenefeld verlaufen. Genutzt wurden die Daten der TIMING Kanäle für die Featureextraction. Als Input für den k-Means-Algorithmus habe ich eine Featureextraction [19] auf meine Daten mit den Features:

- Mean: Summe der Werte geteilt durch ihre Anzahl
- Maximum: Höchster Wert in den Daten
- Minimum: Niedrigster Wert in den Daten
- fft aggregated mit den aggtypes: centroid und variance
- Median: Der Wert bei dem die Hälfte der Werte oberhalb und die andere Hälfte unterhalb des Wertes liegt
- mean_second_derivative_central: Der Mean der Annäherung der zweiten Ableitung
- abs_energy: Die Summe über die quadrierten Werte
- mean_abs_change: Der Mean der Summe über die absoluten Differenzen zwischen aneinander liegenden Werten
- mean_change: Der Mean der Summe über die Differenzen zwischen den aneinander liegenden Werten
- fourier_entropy mit bins = 10

Die Features wurden durch die gemeinsamen wöchentlichen Meetings ausgewählt. Die Cluster sind nummeriert, so gibt es hier bis zu vier Cluster bei den Experimenten (in den nachfolgenden Tabellen mit 0, 1, 2 und 3 benannt).

Ergebnisse des Clustering (K-Means mit zwei Clustern):

Links	***Woche 1***	***Woche 2***	***Woche 3***
10.3 Photoinjectorlaser 1 (43m)	0	1	0
9.3 BAM 47 I1 *(84m)*	0	1	0
10.0 REFMOPT I1 (91m)	0	1	0
5.0 REFMOPT A2M (160m)	0	1	0
6.0 BAM 181 B1 (183m)	0	0	0
6.1 BAM 203 B1 (217m)	0	0	0
5.1 REFMOPT A3M (275m)	0	1	0
5.2 REFMOPT A4M (384m)	0	1	1
6.2 BAM 392 B2 (408m)	0	0	0
5.3 REFMOPT A5M (431m)	0	0	0
6.3 BAM 414 B2 (431m)	0	0	0
7.0 REFMOPT A8S (699m)	0	0	0
7.1 REFMOPT A14S (993m)	0	0	0
7.2 REFMOPT A20S (1207m)	0	0	0
7.3 REFMOPT A26S (1580m)	0	1	0
9.2 BAM 1932 M TL (1950m)	0	1	0
10.1 SLO2 (3502m)	1	0	1
8.2 SLO1 (3515m)	0	1	0

Tabelle 1 Ergebnisse des K-Means für zwei Cluster

Ich habe die Ergebnisse zur besseren Lesbarkeit in einer Tabelle zusammengefasst. Hier werden die verschiedenen Cluster durch 0 und 1 beschrieben, dadurch dass ich den K-Means mit zwei Clustern laufen gelassen habe, ergeben sich auch zwei verschiedene Cluster für das Ergebnis. Die Wochen 1 bis 3 beschreiben hier die weiter oben beschriebenen Daten und umfassen insgesamt den Zeitraum vom 02.11 bis zum 22.11. Die Tabelle bietet einen Überblick über den Verlauf wie sich die zugeordneten Cluster gegebenenfalls verändert haben. Neben dem K-Means mit den zwei Clustern haben wir uns in der gemeinsamen Besprechung dazu

entschieden, dass ich den K-Means mit zwei, drei und vier Clustern durchlaufen lasse. Bei dem Durchlauf des K-Means mit den zwei Clustern kann man erkennen, dass in der ersten Woche alle Links bis auf der 10.1 dem gleichen Cluster zugeordnet werden. Dies unterstützt die bisherigen Erkenntnisse bezüglich des Links, da der Link ja schon in den Heatmaps und Korrelationen mit den Klimadaten herausgestochen ist mit seinem Verhalten. In der zweiten Woche sieht man ein eher ungewöhnliches Verhalten, da diesmal mehrere Links dem gleichen Cluster wie der 10.1 Link zugeordnet werden. Selbst die drei Links, die im Injektorgebäude enden, wurden dem gleichen Cluster zugeordnet. Dies ist ein Ergebnis, welches mich sehr überrascht hat, da der 10.1 Link ca. 3.400m länger ist als die Links, die im Injektorgebäude enden. Meine Erwartung war am Anfang, dass vermutlich wieder nur der 10.1 Link einem einzigen Cluster zugeordnet wird. Die Ursache für dieses Verhalten wurde auch nicht in der gemeinsamen Besprechung gefunden. Für die dritte Woche hat sich wieder ein ähnliches Bild gezeigt, da hier auch wieder der 10.1 Link einem Cluster zugeordnet wird. Diesmal aber zusätzlich mit dem 5.2 Link. Zu erwarten war wieder nur, dass der 10.1 Link allein einem Cluster zugeordnet wird, weil er in den Heatmaps schon stark herausgestochen ist. Nachfolgend habe ich noch den K-Means mit drei und vier Clustern durchgeführt, wobei hier der K-Means mit drei Clustern am interessantesten ist, da von drei Clustern ausgegangen wird für die Gesamtmenge der Links.

Ergebnisse des Clustering (K-Means mit drei Clustern):

Links	***Woche 1***	***Woche 2***	***Woche 3***
10.3 Photoinjectorlaser 1 (43m)	0	1	0
9.3 BAM 47 I1 *(84m)*	0	1	0
10.0 REFMOPT I1 (91m)	0	1	0
5.0 REFMOPT A2M (160m)	2	1	2
6.0 BAM 181 B1 (183m)	0	2	2
6.1 BAM 203 B1 (217m)	0	0	2
5.1 REFMOPT A3M (275m)	0	2	2
5.2 REFMOPT A4M (384m)	0	1	1

6.2 BAM 392 B2 (408m)	0	0	2
5.3 REFMOPT A5M (431m)	0	2	2
6.3 BAM 414 B2 (431m)	0	0	0
7.0 REFMOPT A8S (699m)	0	0	0
7.1 REFMOPT A14S (993m)	0	0	0
7.2 REFMOPT A20S (1207m)	0	2	0
7.3 REFMOPT A26S (1580m)	0	2	2
9.2 BAM 1932 M TL (1950m)	0	2	0
10.1 SLO2 (3502m)	1	0	1
8.2 SLO1 (3515m)	2	1	0

Tabelle 2 Ergebnisse des K-Means für drei Cluster

Für die erste Woche zeigt sich hier ein Bild, was in dieser Art auch erwartet wurde. Hier wird der 10.1 Link einem Cluster allein zugeordnet, sowie die Links 8.2 und 5.0 werden einem Cluster zugeordnet und die restlichen Links werden dem letzten Cluster zugeordnet. Dass der 10.1 Link einem einzelnen Cluster zugeordnet wird, war durch sein bisheriges Verhalten zu erwarten. Besonders der 5.0 Link ist in den Korrelationen mit den Klimadaten auch herausgestochen durch sein Verhalten, was seine Zuordnung in ein anderes Cluster mit dem 8.2 Link erklärt. Ohne die vorherigen Analysen mit den Korrelationen würde man davon ausgehen, dass die ersten drei Links einem Cluster zugeordnet werden, sowie die letzten beiden langen Links einem Cluster zugeordnet werden und dass die restlichen Links dem letzten Cluster zugeordnet werden. Für die zweite Woche wurden die Erwartungen für die Cluster nicht getroffen, da diesmal die Cluster ziemlich gemischt aussehen. Es bildet sich kein klares Bild, aber es sind Tendenzen zu erkennen, wie dass die ersten drei Links dem gleichen Cluster zugeordnet wurden, was sich bisher so in den Experimenten gezeigt hat. Zusätzlich unterscheiden sich auch die letzten beiden Links in den Clustern, was bei den bisherigen Ergebnissen auch immer der Fall war. Die letzte Woche zeigt auch wieder ein gemischtes Ergebnis mit den gleichen Tendenzen wie in der Vorwoche.

Ergebnisse des Clustering (K-Means mit vier Clustern):

Links	***Woche 1***	***Woche 2***	***Woche 3***
10.3 Photoinjectorlaser 1 (43m)	3	1	0
9.3 BAM 47 I1 *(84m)*	3	1	0
10.0 REFMOPT I1 (91m)	3	1	0
5.0 REFMOPT A2M (160m)	0	1	2
6.0 BAM 181 B1 (183m)	3	0	2
6.1 BAM 203 B1 (217m)	3	2	2
5.1 REFMOPT A3M (275m)	3	0	2
5.2 REFMOPT A4M (384m)	1	1	3
6.2 BAM 392 B2 (408m)	1	2	2
5.3 REFMOPT A5M (431m)	3	0	2
6.3 BAM 414 B2 (431m)	1	2	0
7.0 REFMOPT A8S (699m)	1	2	0
7.1 REFMOPT A14S (993m)	1	2	0
7.2 REFMOPT A20S (1207m)	1	2	0
7.3 REFMOPT A26S (1580m)	1	0	2
9.2 BAM 1932 M TL (1950m)	1	0	0
10.1 SLO2 (3502m)	2	3	1
8.2 SLO1 (3515m)	0	1	0

Tabelle 3 Ergebnisse des K-Means für vier Cluster

In der ersten Woche sind die Ergebnisse in einem großen Teil wieder wie erwartet ausgefallen. Hier wird der 10.1 Link wieder einem einzelnen Cluster zugeordnet. Zusätzlich dazu wird der andere lange Link 8.2 gemeinsam mit dem 5.0 Link einem Cluster zugeordnet. Der 5.0 Link ist durch sein Korrelationsverhalten bei den Klimadaten schon herausgestochen, wodurch der 5.0 Link dann auch nicht zu den restlichen Links zugeordnet wurde. Die restlichen Links verteilen sich auf die beiden übrigen Cluster, wobei hier auch wieder die drei Links, die im Injektorgebäude enden, wieder dem gleichen Cluster zugeordnet werden. Für die zweite Woche

zeigt sich wieder ein gemischtes Ergebnis. Es zeigen sich wieder Tendenzen, wie dass der Link 10.1 einem eigenen Cluster zugeordnet wird und sich von dem anderen langen Link 8.2 unterscheidet. Ebenso ist es der Fall, dass die ersten drei Links wieder dem gleichen Cluster zugeordnet werden, aber diesmal zusätzlich mit dem 5.0 Link, der sonst immer stark herausgestochen ist in seinem Verhalten bei der Korrelation mit den Klimadaten. Das sonstige Clusteringverhalten weist keine erwarteten Muster auf. Für die dritte Woche weisen sich wieder ähnliche Tendenzen auf, wie bei der zweiten Woche. Hier wird auch dem 10.1 Link ein eigenes Cluster zugeordnet und er unterscheidet sich wieder von dem anderen langen Link 8.2. Ebenso werden wieder die ersten drei Links dem gleichen Cluster zugeordnet.

3.2 Art der Daten

Die Daten sind so aufgebaut, dass sie für jeden Zeitschritt in einer Liste gespeichert werden. Diese Liste besteht wiederrum wieder aus Listen. Jeder Eintrag dieser Listen wird einem Subchannel zugeordnet. Diese Listen bestehen aus Dictionaries, welche mit ihren Keys auf verschiedene Werte verweisen.


```
{'data': array([[0.       , 0.0009613]], dtype=float32), 'type': 'A_TS_GSPECTRU
M', 'timestamp': 1601881799.981427, 'macropulse': 871205286, 'miscellaneous':
{'index': 0, 'comment': 'XFEL.SYNC/LINK.LOCK.DAQ/XTIN.AMC5.CONTROLLER.LSU.2.CTR
L.MON', 'timestamp': 1601881799, 'status': 871205286, 'daqname': 'XFEL.SYNC/LIN
K.LOCK.DAQ/XTIN.AMC5.CONTROLLER.LSU.2.CTRL.MON', 'start': 0.0, 'incr': 1.0, 'gr
oups': 1, 'groupsize': 1, 'groupincr': 0.0, 'stat_mean': 0.0}}
```


Abbildung 25 Output der getData Methode

Im oberen Screenshot sieht man einen Ausschnitt von dem Output der Methode, um die Daten auszulesen. Wichtig in diesem Dictionary waren für mich nur der Datenwert aus dem zweidimensionalen Vektor und der Timestamp, welcher hier noch als Linuxtimestamp vorliegt.

Interessant waren hier wiederrum für die Analyse nur die Timestamps und die dazugehörigen Datenwerte der Subchannel. Der wichtigste Subchannel war hier wiederrum der, in dem die Meandaten liegen, da die Daten, wie oben beschrieben, in einer höheren Frequenz vorkommen und der Mean daraus berechnet wird, damit man Datenpunkte in einer Frequenz von 10 Hz, also alle 100 Millisekunden, vorliegen hat. So ergeben sich für einen Tag ungefähr 860.000 Events. Zur einfacheren Weiterverarbeitung der Daten habe ich sie in einer Matrix gespeichert.

Diese Matrix ist so aufgebaut, dass auf Index 0 der Macropulse liegt und dann abwechselnd die Timestamps und die dazugehörigen Datenwerte. Die Matrix ist nach aufsteigender Länge der Links sortiert.

3.3 Technische Informationen

Der praktische Teil wurde komplett remote über den eigenen Rechner durchgeführt. Es war eine VPN-Verbindung mit dem DESY-Netzwerk nötig, wofür ich einen eigenen Account erhalten habe, mit dem ich mich verbinden konnte. Zusätzlich war noch eine Verbindung zu dem XFEL nötig. Diese Verbindung wurde durch das Programm PuTTY hergestellt. Die Daten konnte ich durch das PythonDAQClientInterface [10] von DESY auslesen. Der pratkische Teil wurde in der Programmiersprache Python bearbeitet, da sich Python bei Datenanalyse anbietet. Als Entwicklungsumgebung habe ich mich hier für Jupyter Notebook entschieden, da Notebooks eine anschauliche Darstellung bieten. Den praktischen Teil der Arbeit habe ich in zwei verschiedenen Notebooks angefertigt. In dem Notebook data_aquisition liegt nur der Code, den ich benutzt habe, um die Daten auszulesen und sie dann als Backup abzuspeichern. Im Notebook data_analysis sind die Graphen zu finden und die Analysen mit dem Clustering und den Korrelationen. Für die Darstellung der Graphen habe ich mich hauptsächlich für die plot Methode von matplot.pyplot [20] entschieden. Die Graphen für die ganzen Rohdaten sind im Notebook zu finden.

3.3.1 Wichtigste verwendete Funktionen

Neben den ganzen getter-Funktionen und einfachen statistischen Methoden (z.B den Durchschnitt der Daten für eine bestimmte Range), welche ich aber im späteren Verlauf der Arbeit nicht mehr benutzt habe. Im Folgenden gehe ich auf die wichtigsten benutzen Funktionen ein:

- reArangeData() im Notebook data_aquisition:
    - Keine Inputparameter.
    - Returnparameter: Hier eine Matrix mit den Daten, bei der auf dem Index 0 der Macropulse liegt und darauffolgend abwechselnd die Timestamps und die dazugehörigen Daten.

  - reArangeData war die wichtigste Funktion bei dem Speichern der Daten. In diesem Schema hatte ich für den weiteren Verlauf die Daten gespeichert.
- convertTimestamps(timestampList) im Notebook data_analysis:
  - Inputparameter: Liste mit Linuxtimestamps
  - Returnparameter: Liste mit umgewandelten Timestamps
  - Diese Funktion dient dazu, um die x-Achse für die Rohdaten leserlicher zu machen, um die Zeit direkt ablesen zu können
- getTimestampListCorr(resultCor, frequency) im Notebook data_analysis:
  - Inputparameter: Liste mit dem Ergebnis aus der Korrelation, Frequenz der Datenpunkte
  - Returnparameter: vorbereitete Liste mit Timestamps für Korrelationen
  - Diese Funktion dient dazu eine Timestampliste für die x-Achse bei den Korrelationen vorzubereiten. Die Funktion baut die Timestamps von der Mitte aus absteigend (links) und aufsteigend (rechts) auf. Die Frequenz dient hier dazu, die Anzahl der Datenpunkte variabel zu halten. Dies ist wichtig, wenn jedes 100. Element oder jedes 10. oder auch jedes Element genutzt werden soll, da hier die Zeitschritte variieren.
- getCorrelationPlots(corrList, nameList, title) im Notebook data_analysis:
  - Inputparameter: Liste mit den Listen die korreliert werden sollen, Namensliste für die Beschriftung der Legende, Titel für den Graphen
  - Keine Returnparameter
  - Diese Funktion wurde genutzt, um die Korrelationen zwischen den Links zu bilden, sowohl Kreuzkorrelation als auch die Autokorrelation.
- getTimeRange(climateList, startDate, endDate) im Notebook data_analysis:
  - Inputparameter: Liste mit Werten, in meinem Fall Klimadaten, Startdatum, Enddatum
  - Returnparameter: Liste mit Werten die zwischen Startdatum und Enddatum liegt
  - Diese Funktion habe ich gebraucht, da die Klimadaten als kompletter Monat vorlagen und um die Daten wochenweise zu korrelieren, habe ich diese Funktion benutzt

- getCorrelationsPlotsClimate(climateList, timestampList, corrList, nameList, title) im Notebook data_analysis:
    - Inputparameter: Liste mit Klimadaten, Liste mit Timestamp der Daten der Links, Liste mit Daten der Links, Namensliste für Legende und Titel für den Graphen
    - Keine Returnparameter
    - Diese Funktion ist für die Korrelation der Daten der Links mit den Klimadaten zuständig. Bevor ich die Korrelation zwischen den Daten durchführen konnte, musste ich die Länge der Liste mit den Klimadaten per interp-Methode von numpy [21] an die Liste der Daten der Links anpassen

3.3.2 Aufbau der verwendeten Notebooks

- data_aquisition: Das data_aquisition Notebook ist sehr simpel gehalten. In diesem Notebook werden die Daten nur ausgelesen und dann wird das Backup erstellt und die ausgelesenen Daten werden in dem Backup gespeichert.
- data_analysis: Das data_analysis Notebook ist für die Analyse der Daten gedacht. Nach den Imports folgen die verwendeten Funktionen in dem Notebook. Danach dient der Code dazu, die Daten aus dem Backup auszulesen, sie zu konkatenieren, da die Daten tageweise ausgelesen wurden und die Korrelationen wochenweise beobachtet werden und sie zum Schluss ggf. noch vom Mittelwert zu befreien und die Daten Min-Max zu skalieren. Zur besseren Übersicht habe ich die Plots rausgenommen, die ich zum Ausprobieren und Kennenlernen der Daten erstellt habe, da sie unübersichtlich sind und nicht dem allgemeinen Verständnis dienen. Darauffolgend habe ich mich zuerst mit dem Zeitraum vom 03.10 – 11.10 beschäftigt mit den dazugehörigen Korrelationen. Danach habe ich mich mit dem Zeitraum ab dem 02.11 beschäftigt. Ab dem Teil von dem Notebook beginnt das Betrachten der Heatmaps, Korrelationen und Rohdaten. Als nächstes habe ich mit den Korrelationen mit allen vorliegenden Klimadaten beschäftigt. Zum Schluss habe ich mich noch mit der Feature Extraction und dem Clustering im Notebook beschäftigt.

3.4 Probleme bei den Experimenten

Das erste wirkliche Problem, welches mir bei meiner Analyse begegnet ist, dass ich mir die Frage gestellt habe, wie ich die Daten speichere, um später wieder auf sie zurückgreifen zu können. Da das Auslesen der Daten für einen Tag und einen Link circa 15 Minuten dauert, ist es nicht performant genug, sie jedes Mal neu auszulesen, weil ich zum Teil die Daten direkt für eine Woche für alle Links brauche oder auch für drei Wochen. Dieses Problem konnte ich durch das h5py Package lösen, welches Python mir bietet. Dadurch kann ich die Daten performant abspeichern und wieder laden, so muss ich die Daten nur einmalig auslesen. Ein weiteres Problem bezogen auf die Backups war, dass sollte das Notebook oder allgemein die remote-Verbindung abstürzen, kann es dazu kommen, dass die Daten, welche bis zu dem Zeitpunkt ausgelesen wurden, einfach verloren gehen und neu ausgelesen werden müssen. Noch ein Problem, welches sich bei mir bei der Speicherung der Backups ergeben hat, war auf meine Unaufmerksamkeit zurückzuführen. Da ich die Daten immer tageweise ausgelesen habe und dann nach dem Namensschema Bezeichnung des Links + Datum des betrachteten Tages abgespeichert habe, ist mir der Fehler zum Teil unterlaufen, dass ich den Tag angepasst habe, um den neuen Tag auszulesen, aber die Bezeichnung nicht geändert habe, wirft h5py dort ein Fehler auf, weil es diese Bezeichnung schon gibt. Der aufwändigere Fehler ist, dass man das Datum beim Auslesen nicht anpasst, aber die Bezeichnung ändert und dann nach dem normalen Schema weitermacht. Dieser Fehler fällt nicht direkt auf und nur bei genauerer Betrachtung der Daten, fällt es auf, dass die Daten doppelt vorliegen. Je nach dem, wann der Fehler auffällt, zieht sich die Dauer für das Ausbessern dieses Fehlers hin, weil die Analysen neu erstellt werden müssen, in denen diese Daten verwendet wurden. Ein weiteres Problem was aufgetreten ist, war dass ich bei der Mittelwertsbereinigung nicht darauf geachtet hatte, dass ich den Mittelwert von der kompletten Woche bilde, sondern dass ich den Mittelwert für jeden Tag gebildet habe und dann den Mittelwert von den Daten abgezogen habe und die Woche dann zusammengefügt habe. Dieser Fehler hat sich dann an dem Verhalten mancher Graphen bei der Korrelation gezeigt. Graphen die allgemein viel Bewegung aufweisen, haben auch einen sehr hohen Unterschied im Mittelwert über die Tage verteilt. Daraus haben sich dann große Sprünge ergeben innerhalb der Rohdaten zwischen den Tagen und wenn man sich nicht die Rohdaten für diesen Link anguckt, kann es sein, dass einem dieses Ergebnis nicht direkt in der

Korrelation auffällt, sondern dass man es als Ausreißer in den Daten wahrnimmt. Noch eine Schwierigkeit hat sich bei mir bei der Formatierung der x-Achse für die Korrelationsgraphen ergeben. Hier ging es darum, dass man in der Mitte der Achse den Nullpunkt hat und dann nach links und rechts auf der Achse die Zeit um seine vorherbestimmten Zeitschritte erhöht oder reduziert. Dies war für mich nicht direkt eingänglich und es hat mich einen gewissen Aufwand gekostet, um dieses Problem zu lösen. Noch ein Problem, welches ich oben angesprochen habe, liegt darin, dass die Korrelationen ohne die Mittelwertsbereinigung bei den Links der TIMING Kanäle ein sehr unübliches Bild für die Korrelationen ergeben haben. Nach Rücksprache in unseren wöchentlichen Meetings, kamen wir darauf, dass es an dem hohen Offset der TIMING Daten liegt und das es durch die Bereinigung des Mittelwerts behoben werden konnte, wie schon oben im Kapitel 2.4.2 beschrieben. Dies habe ich dann auch für die Links der CTRLOUT Kanäle durchgeführt, damit dieses Problem nicht auch bei den CTRLOUT Kanälen auftritt. Dies ist zwar unwahrscheinlich, weil der Offset sehr gering ist bei den CTRLOUT Kanälen, da sie sich immer in einem Grenzbereich bewegen von den Daten. Ein Problem mit der Performanz hat sich bei mir bei dem Berechnen der Korrelationen ergeben. Am Anfang hatte ich noch die Korrelation mit allen Datenpunkten berechnet, was einer ungefähren Laufzeit von zwei Stunden entsprach, um eine Korrelation zwischen zwei Links für eine Woche zu bilden. Danach bin ich dazu gegangen, dass ich zuerst jedes 100. Element für die Berechnung genommen habe und dann die Ergebnisse mit den Korrelationen, die ich davor berechnet habe, verglichen und bin zu dem Schluss gekommen, dass ich bei der Methode bleibe, weil die Ergebnisse identisch sind und die Performanz sehr viel besser wurde. Danach kam ich, nach Besprechung in unserem wöchentlichen Meeting, auf die Methode, dass ich die Korrelation per correlate von Scipy mit der Methode „fft“ bilde.

4 Zusammenfassung und Ausblick

Das Ziel der Arbeit war es eine Datenanalyse auf die Daten der Links des European XFEL durchzuführen. Hierbei ging es zuerst darum, die Daten zu sammeln, sie zu filtern und zu formatieren, anschließend wurde sich zuerst mit den Rohdaten beschäftigt, um zu gucken, ob es bei der Betrachtung schon Auffälligkeiten gibt. Nachfolgend wurden Korrelationen zwischen den Daten der Wochen gebildet und zusätzlich wurden noch die Klimadaten in die Korrelationen mit einbezogen. Zum Schluss wurde sich noch mit Clustering beschäftigt. Hier ging es darum, zu erkennen, welche Cluster sich bilden und ob diese zu erklären waren.

Als wichtigste Erkenntnisse aus der Arbeit und den Experimenten kann man zum einen das Entdecken der Oszillationen aus den Rohdaten (siehe Abbildung 17) sehen. Die Ursache für dieses Phänomen wurde auch in den gemeinsamen Meetings leider nicht gefunden. Hier könnte man für die zukünftige Experimente weiter ansetzen und gucken, ob man eventuell durch genauere Untersuchungen der Daten für diesen Zeitraum oder auch durch hinzuziehen von anderen Daten auf die Ursache dieser Oszillationen kommt. Eine weitere Erkenntniss war, dass sich die Links zum Teil wie erwartet verhalten haben (siehe Heatmaps Abbildungen 8,9,11-14), wie zum Beispiel, dass sich die Links, die im Beschleunigerteil im Tunnel enden, meistens sehr ähnlich Verhalten, aber auch dass sich zum Beispiel die Links 10.1 und 8.2 in ihrem Verhalten unterscheiden, obwohl sie ähnlich lang sind, aber ab einem bestimmten Punkt einen unterschiedlichen Weg nehmen. Hier könnte man wieder mit weiteren Untersuchungen ansetzen, warum der Link 8.2 sich ähnlich zu den Links verhält, die im Tunnel enden, aber der Link 10.1 weicht sehr stark in seinem Verhalten von den anderen Links ab. Hier könnte es hilfreich sein, wenn man Daten für einen größeren Zeitraum sammelt und dann dieselben Analysen über den Zeitraum laufen lässt und guckt, ob sich hier dieses Verhalten bestätigt. Weitere Erkenntnisse konnte aus den Korrelationen mit den Klimadaten gewonnen werden. Zum einen, dass sich viele Klimadaten gar nicht für Korrelationen geeignet sind, da die Sensoren eine zu geringe

Auflösung haben und die Daten gleichzeitig relativ konstant sind, sodass der Sensor über einen Zeitraum von einer Woche nur drei verschiedene Werte sendet. Bei den Sensoren, die eine hohe Auflösung haben und sich so für Korrelationen anbieten, hat sich hier auch das Bild ergeben, dass die Korrelationen bei den drei Links, die im Injektorgebäude enden, ein schwaches Korrelationsverhalten gezeigt hat und die Links 10.1 und 5.0, wie oben beschrieben, ein starkes Korrelationsverhalten aufweisen. An der Korrelation mit den Klimadaten könnte man auch in der weiteren Betrachtung der Daten ansetzen. Hier bietet es sich auch an den Zeitraum der Daten zu erweitern und ggf. ein Einbeziehen von anderen Klimadaten und mehreren Sensoren. Durch das Clustering hat sich das bisher beobachtete Bild bei den Korrelationen zu einem großen Teil bestätigt. Hier hat sich auch gezeigt, dass auch hier das Verhalten des Links 10.1 bestätigt wurde, da er meist allein einem einzigen Cluster zugeordnet wurde. Den Link 10.1 könnte man für die Zukunft genauer betrachten, um eine Ursache für sein Verhalten herauszufinden, da er bei allen Analysen durch sein Verhalten herausgestochen ist. Dies wäre ein Anhaltspunkt, weil der andere lange Link 8.2 (siehe Heatmaps Abbildungen 8,9,11-14) nicht in seinem Verhalten so heraussticht wie der 10.1 Link.

Literaturverzeichnis

[1] Miller, Kyle; Dubrawski, Artur: *System-Level Predictive Maintenance: Review of Research Literature and Gap Analysis* (2019) URL https://arxiv.org/pdf/2005.05239.pdf (Zugriffsdatum: 13.03.2021)

[2] URL https://www.desy.de/forschung/index_ger.html (Zugriffsdatum: 13.03.2021)

[3] URL https://www.desy.de/forschung/beschleuniger/index_ger.html (Zugriffsdatum: 13.03.2021)

[4] URL https://www.desy.de/forschung/anlagen__projekte/european_xfel/index_ger.html (Zugriffsdatum: 13.03.2021)

[5] DESY XFEL Project Group: *The European X-Ray Free-Electro Laser Technical design report* (2007) [S. 41-48] URL https://bib-pubdb1.desy.de/record/77248/files/european-xfel-tdr.pdf (Zugriffsdatum: 16.03.2021)

[6] Senthilnathan, Samithambe: *Usefulness of correlation analysis* (2019) https://www.researchgate.net/publication/334308527_Usefulness_of_Correlation_Analysis (Zugriffsdatum: 16.03.2021)

[7] Meyer, Martin: *Anhang B zum Buch: Signalverarbeitung Analoge und digitale Signale, Systeme und Filter*. Heidelberg: Springer (2011) [S. 30] URL https://www.springer.com/cda/content/document/cda_downloaddocument/v_41_4909.pdf%3FSGWID=0-0-45-1362795-p174302740

[8] Oberst, Ulrich: *The Fast Fourier Transform* (2007) URL https://www.researchgate.net/publication/220259110_The_Fast_Fourier_Transform (Zugriffsdatum: 16.03.2021)

[9] Omran, Mahamad; Engelbrecht, Andries; Salman, Ayed A.: *An overview of clustering methods* (2007) URL https://www.researchgate.net/publication/220571682_An_overview_of_clustering_methods (Zugriffsdatum: 16.03.2021)

[10] URL https://ttfinfo.desy.de/DOOCSWiki/Wiki.jsp?page=PythonDAQClientInterface (Zugriffsdatum: 13.03.2021)

[11] URL https://docs.h5py.org/en/stable/ (Zugriffsdatum: 13.03.2021)

[12] URL https://www.kite.com/python/docs/pandas.HDFStore (Zugriffsdatum: 13.03.2021)

[13] URL https://pypi.org/project/hickle/ (Zugriffsdatum: 13.03.2021)

[14] URL https://numpy.org/doc/stable/reference/generated/numpy.correlate.html (Zugriffsdatum: 13.03.2021)

[15] URL https://seaborn.pydata.org/generated/seaborn.heatmap.html (Zugriffsdatum: 13.03.2021)

[16] URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.correlate.html (Zugriffsdatum: 13.03.2021)

[17] URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (Zugriffsdatum: 13.03.2021)

[18] URL https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (Zugriffsdatum: 13.03.2021)

[19] URL https://tsfresh.readthedocs.io/en/latest/api/tsfresh.feature_extraction.html (Zugriffsdatum: 13.03.2021)

[20] URL https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html (Zugriffsdatum: 13.03.2021)

[21] URL https://numpy.org/doc/stable/reference/generated/numpy.interp.html (Zugriffsdatum: 16.03.2021)

A Anhang

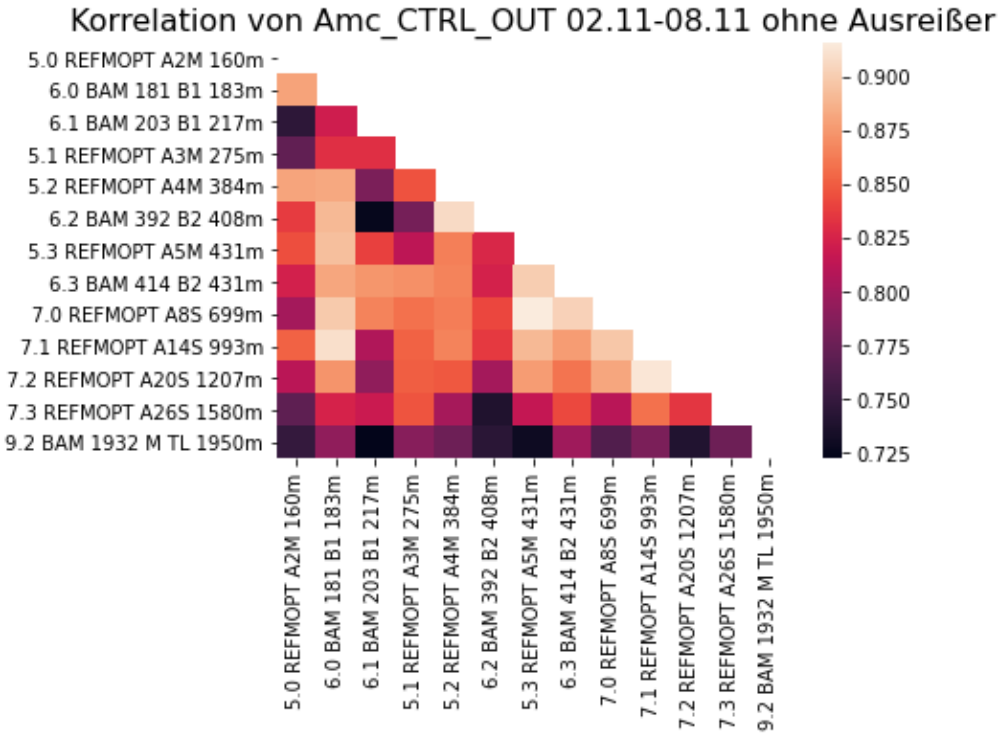

Abbildung 26 Heatmap vom 02.11-08.11 für CTRL OUT ohne Ausreißer

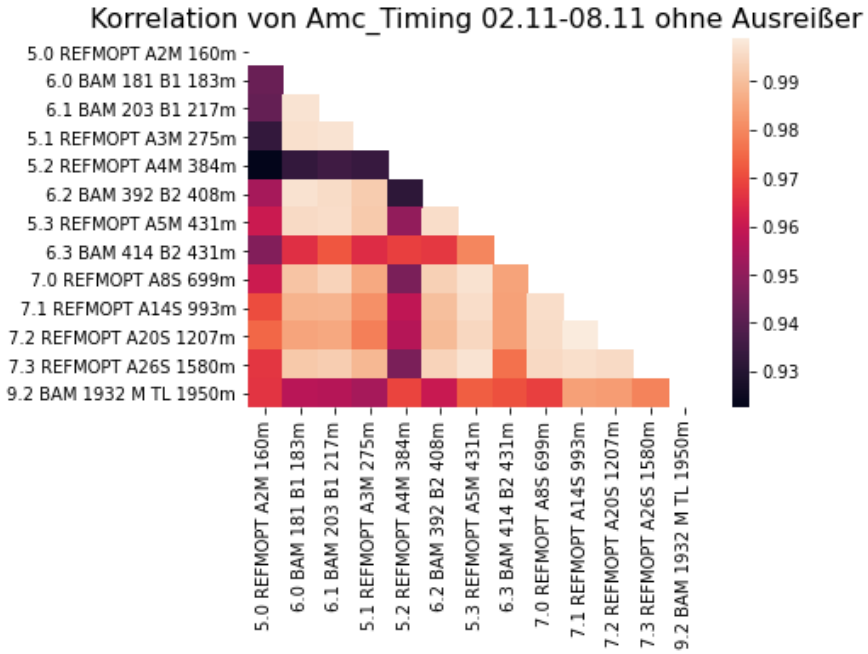

Abbildung 27 Heatmap vom 02.11-08.11 für TIMING ohne Ausreißer

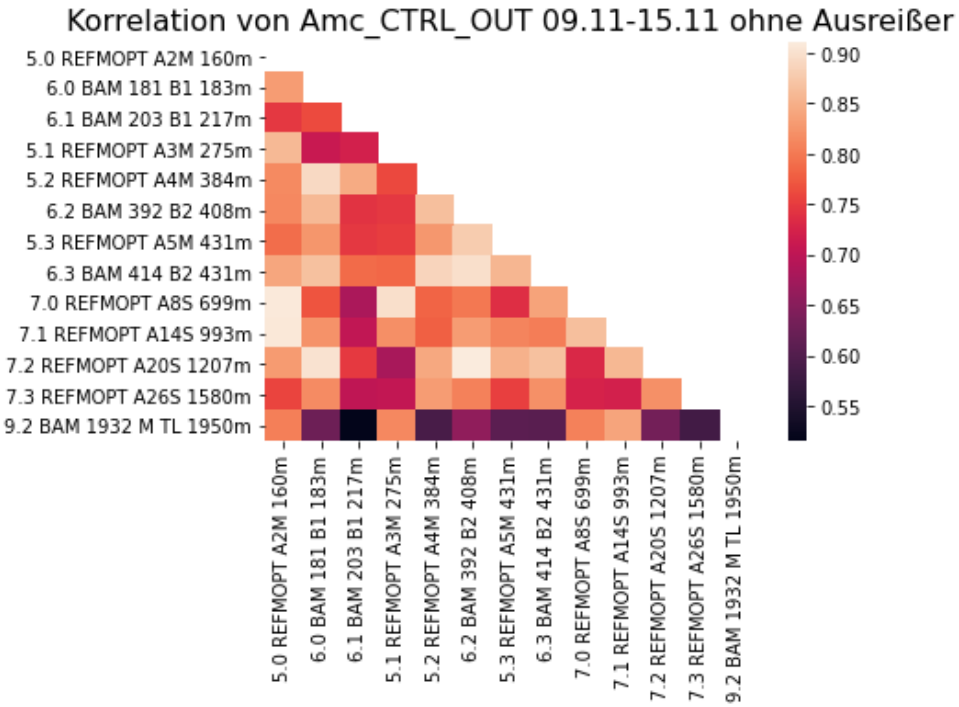

Abbildung 28 Heatmap vom 09.11-15.11 für CTRL OUT ohne Ausreißer

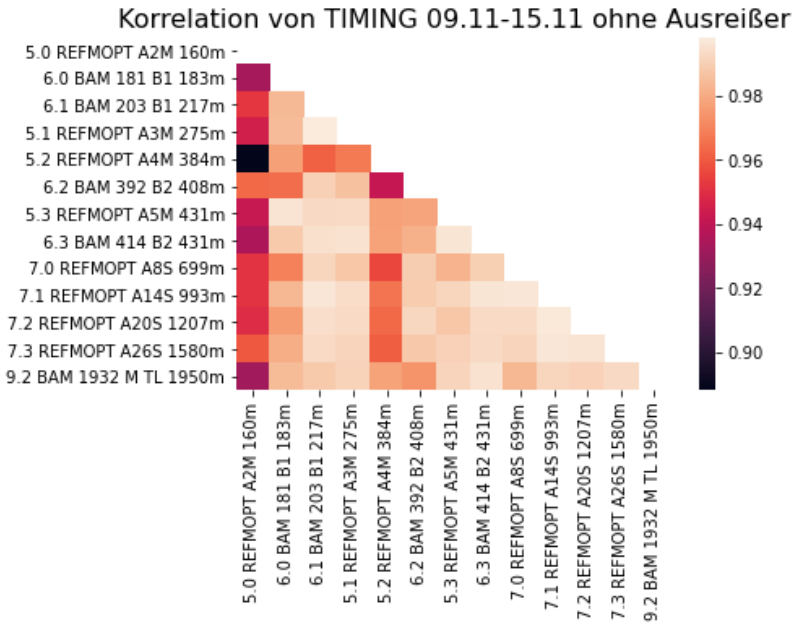

Abbildung 29 Heatmap vom 09.11-15.11 für TIMING ohne Ausreißer

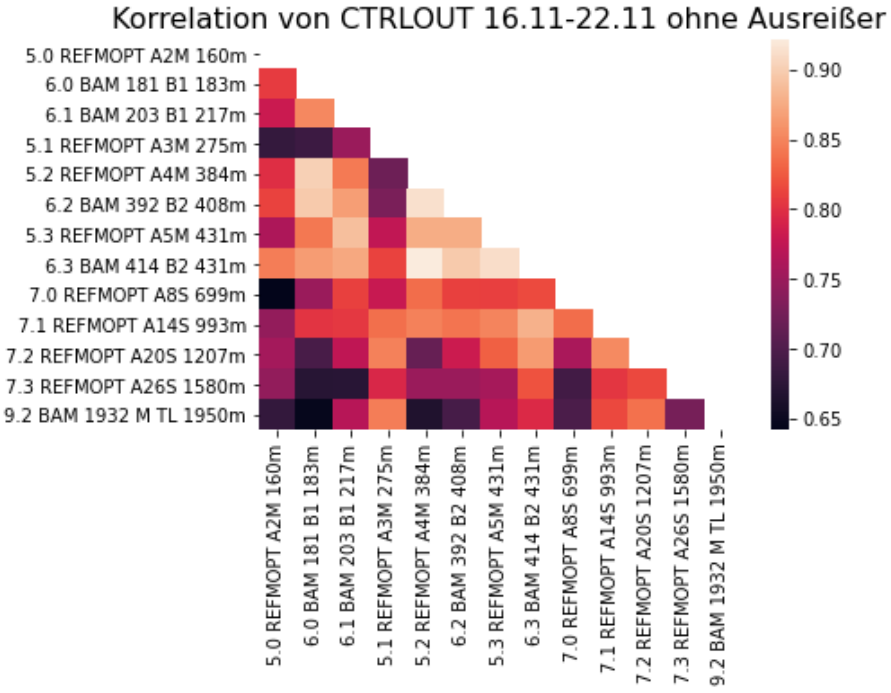

Abbildung 30 Heatmap vom 16.11-22.11 für CTRL OUT ohne Ausreißer

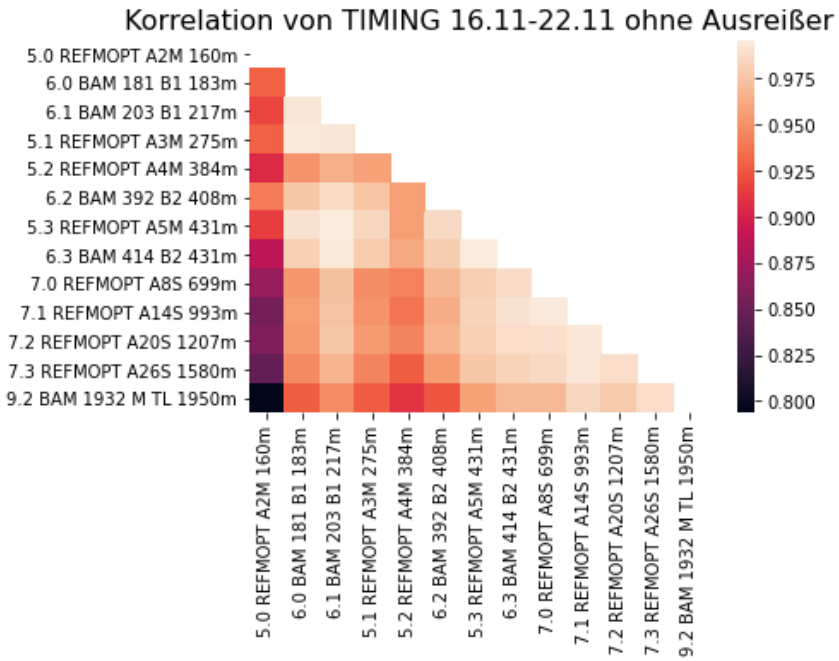

Abbildung 31 Heatmap vom 16.11-22.11 für TIMING ohne Ausreißer

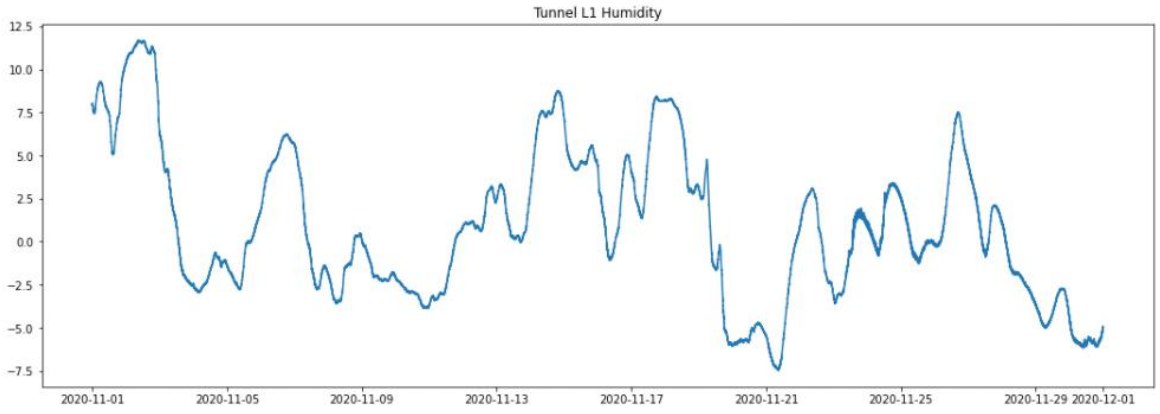

Abbildung 32 Rohdaten für Luftfeuchtigkeit vom Sensor L1

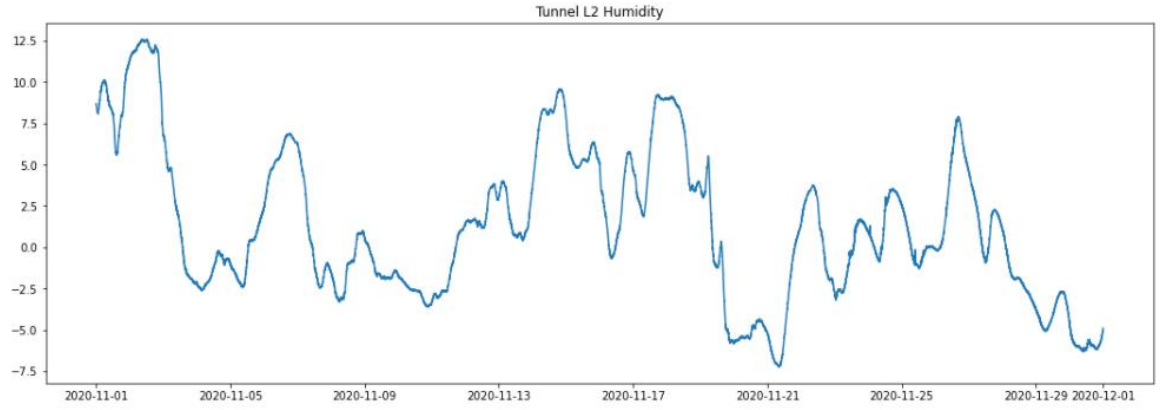

Abbildung 33 Rohdaten für Luftfeuchtigkeit vom Sensor L2

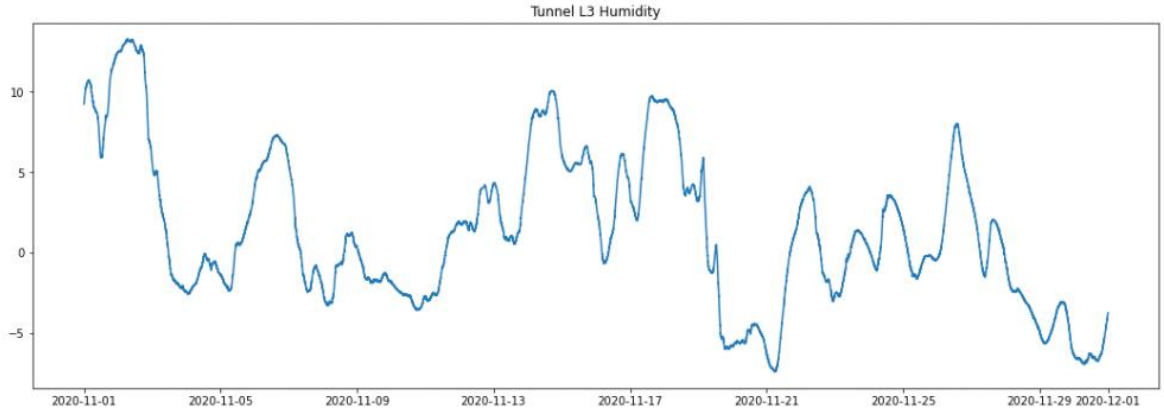

Abbildung 34 Rohdaten für Luftfeuchtigkeit vom Sensor L3

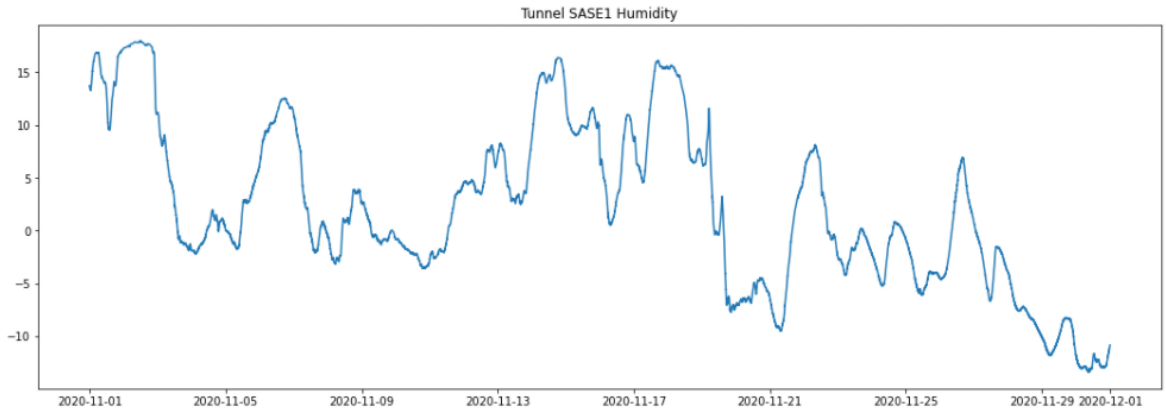

Abbildung 35 Rohdaten für Luftfeuchtigkeit vom Sensor SASE1

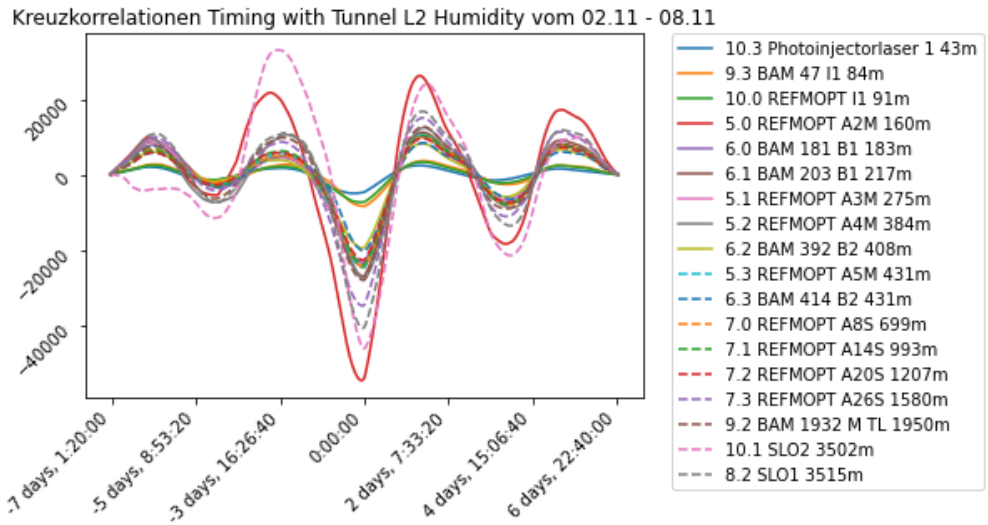

Abbildung 36 Kreuzkorrelation vom 02.11-08.11 für Sensor L2

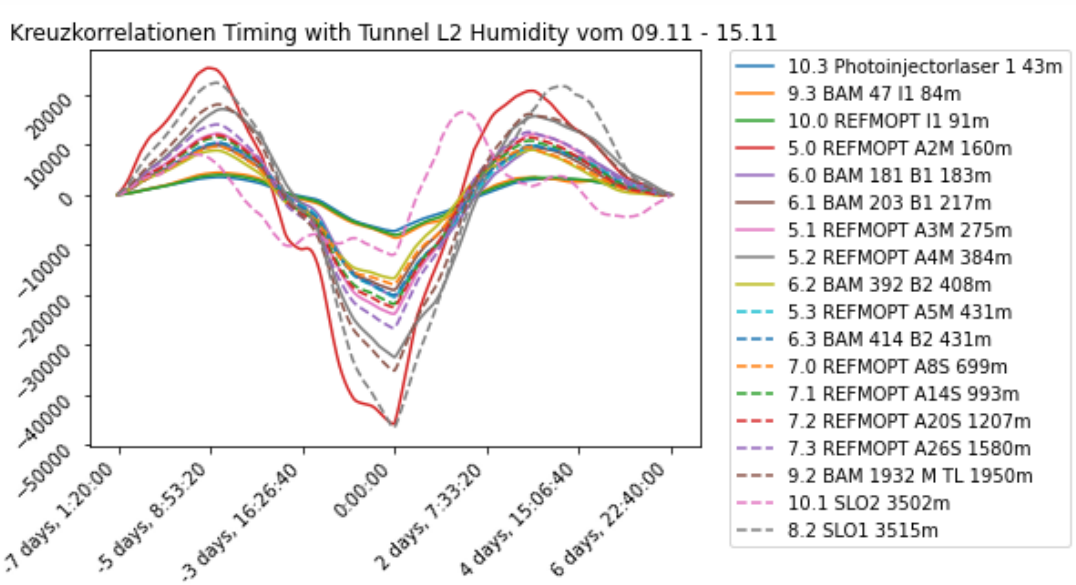

Abbildung 37 Kreuzkorrelation vom 09.11-15.11 für Sensor L2

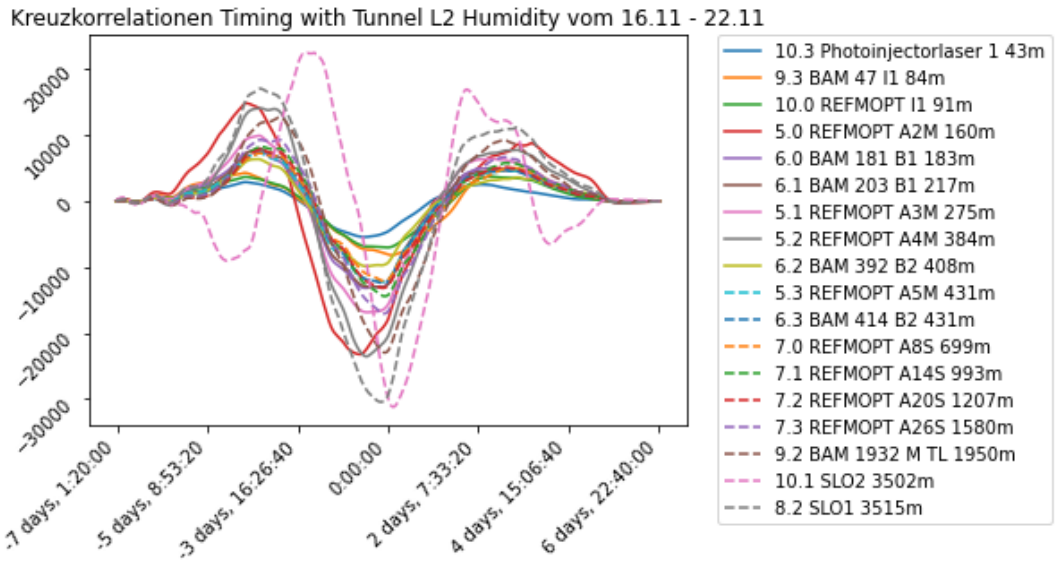

Abbildung 38 Kreuzkorrelation vom 16.11-22.11 für Sensor L2

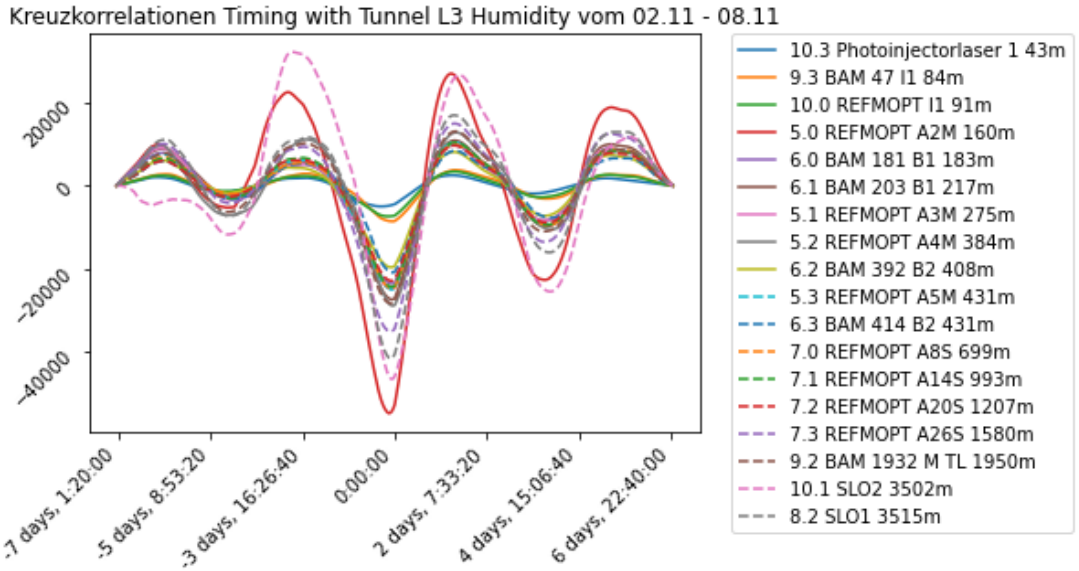

Abbildung 39 Kreuzkorrelation vom 02.11-08.11 für Sensor L3

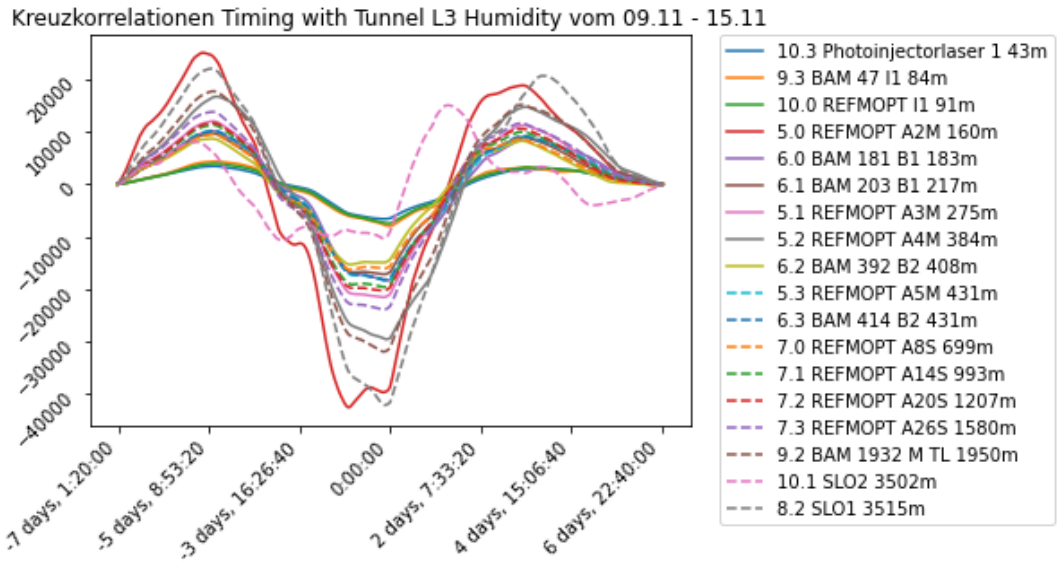

Abbildung 40 Kreuzkorrelation vom 09.11-15.11 für Sensor L3

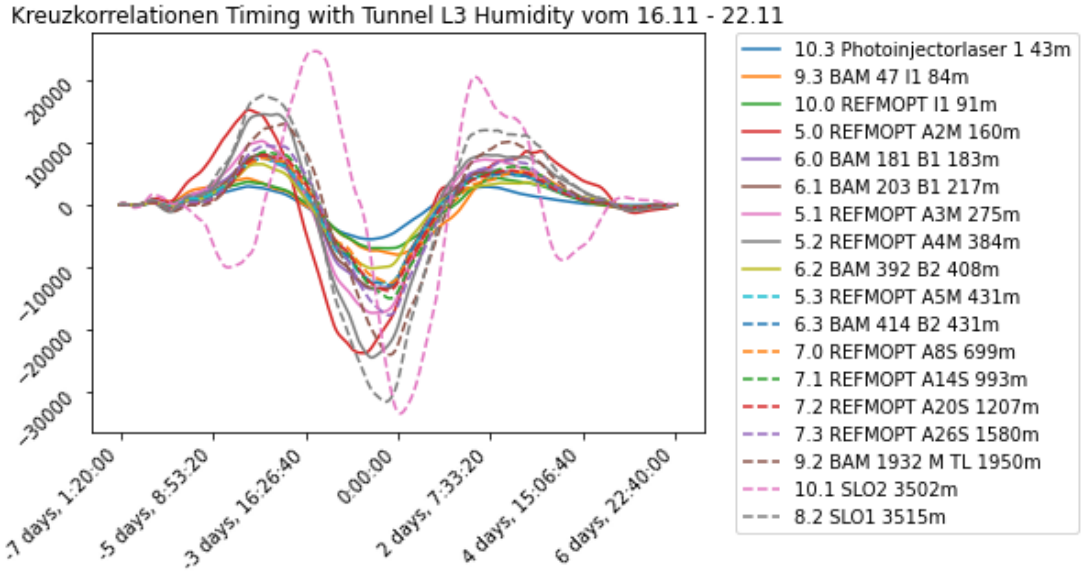

Abbildung 41 Kreuzkorrelation vom 16.11-22.11 für Sensor L3

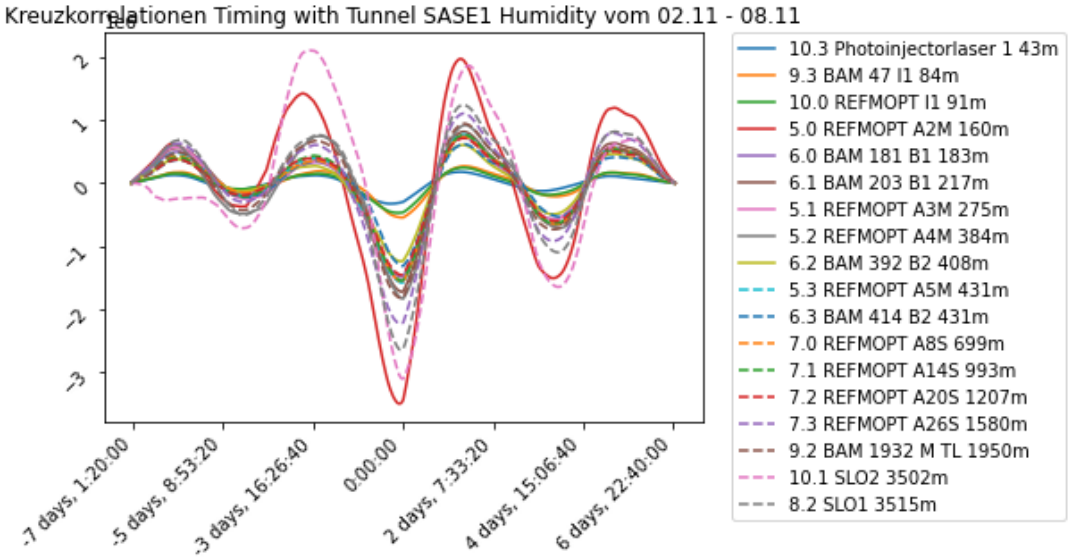

Abbildung 42 Kreuzkorrelation vom 02.11-08.11 für Sensor SASE1

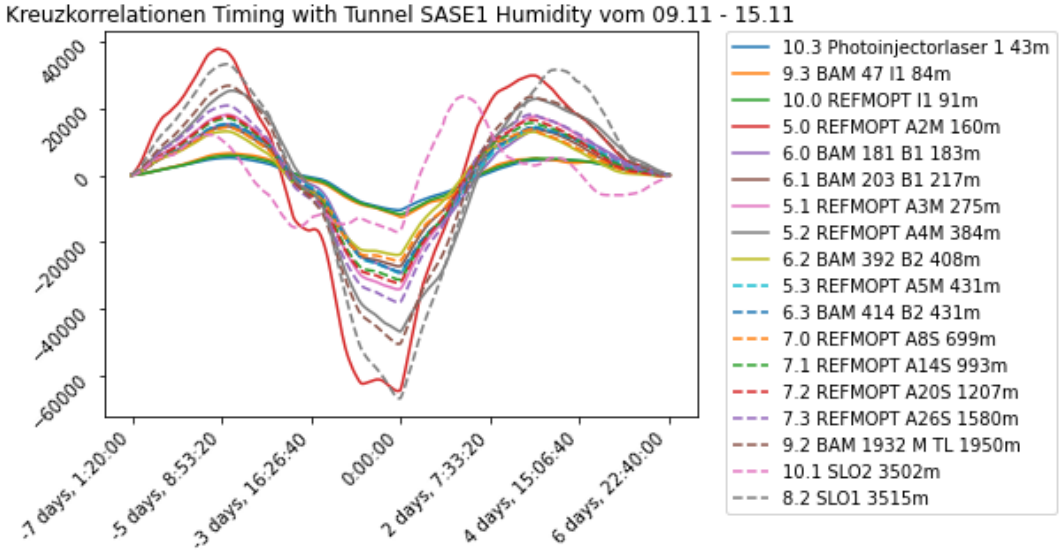

Abbildung 43 Kreuzkorrelation vom 09.11-15.11 für Sensor SASE1

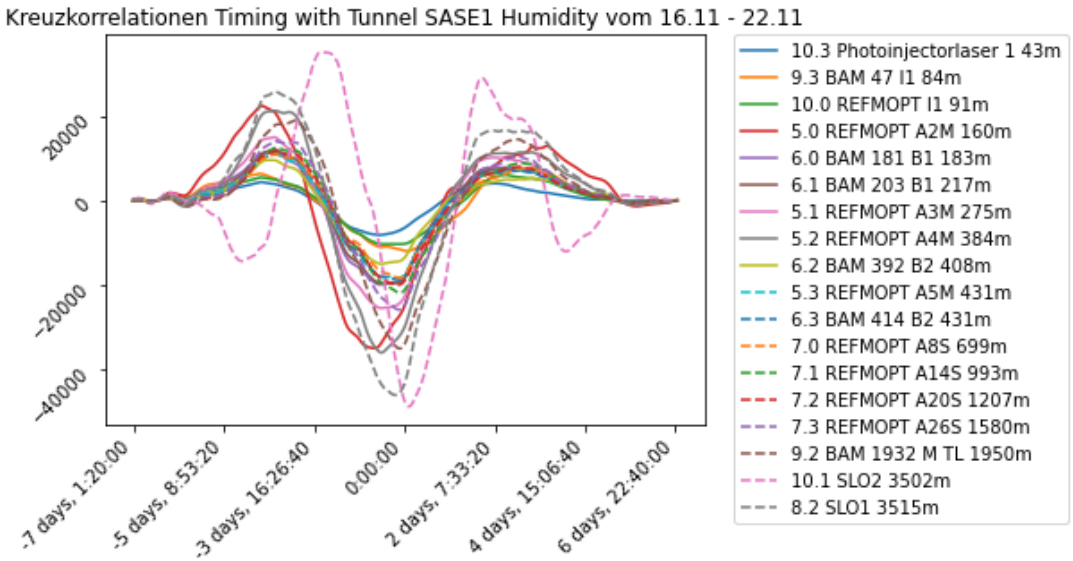

Abbildung 44 Kreuzkorrelation vom 16.11-22.11 für Sensor SASE1

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original