

Technischer Bericht

Zyklussynchrone Temperaturmessung für
Kunststoffspritzgiessmaschinen

Studienarbeit HS15

Betreuer: Prof. Oliver Augenstein

Autoren: Luca Aquino, Silvan Habegger

Inhaltsverzeichnis

1	Allgemein	3
1.1	Aufgabenstellung	4
1.2	Abstract	6
2	Management Summary	7
2.1	Ausgangslage	7
2.2	Vorgehen, Technologie	7
2.3	Ergebnisse	7
2.4	Ausblick	7
3	Technischer Bericht	8
3.1	Ausgangslage	8
3.1.1	Maschinenkomponenten	8
3.1.2	Funktionsweise	8
3.2	Problembeschreibung	9
3.2.1	Schwächen der heutigen Situation	9
3.2.2	Massnahmen	9
3.2.3	Nicht-Funktionale Anforderungen	10
3.2.4	User-Stories	11
3.2.5	Bugs	14
3.2.6	Testing	15
3.2.7	Tasks	15
3.3	Lösungskonzept	15
3.3.1	Erwartetes Resultat	15
3.3.2	Entscheidungsfindung	15
3.4	Software-Architektur & -Design	16
3.4.1	Frontend	17
3.4.2	Backend	17
3.4.3	Datenbank	21
3.4.4	Embedded System	21
3.5	Analyse	22
3.5.1	Lineare Regression	22
3.5.2	Overfitting	23
3.5.3	Crossvalidation	24
3.6	Umsetzung	26
3.6.1	Generierung synthetischer Daten	26
3.6.2	Analyse Realdaten	27
3.6.3	Lineare Regression mit Fourier-Komponenten	29
3.7	Datenanalyse Tools	33
3.7.1	Messdatenauswertung	33
3.7.2	Resultate	33
3.7.3	Weiterführende Informationen	34
3.8	Ergebnisdiskussion	35
3.8.1	Entwickeltes Verfahren	35
3.8.2	Fazit Technologiewahl	35
3.9	Ausblick	36

3.9.1	Kalman-Filter	36
4	Benutzerhandbuch	37
4.1	Übersicht	37
4.2	Vorgehen	38
4.2.1	Messung laden	38
4.3	Zeitfensterauswahl	39
4.3.1	Übersicht	39
4.3.2	Zonenauswahl	39
4.3.3	Timeline	39
4.3.4	Datepicker	40
4.3.5	Vergleich von Timeline zu Datepicker	40
4.4	Messdaten Auswertung	41
4.4.1	Plot	41
4.4.2	Messlinie	41
4.4.3	Lineare Regression	42
4.4.4	Zyklus-Starts	42
4.4.5	Prediction Intervall	42
4.4.6	Ergebniskurve	43
4.4.7	Innerzyklische Schwankungen	43
4.5	Resultate	44
4.5.1	Plot	44
4.5.2	Ergebniskurve	44
4.5.3	Globale Terme	44
4.5.4	Moving Average	45
4.5.5	Zyklus-Starts	45
4.6	Weiterführende Informationen	46
4.6.1	Intervall	46
4.6.2	Trends	47
4.6.3	Benchmarks	47
5	Anhang	49
5.1	Installationsanleitung	49
5.1.1	Workspace: Frontend	49
5.1.2	Workspace: Backend	49
5.1.3	Datenbank	51
5.1.4	Server: Frontend	54
5.1.5	Server: Backend	54
5.2	Semantisches Sprachmodell	55
5.3	Test-Logs	56
5.4	Projektplanung	56
5.4.1	Projektverlauf	56
5.4.2	Projektaufwand/Zeitabrechnung	58
5.5	Persönliche Berichte	61
5.5.1	Luca Aquino	61
5.5.2	Silvan Habegger	61
5.6	Konfigurationen	63
5.6.1	Apache2	63
5.6.2	Tomcat	63
5.7	Formulare	64
5.7.1	Eigenständigkeitserklärung	64

Kapitel 1

Allgemein

1.1 Aufgabenstellung

Zyklussynchrone Temperaturmessung für Kunststoffspritzgiessmaschinen

Studiengang: Informatik (I)
 Semester: HS 2015/2016 (14.09.2015-14.02.2016)
 Durchführung: Bachelorarbeiten, Studienarbeiten

Fachrichtung: System-Software
 Institut: Diverses
 Gruppengrösse: 2 Studierende
 Status: zugewiesen

Verantwortlicher: Augenstein, Oliver
 Betreuer: Augenstein, Oliver
 Gegenleser: [Nicht definiert]
 Experte: [Nicht definiert]
 Industriepartner: Netstal

Ausschreibung: **Problemdomäne / Hintergrund**

Auf Kunststoffspritzgiessmaschinen sind an verschiedenen Stellen, beispielsweise in der Plastifizierung oder im Werkzeug, Heizungen vorhanden, deren Temperaturen von der Maschinensteuerung geregelt werden. Die kleinste Einheit die geregelt wird heisst Temperaturzone. Eine Temperaturzone besteht meistens aus Heizung und Temperatursensor und ist in der Maschine an einer festen Stelle installiert. Die Maschinensteuerung enthält Komponenten, um die Temperaturen der einzelnen Zonen zu messen und Stellglieder um die Heizungen anzusteuern.

Die vorhandene Hardware misst die Temperaturen in jeder Zone mit einer Periodendauer von 640ms. Eine Plastifiziereinheit kann bis zu ca. 20 Zonen enthalten. Ein Werkzeug kann bis zu ca. 200 Zonen enthalten. Temperaturen werden mit einer Auflösung von 0.1°C gemessen.

Beim Spritzgiessen spricht man vom „Zyklus“, durch den der Zeitraum in dem der Spritzgiessvorgang beschrieben wird, in dem mit der Fertigung eines Werkstück (n) begonnen und beendet wird bis zu der Zeit, an der die Fertigung des nächsten Werkstücks (n+1) beginnt. Gängige Zykluszeiten auf unseren Maschinen liegen bei 5s bis >30s.

Der Start eines Zyklus ist nicht synchron mit der Temperaturmessung. Er hängt von dem vom Kunden programmierten Spritzgussablauf ab. Im Ablauf wiederum hängen die Zeiten von physikalischen Vorgängen (Druck, Geschwindigkeit, usw.) ab. D.h. die Zykluszeiten weisen immer kleine Schwankungen auf.

Problemstellung

Für die Qualitätssicherung beim Spritzgiessprozess ist es interessant, prozessspezifische Messwerte festzuhalten und Trends verfolgen zu können. Dabei wird der Anspruch gestellt, dass sich alle Messwerte auf Zykluszeiten beziehen und in ganz spezifischen Fällen sogar auf Zeiten innerhalb eines einzelnen Zyklus (= Werkstück).

Da es wie oben erwähnt, keine Synchronität zwischen Zyklus und Temperaturmessung gibt, muss davon ausgegangen werden, dass ein auf den Zyklusbeginn bezogener Messwert tatsächlich mit einer Streuung von 640ms auf der Zeitachse erfasst wird.

Die Zonen sind zwar thermisch träge, trotzdem ist bekannt, dass die durch den Zyklus

ausgelösten Temperaturänderungen einen Einfluss auf das Messresultat in dieser Zeitspanne haben.

Der Jitter auf der Zeitachse führt somit auch zu Jitter auf der Temperaturachse. Dieser Jitter ist so gross, dass bisher auf eine Auswertung der Temperaturmesswerte verzichtet wurde.

Aufgabe

Die Messdaten sollen durch eine Software mit statistischen Methoden so mit den Zykluszeiten synchronisiert werden, dass an den verschiedenen Zonen zyklussynchrone Temperaturmessungen aufgezeichnet werden können.

Da bei jedem Zyklus die Messung innerhalb der Periode von 640ms an einem anderen Ort liegen kann, füllen sich bei genügend Zyklen mehr und mehr Messwerte innerhalb einer Periode ein und geben ein immer genaueres Abbild. Voraussetzung dabei ist, dass sich die Temperatur von Zyklus zu Zyklus nicht ändert.

Da aber genau diese Voraussetzung nicht gegeben ist - wir möchten ja einen Trend erfassen können - braucht es erweiterte Auswertungen.

Im Zuge der Arbeit soll eine OPC-UA-basierte Client-Server-Anwendung bestehend aus einem Laptop und einem Raspberry Pi2 entstehen, die das Verschicken von vorgängig produzierten Messdaten emuliert und das Entgegennehmen, statistische Aufarbeiten und Darstellen dieser Messdaten mit einem geeigneten User Interface demonstriert. Insbesondere sollen die folgenden Punkte erreicht werden:

- Bestimmung und Darstellung des Temperaturverlaufs jeder Zone über die Zeitachse mit einer hohen Auflösung (z.B. $\sim 10\text{ms}$) relativ zum Zyklusbeginn. Die Zeitachse umfasst eine Zykluszeit. Zusätzlich ein Gütefaktor ob die Messungen nur eine kleine Varianz über die Zeit aufweisen. Die Bestimmung der Anzahl Zyklen bis eine verlässliche Kurve vorhanden ist, soll adaptiv sein.
- Verlauf der Temperatur über die Zeitachse (Trend). Anfang und Ende der Zeitachse können frei gewählt werden. Der Zeitpunkt, ab wann der Messwert ab Beginn des Zyklus erfasst werden soll, kann gewählt werden (Sampling Point). Für die Darstellung werden diejenigen Messungen berücksichtigt die in einem Toleranzfenster (z.B. $\pm 5\text{ms}$) um den Sampling Point herum gemessen wurden.

Eingesetzte Technologie

Auf Grund der Komplexität der Maschinen ist ein direktes Arbeiten mit Spritzgussmaschinen nicht möglich. Aus diesem Grund wird die Schnittstelle zur Maschine mit Hilfe von Testdaten, die an einen Raspberry Pi2 geliefert werden, emuliert.

Die eingesetzten Technologie sind Java (Client), OSGI, OPC-UA (Kommunikationsprotokoll zur Spritzgussmaschine), Linux und Raspberry Pi2.

Voraussetzungen: Interesse an der Entwicklung von hardwarenahen Schnittstellen (OPC-UA, Raspberry Pi2) und an statistischen Datenauswertungen.



1.2 Abstract

Ausgangslage

Beim Plastikspritzgiessverfahren kommt es aufgrund von wiederholten mechanischen Abläufen zu Temperaturschwankungen innerhalb von Heizelementen. Die Elemente erhitzen den Plastik und bringen ihn auf eine optimale Konsistenz. Treten auf Basis von äusseren Einflüssen Temperaturschwankungen auf, wird versucht, diese durch einen Temperaturregler auszugleichen. Dadurch wird die Instabilität der Temperatur noch mehr verstärkt.

Vorgehen / Technologien

Damit diesem Phänomen entgegen gewirkt werden kann, wird ein Verfahren erarbeitet, das die wiederholten Schwankungen aus dem Temperatursignal entfernt. Mit einem eigens programmierten R-Skript kann eine lineare Regression auf einem Datenbereich durchgeführt, sowie deren Güte ermittelt werden. Dieses Verfahren ist sowohl für emulierte Daten, als auch für reale Temperaturmessungen anwendbar. Über eine Java-Webapplikation wird auf diese Funktion zugegriffen.

Ergebnisse

Die Ergebnisse des Projektes umfassen Auskunft über die Charakteristiken und Zusammenhänge der von Netstal gelieferten Daten, wie auch ein unter bestimmten Voraussetzungen anwendbares Verfahren zur Nachbildung von Temperaturverläufen.

Im weiteren Verlauf der Studienarbeit wurde ein webbasiertes Analyse-Werkzeug implementiert, das sich um die Aufbereitung von Daten kümmert und deren Auswertung über einen selektierbaren Zeitbereich ermöglicht. Als Auswertungsmethoden kommen das entwickelte Verfahren zur globalen Trend- und innerzyklischen Charakteristik-Analyse sowie Moving Average zum Einsatz. Ausserdem wird über die Verlässlichkeit der Verlaufs-Nachbildung (ermittelt durch statistische Tests) Auskunft gegeben.

Kapitel 2

Management Summary

2.1 Ausgangslage

Beim Plastikspritzgiessverfahren wird ein Granulat mittels Heizelementen erhitzt, bis es die optimale Konsistenz erreicht. Der Plastik durchläuft dabei mehrere Heizelemente, welche ihre Temperatur auf einen zuvor eingestellten Wert regeln. Während des Spritzgiessens können Temperaturschwankungen in diesen Heizelementen auftreten, welche die Temperaturregelung beeinflussen. Diese Unregelmässigkeiten können Auswirkungen auf die Konsistenz des Plastiks haben und zu einem Qualitätsverlust der Produkte führen.

Die Temperaturschwankungen, welche durch den Spritzgiessvorgang entstehen, können durch Anwenden mathematischer und statistischer Algorithmen herausgefiltert werden. Dadurch erhält man nur noch Temperaturänderungen, welche unabhängig von den Spritzgiessvorgängen sind.

2.2 Vorgehen, Technologie

Da das Spritzgiessen ein sich immer ähnlich wiederholender Ablauf ist, entsteht eine Temperaturschwankung, welche stets die gleiche Charakteristik aufweisen sollte. Durch ein mathematisches Verfahren wird versucht, diese Schwankung möglichst exakt nachzubilden, um sie aus dem Temperaturverlauf herauszurechnen.

2.3 Ergebnisse

Im Verlauf der Arbeit wurden zwei Verfahren entwickelt, welche unter bestimmten Umständen ¹ den Anteil von Spritzgiess-Schwankungen in der Temperatur minimiert oder sogar ganz entfernt. In einem Webtool können, zuvor in die Anwendung geladene, Daten mit den beiden Verfahren analysiert und verglichen werden.

2.4 Ausblick

Um die Regelung nachhaltig stabiler zu machen, wird empfohlen die aktuelle Vorgehensweise mit einem anderen Ansatz zu ersetzen ².

¹siehe Kapitel 3.6.2

²siehe Kapitel 3.9.1

Kapitel 3

Technischer Bericht

3.1 Ausgangslage

Netstal entwickelt und baut Maschinen, welche Produkte aus Plastik mittels Spritzgiessen herstellt. Dabei wird das verflüssigte Plastik in eine Form gespritzt und erhärtet nach der Abkühlung zum gewünschten Produkt.

3.1.1 Maschinenkomponenten

Gussformen

Die Gussform der Maschine besteht aus zwei massiven Metallklötzen mit Hohlräumen, die als Negativ zum Produkt fungieren. Diese werden unter hohem Druck¹ aneinander gepresst, um einen möglichst hohen Einspritzdruck zu ermöglichen, was wiederum schnellere Fertigungszyklen erlaubt.

Plastifiziereinheit

Die Plastifiziereinheit der Maschine bringt das, zu Beginn noch als Granulat vorhandene, Plastik auf die richtige Temperatur und Viskosität. Dabei wird es durch eine rotierende Spirale kontinuierlich durch mehrere Temperaturzonen² transportiert. Diese Zonen bestehen aus einem Heizelement mit Sensor. Für jede Zone kann eine eigene Temperatur eingestellt werden. Die Sensoren messen Temperaturen mit einer Auflösung von 0.01°C in einer fest definierten Periode (je nach Werkstück unterschiedlich). Eine Maschine kann bis zu 200 Zonen enthalten.

Werkzeugzonen

Die Aufgabe der sogenannten Werkzeugzonen liegt nicht wie bei der Plastifizierung im Schmelzen des Granulats, sondern bei der Aufrechterhaltung einer konstanten Temperatur des bereits verflüssigten Plastiks. Die Werkzeugzonen befinden sich zwischen der Plastifiziereinheit und der Ausstosskomponente und sind in der Grafik als Zone 401-404 bezeichnet.

3.1.2 Funktionsweise

Das Granulat durchläuft die verschiedenen Temperaturzonen, wird dabei verflüssigt, in die Gussform gespritzt und erhärtet schliesslich zum fertigen Produkt. Die Zykluszeiten hängen dabei von physikalischen Grössen wie z.B. Druck und Geschwindigkeit des Einspritzens ab und können variieren³. Die Temperaturmessung erfolgt nach einer festgelegten Periodenzeit, welche ebenfalls vom Produkt abhängt. Die Messzeiten sind nicht mit den Zyklen synchronisiert.

¹auch Schliesskraft genannt; Bis zu 10'000 kN

²kurz "Zone" genannt

³insbesondere sind verschiedene Zyklenlängen bei unterschiedlichen Produkte auszumachen

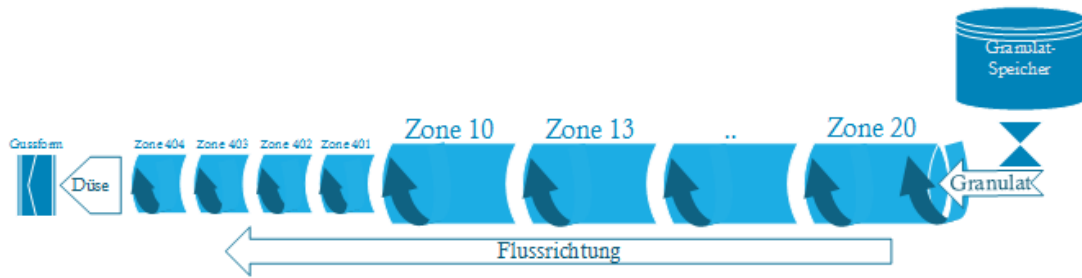


Abbildung 3.1: Abstrahierung der Maschine



Abbildung 3.2: Abstrahierung im Kontext der Maschine

3.2 Problembeschreibung

3.2.1 Schwächen der heutigen Situation

Problem

Durch das Verfahren beim Spritzgießen können zykluscharakteristische Temperaturschwankungen in den Zonen auftreten. Dies kann durch Wärmeübertragung⁴ der Gussform oder Druckänderung in der Plastifiziereinheit auftreten.

Auswirkung

Durch die Temperaturschwankung und die nicht mit dem Zyklus synchronisierten Temperaturmessungen entsteht für den Temperaturregler ein falsches Bild. Wird zum Beispiel die Temperaturmessung n in einem Moment gemacht, in welchem die Temperatur zyklusbedingt niedrig ist und die Messung $n+1$ während einer Phase mit erhöhter Temperatur gemacht wird, so entsteht ein Unterschied von bis zu 0.3°C . Der Regler versucht den Temperaturänderungen entgegen zu wirken und verschlimmert dadurch die ganze Situation weiter, da die Auswirkungen der veränderten Heizeinstellung verzögert einsetzen.

3.2.2 Massnahmen

Mittels statistischen Methoden kann die unerwünschte Schwankung aus dem Temperatursignal entfernt werden, um so den Einfluss der innerzyklischen Temperaturänderungen zu minimieren.

⁴beziehungsweise Kälteübertragung

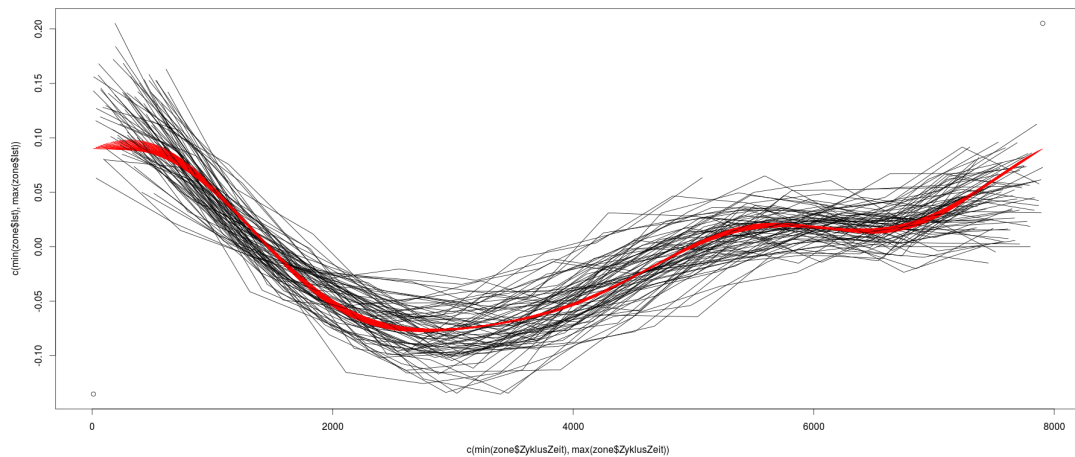


Abbildung 3.3: Innerzyklische Temperaturschwankung, Plot von Prof. Augenstein

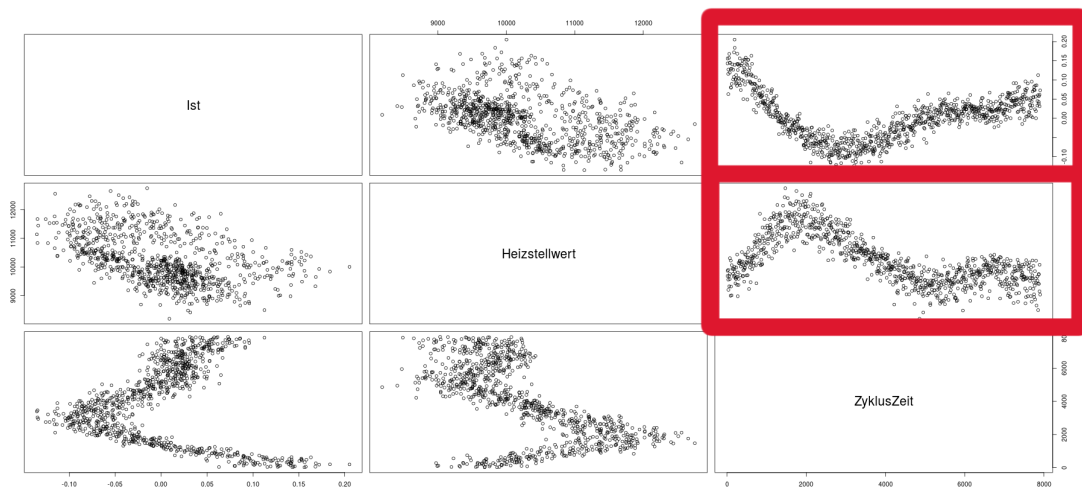


Abbildung 3.4: Ist- und Heizstellwert korrelieren umgekehrt zu einander, Plot von Prof. Augenstein

3.2.3 Nicht-Funktionale Anforderungen

Zugänglichkeit

Die Applikation soll von Personen ohne technischen Hintergrund (fachliches Wissen wird hingegen vorausgesetzt) bedient werden können. Zudem sollen die Auswertungen von mehreren Personen gleichzeitig einsehbar sein.

Priorität: *hoch*

Verfügbarkeit

An die Verfügbarkeit gelten keine speziellen Anforderungen. Ein Ausfall der Cloud-Applikation hat auch über längere Zeit keinen Einfluss auf andere Systeme.

Priorität: *keine*

Zugriffszeiten

Bei Zugriff auf die Applikation sollen keine längeren Wartezeiten entstehen; ein Request zur Berechnung einer linearen Regression über 700 Datenpunkte soll auf zeitgemässer Infrastruktur⁵ innerhalb von 1s verarbeitet werden.

Priorität: *niedrig*

Mengengerüst

Es wird durchschnittlich mit einem Zugriff pro Tag gerechnet (Qualitätssicherung). Die Applikation soll jedoch für mindestens 3 parallele Zugriffe ohne merkbare Leistungseinbrüche funktionieren.

Priorität: *niedrig*

3.2.4 User-Stories

Die User-Stories sind in diesem Projekt die zentralen Anforderungen und werden auch Issues genannt. Sie gehören in der Regel einem übergeordneten Epic⁶ an. Weitere Details zur Entwicklung der User-Stories während des Projektverlaufs sind unter 5.4 zu finden.

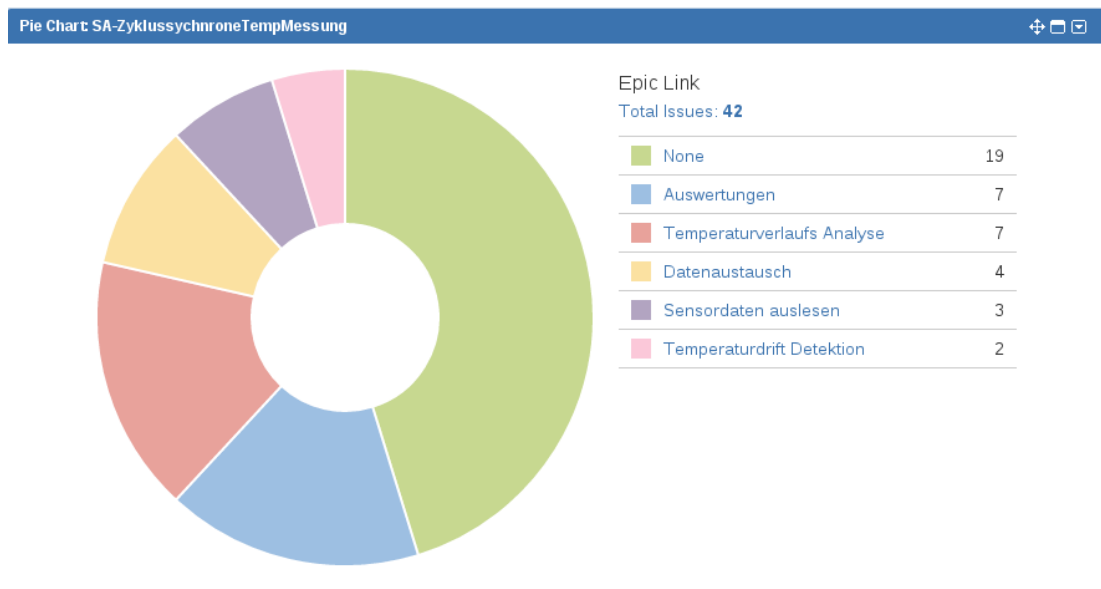


Abbildung 3.5: Aufteilung User-Stories nach Epics

SAZ-3 User-Stories betreffend Temperaturverlaufsanalyse (Epic)

Bündelt User-Stories, welche die Aufbereitung der Temperaturmessdaten beinhalten.

SAZ-4 User-Stories betreffend Auswertungen (Epic)

Bündelt alle User-Stories, die Auswertungsarten und Scopes der Temperaturverläufe betreffen.

SAZ-6 User-Stories betreffend Realtime Temperaturdrift-Detektion (Epic)

Bündelt User-Stories, die Realtime Temperaturdriftdetektionen beschreiben.

⁵2x 2.2GHz CPU und min. 2 GB RAM

⁶grosser, mehrere User-Stories umfassender, Aufgabenbereich

SAZ-7 User-Stories betreffend Auslesung Sensordaten (Epic)

User-Stories betreffend Emulation und Auslesen der Sensordaten. Entscheidend, ob das Verfahren nur auf reale Daten angewendet werden kann, oder die Möglichkeit mit synthetischen oder emulierten Daten zu arbeiten besteht.

SAZ-8 User-Stories betreffend automatisierten Datenaustausch (Epic)

User-Stories, die den automatisierten Datenaustausch zwischen Maschine und Cloud-Applikation betreffen.

SAZ-9 Kennzahlen mittlere und maximale Temperaturschwankung über definierbaren Zeitraum

Als Marketing (Netstal) möchte ich Temperaturverlaufs-Auswertungen zur maximalen und mittleren Temperaturschwankung über einen von mir definierbaren Zeitintervall erhalten, um treffendere Informationen zu den Maschinen liefern zu können.

SAZ-10 Zyklusbasierte Auswertung Temperaturverlauf

Als Hersteller (Netstal) möchte ich Auswertungen auf Zyklusbasis vergleichen können, um so Temperaturschwankungen zwischen Zyklen möglichst klein zu halten.

SAZ-11 Auswertung Temperaturschwankungen

Als Produktionsleiter (Kunde) möchte ich Temperaturverlauf in einem von mir definierten Zeitintervall angezeigt bekommen, um diese mit einer Auswertung von Produktionsmängel vergleichen und so Zusammenhänge aufdecken zu können.

SAZ-12 Zonenbasierte Auswertung Temperaturverlauf

Als Produktionsmechaniker möchte ich zonenbasierte Auswertungen der Temperatur erhalten, um Fehlkonfigurationen einer Heizeinheit aus dem Temperaturverlauf zu erkennen.

SAZ-13 Datenbereitstellung

Als Cloud Sytem möchte ich Anfragen zu den Temperaturdaten entgegennehmen und verarbeiten, um den Kommunikationspartner bestimmen zu lassen, welche Daten benötigt werden.

SAZ-14 Datenabholung

Als Cloud möchte ich Temperaturdaten vom Embedded System erhalten und abspeichern, um diese zu analysieren.

SAZ-15 Zeitraumbasierter Datendownload

Als Cloud möchte ich Anfragen ans Embedded System stellen können, um genau spezifizieren zu können, welche Daten ich will.

SAZ-16 Datenupload

Als Embedded System möchte ich die gespeicherten Temperaturdaten verschicken, um anderen den Zugriff darauf zu ermöglichen.

SAZ-17 Warnlampe bei Temperaturdrift

Als Maschinenbenutzer (Kunde) möchte ich mittels optischem System (Lämpchen) benachrichtigt werden, ob aktuell (Realtime) ein Temperaturdrift stattfindet, um die nötigen Schritte einzuleiten.

SAZ-18 Temperaturdrift Erkennung

Als Embedded System möchte ich mittels Statistik herausfinden, ob es aktuell (Realtime) einen Temperaturdrift gibt, um den Maschinenbenutzer zu informieren.

SAZ-19 Rekonstruktion Temperaturverlauf

Als Cloud / als Embedded System möchte ich den Temperaturverlauf einer Zykuslänge mithilfe von Messdaten mehrerer vorhergehenden Zyklen bestmöglich rekonstruieren, um eine höhere zeitliche Auflösung zu erhalten.

SAZ-20 Performante Temperaturverlaufsanalyse

Als Embedded System möchte ich die Rekonstruktion des Temperaturverlaufs möglichst performant durchführen, um meine knappen Ressourcen zu schonen.

SAZ-21 Synchronisation Zyklus - Messung

Als Entwickler möchte ich die Zyklen auf die Temperaturmessung synchronisieren, um die Position der Messung auf den normalen Temperaturverlauf der Maschine abbilden und dadurch Schwankungen erkennen zu können.

SAZ-22 Erkennung variabler Zykluslängen

Als Entwickler möchte ich die Auswertungen über mehrere Zyklen mit variablen Zykluslängen ermöglichen, um Temperaturverläufe noch genauer zu bekommen

SAZ-23 Emulation periodischer Temperaturmessung

Als Entwickler möchte ich in 640ms Perioden die Temperatur der Zone emulieren, um keine physikalische Maschine zu benötigen. Die verwendeten Daten sind generiert, sollen aber einen realistischen Temperaturverlauf emulieren.

SAZ-24 Sensordaten periodisch auslesen

Als Arduino möchte ich die Kommunikation zum Sensor übernehmen, um das Embedded System zu entlasten.

SAZ-28 Emulation Zyklusstart-Signal

Als Entwickler möchte ich, dass ein Zyklusstartsignal für Zyklen variabler Länge emuliert wird, um dadurch eine Bedingungen für zyklussynchrone Auswertung von Temperaturverläufen zu simulieren.

SAZ-29 Erstellung Webservice

Als Entwickler möchte ich ein Webservice Projekt, um damit ein Webtool zu entwickeln, welches mithilfe von R eine lineare Regression durchführt und die Daten mittels HTTP-Schnittstelle zur Verfügung stellt.

SAZ-30 Webtool GUI erstellen, Elterntask: SAZ-44

Als Entwickler möchte ich ein GUI-Projekt, um dem User die Daten des Webservices darzustellen.

SAZ-31 Auswertungsintervall spezifizieren

Als Webtool-Benutzer möchte ich die Zeitspanne der angezeigten Daten festlegen können, um speziell interessante Daten anzeigen zu lassen.

SAZ-32 Qualitätsmerkmale der Temperaturverlaufsrekonstruktion, Elterntask: SAZ-43

Als Benutzer des Webtools möchte ich Qualitätsmerkmale der Temperaturverlaufsrekonstruktion, um das Model beurteilen zu können.

SAZ-33 Ermittlung periodischer Temperaturverläufe

Als Entwickler wie auch als Auftraggeber (Netstal) möchte ich Auskunft über gelieferte Realdaten haben, um eine Aussage über die Machbarkeit des Projektes auf deren Basis treffen zu können.

SAZ-34 Darstellung Auswähl-Graph, Elterntask: SAZ-31

Als Benutzer der Webapplikation möchte ich die Zyklen, welche angezeigt werden sollen aus einem Graph auslesen, um besser entscheiden zu können welche wichtig sind.

SAZ-35 Anzeige Benchmarks, Elterntask: SAZ-43

Als Angestellter von Netstal möchte mir ein Bild über die von der Linearen Regression generierten Werte machen, um mir durch Vergleiche der bisherige Auswertungsmethoden und weiteren Benchmarks ermöglicht wird.

SAZ-36 Trendinformationen zu ausgewähltem Temperaturverlauf, Elterntask: SAZ-42

Als Netstal-Angestellter möchte ich genaue Informationen zu Trends (zyklusübergreifend) im Temperaturverlauf angezeigt bekommen, um mir ein Bild über deren Einfluss und Intensität zu machen zu können.

SAZ-37 Grafische Darstellung Zykluscharakteristik, Elterntask: SAZ-43

Als Benutzer des Webtools möchte ich durch eine grafische Aufbereitung der Charakteristik, periodische Temperaturschwankungen über den Zeitraum eines Zyklus angezeigt bekommen, um damit längere Temperaturverläufe besser verstehen zu können.

SAZ-42 Anzeigen Resultateplot

Als Benutzer des Webtools möchte ich den Temperaturverlauf ohne die innerzyklischen Schwankungen sehen, um den Temperatur Regler besser verstehen und konfigurieren zu können.

SAZ-43 Weiterführende Informationen Anzeigen

Als Benutzer des Webtools möchte ich mehr Informationen über lineare Regression und deren Güte wissen, um so besser entscheiden zu können ob diese gut war.

SAZ-44 Messdatenauswertung

Als Benutzer des Webtools möchte ich mir die Messdaten und die lineare Regression anzeigen lassen, um mir einen Überblick der ausgewählten Daten zu verschaffen.

3.2.5 Bugs

SAZ-38 Messdaten Synchronisierung (Bug)

Messdaten werden nicht korrekt synchronisiert (nur bei Aufbereitung für lineare Regression).

SAZ-40 Rechtschreibung und Beschreibungen Frontend (Bug)

Aufbereitung Frontend zur besseren Verständlichkeit für Kunde.

SAZ-41 Messdaten-Wolke in Charakteristik-Report ungenau (Bug)

Die Messdaten-Gruppierung im Zyklus-Charakteristik Graphen werden nicht korrekt berechnet.

3.2.6 Testing

SAZ-39 Testing

Als Entwickler möchte ich Unit-Tests ausführen, um die Funktionstüchtigkeit des Codes sicherzustellen.

3.2.7 Tasks

SAZ-25 Dokumentation

Sammeltask für Dokumentationsarbeiten.

SAZ-26 Meetings

Besprechungen mit Betreuer und Netstal.

SAZ-27 Projektmanagement

Task für in das Projektmanagement einzuordnende Tätigkeiten.

3.3 Lösungskonzept

Um Verläufe auf Basis einzelner Temperaturmessungen zuverlässig abbilden und vorhersagen zu können, wird anhand von linearer und nichtlinearer Regression die Machbarkeit dieser Aufgabe ermittelt. Dafür werden in ersten Schritten synthetische und emulierte Daten verwendet, um im weiteren Projektverlauf auf reale Datenabzüge und Temperaturmessungen zurückgreifen zu können.

Nach Entwicklung eines passenden Verfahrens mit einer auf mathematischen/statistischen Berechnungen spezialisierten Programmiersprache (R/Mathematica/MatLab) wird eine Auswertungsplattform für einen Server entwickelt, in den Datenabzüge von Maschinen importiert werden und diese durch voll umfängliche statistische Methoden aufbereitet.

3.3.1 Erwartetes Resultat

Die vorliegende Studienarbeit ist sowohl ein Software-, wie auch ein Forschungsprojekt. Primär wird die Entwicklung eines Verfahrens in Textform angestrebt, sowie die Implementation eines Webtools, das dessen Anwendung demonstriert. Das beschriebene Verfahren soll weiterführende Arbeiten ermöglichen, wie beispielsweise die Real-Time-Datenanalyse auf einem Embedded-System.

3.3.2 Entscheidungsfindung

Für die Cloud-Applikation bietet sich die Implementation einer Webapplikation in einer gängigen Sprache an. Dafür zur Auswahl stehende Technologien sind:

Sprache	Einarbeitung	Zusätzlicher Aufwand	Wartbarkeit	Risiko
Java	Sehr gering	Mittel-Hoch <ul style="list-style-type: none"> • Einrichtung Workspace <ul style="list-style-type: none"> – OR-Mapper – Dependency Injection • Aufsetzen CI 	Gut bis Sehr Gut	Klein
NodeJS	Mittel (Geringe Vorkenntnisse, gilt aber als einfach erlernbar)	Mittel <ul style="list-style-type: none"> • Aufsetzen von CI • Abklärungen R-Integration 	Mittel	Mittel bis Hoch

Trotz etwas höherem initialen Aufwand sieht das Team Java als die geeignetere Programmiersprache für das Projekt. Deshalb wird bei der Implementierung der Cloud-Applikation auf diese für uns vertraute Technologie gesetzt.

Datenaufbereitung

Um das im Verlauf des Projekts entwickelte Verfahren anwenden zu können, haben sich nach kurzer Recherche zwei Möglichkeiten zu dessen Implementation innerhalb der bereits festgelegten Rahmenbedingungen der Cloud-Applikation herauskristallisiert.

- **Renjin:** Ein Java-Framework, das die Ausführung von R-Skripten innerhalb der JVM ermöglicht
- **Eigene Implementation:** Native Entwicklung der linearen Regression in Java

Da sich das Team durch Einsatz des Renjin Frameworks grosse Zeitersparnisse verspricht, wird dieses für das Aufbereiten von Messdaten verwendet.

3.4 Software-Architektur & -Design

Der Benutzer interagiert mit dem Frontend (genauer spezifiziert in Kapitel 3.4.1), welches seinerseits mit dem Backend kommuniziert. Das Backend führt auf Abruf eine lineare Regression, Crossvalidation und Temperaturmessungs-Abfrage für einen spezifiziertes Zeitintervall durch und stellt die berechneten Werte dem Frontend zur grafischen Aufbereitung zur Verfügung. Das Backend von ZykTemp wird als eine “resource-oriented architecture“ aufgebaut, die Schnittstellen sind entsprechend nach den RESTful-Prinzipien modelliert. Details zum Backend sind dem Kapitel 3.4.2 zu entnehmen.

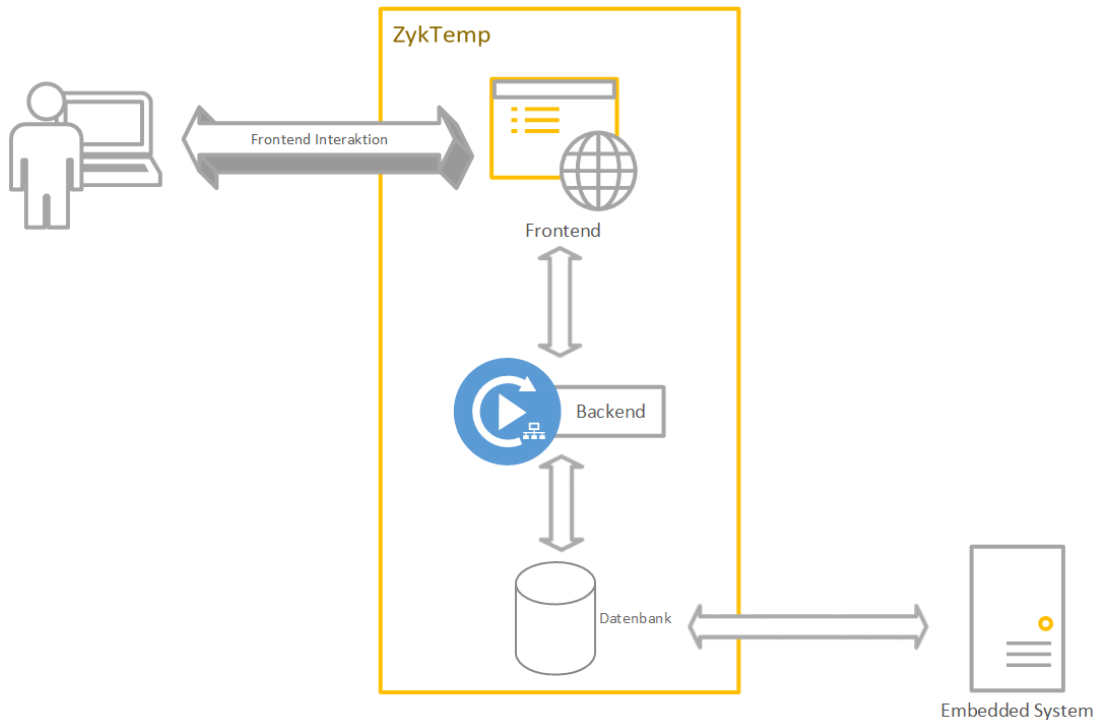


Abbildung 3.6: Übersichtsdiagramm ZyTemp (stark abstrahiert)

3.4.1 Frontend

Als "Frontend" wird die Browser-Applikation von ZyTemp bezeichnet, die vom Benutzer aufgerufen wird. Die Frontend-Applikation holt sich über die RESTful-HTTP-Schnittstelle des Backends benötigte Ressourcen, welche im Browser in Form einer Auswertung angezeigt werden.

3.4.2 Backend

Bei der als "Backend" bezeichneten Komponente handelt es sich um einen Java-WebService, der den RESTful-Service Design Prinzipien folgt. Der Service kümmert sich primär um die Persistierung und Aufbereitung von Daten.

Wie folgende Grafik zeigt, teilt sich das Backend in vier Schichten auf.

Die Controller Schicht dient als Schnittstelle zum Frontend. Zur Service-Schicht gehören Klassen, die Business-Logik abhandeln und Messdaten für Frontend aufbereiten.

Kommunikation

Bei Aufruf einer URL über das HTTP-Protokoll wird eine Resource⁷ in JSON-konformer Notation zurückgeliefert, wobei jeweils ein HTTP-Status mitgesendet wird, der den Status dieser Abfrage widerspiegelt.

⁷Als Resource können sämtliche Java-Klassen in Frage kommen, die zur Datenhaltung verwendet werden und wenig bis gar keine Logik beinhalten - so z.B. eine Klasse "Measurement"

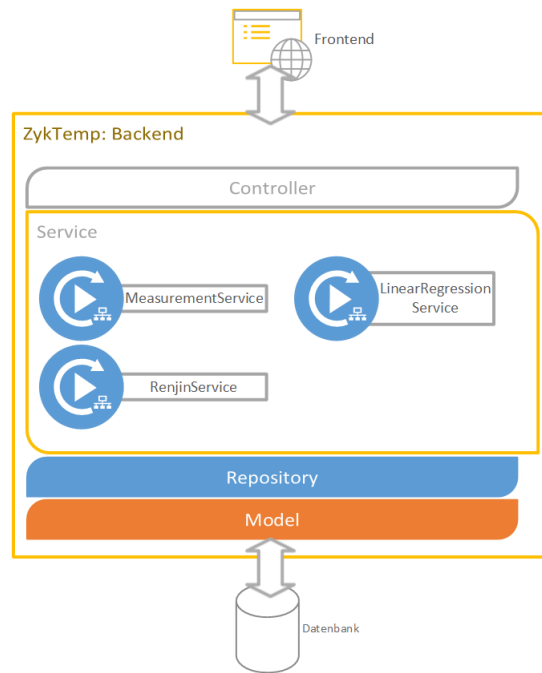


Abbildung 3.7: Übersichtsdigramm Backend (abstrahiert)

Struktur

Die Package-Struktur des Zyklus-Webservices orientiert sich an den in der Abbildung 3.7 dargestellten Schichten und setzt sich wie folgt zusammen:

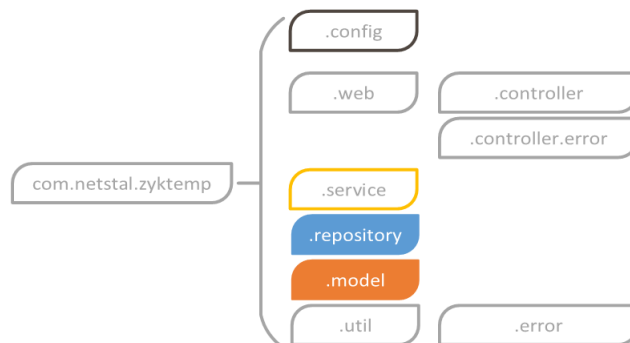


Abbildung 3.8: Packagestruktur Zyklus

com.netstal.zyktemp.web.controller

Das "controller"-Package enthält alle Klassen, die per HTTP-Request zugänglich sind.

Klassenstruktur

- `LinearRegressionController`
Aufrufe an Analyse-Funktionen (Lineare Regression, Crossvalidation, Prediction-Interval...)
- `MeasurementController`
Schnittstelle für einfache Daten-Abfragen
- `.error.ExceptionHandler`
Definiert, wie Exceptions abgehandelt werden

Schnittstellen: Die Controller-Klassen delegieren Business-Logik aufrufe an das Package “service“ und verwenden für den Datenaustausch Klassen aus dem “model“-Package.

Aufgabe: Controller-Klassen bilden eine Facade für das unterliegende System und delegieren Aufrufe nach einer Validierung von Input-Daten.

com.netstal.zyktemp.service

Service-Klassen sind für die Abhandlung von Business-Logik zuständig.

Klassenstruktur

- **LinearRegressionService**
Bereitet Daten auf, mit welchen im Anschluss Analysefunktionen durchgeführt werden
- **MeasurementService**
Kümmert sich um die Aufbereitung von Mess- und Zyklus-Daten (Synchronisierung, Gruppierung etc.)
- **RenjinService**
Schnittstelle zu Renjin; Handelt Skriptaufrufe und Datentransfers von der Skript-Engine zu Java ab

Schnittstellen: Zugriff auf Repository-Klassen für Verbindung zur Datenbank. Die RenjinService-Klasse greift zudem auf die Renjin-Skript-Engine zu.

Aufgabe: Zentrale Stelle für Abhandlung von Business-Logik.

com.netstal.zyktemp.repository

Data-Access-Objekte sind im “repository“-Package untergebracht.

*Klassenstruktur*⁸

- **CycleRepository**
Queries zur Abfrage und Gruppierung von Zyklen
- **MeasurementRepository**
Abfragen zu den in der Datenbank gespeicherten Temperaturmessungen

Schnittstellen: Die Repository-Klassen greifen auf Klassen aus dem “model-Package“ zu, die zur Datenhaltung verwendet werden.

Aufgabe: Die Repositories definieren die Schnittstelle zur Datenbank und werden für jedes Java-Objekt mit einem Mapping auf eine Tabelle erstellt. In den Repository-Klassen werden Queries gesammelt.

com.netstal.zyktemp.util

Utility-Klassen, die in kontextübergreifend Anwendungen finden.

- **.error.ZykTempException**
Exception-Klasse, die von einem Exception-Handler aufbereitet wird
- **.error.ErrorResponse**
JSON-Repräsentation eines Errors, wie er an das Frontend übergeben wird

⁸siehe Abbildung 3.9

- `.error.ErrorField`
Datenklasse für genauere Spezifikation von Fehlern

Schnittstellen: Keine

com.netstal.zyktemp.model

Im “model“-Package sind Klassen zur Datenhaltung und Transfers abgelegt.

Klassenstruktur

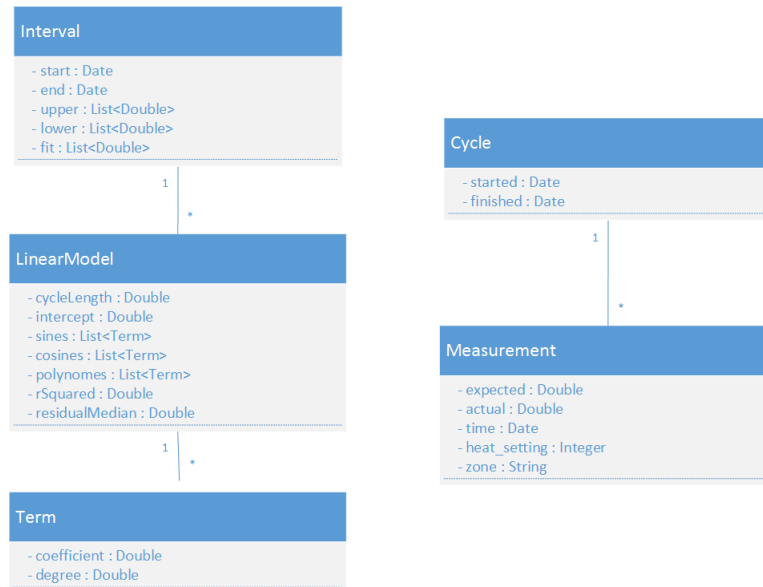


Abbildung 3.9: Datenklassen ZyktTemp

Schnittstellen: Keine

Aufgabe: Objekte die zur Datenhaltung verwendet werden.

com.netstal.zyktemp.config

Enthält Spring-Konfigurationsklassen

Klassenstruktur

- `ApplicationConfig`
Konfiguriert Environment, Instanziert Renjin-Engine und definiert Packages, die nach Spring-Annotations gescannt werden sollen
- `JpaConfig`
Baut Verbindung zur Datenbank auf und erstellt Repository- und Entity-Objekte
- `WebAppInitializer`
Initialisiert WebApplikation und spezifiziert Konfigurationsklassen
- `WebMvcConfig`
Sucht nach Controller-Klassen und definiert, wie Antworten für das Frontend transformiert werden sollen (Java-Objekte zu JSON oder möglicherweise andere Repräsentation)

Aufgabe: Bündelt Konfigurationsdateien, durch die der Webservice aufgebaut und gestartet

wird.

3.4.3 Datenbank

Die “Datenbank“ Komponente wird zur Haltung von Messdaten und Zykluszeiten verwendet. Diese Daten werden durch eine Import-Prozedur manuell importiert. Da die Datenbank-Implementation vom Webservice entkoppelt wurde⁹, besteht je nach verwendeter Technologie die Möglichkeit, Daten per HTTP oder einfache Imports per .csv-Files durchzuführen¹⁰.

3.4.4 Embedded System

Das Embedded-System verfolgt den Zweck, Sensordaten auszulesen und über eine vom Backend vorberechnete Funktion Ausreisser und Trends zur Laufzeit zu erkennen. Nachträglich sollen Messdaten an das Backend zurückgeliefert werden. Mit den kontinuierlich eingelieferten Daten soll die vorberechnete Funktion¹¹ aktuell gehalten werden, was durch einen erneuten Datenaustausch vom Backend zum Embedded System geschieht.

Durch eine Fokus-Anpassung der Studienarbeit wird diese Komponente nicht implementiert¹².

⁹Durch Auswechseln des Treibers kann die Datenbank ohne grossen Mehraufwand ausgetauscht werden

¹⁰siehe Kapitel 5.1.3

¹¹eine durch das entwickelte Verfahren generierte Funktion zur Vorhersage von Messwerten

¹²Siehe Kapitel 3.6.2

3.5 Analyse

3.5.1 Lineare Regression

Ziel der linearen Regression ist es, für ein Set von Daten eine Funktion zu finden, welche die Punkte im Mittel möglichst genau beschreibt. Für eine Funktion der Form $Y = aX + b$ werden die Parameter a (Steigung) und b (Achsenabschnitt) mithilfe dieser Daten so gewählt, dass der mittlere quadratische Fehler so klein wie möglich gehalten wird (vereinfachte Erklärung aus [2]).

Ansatz nichtlineare Regression

Die nichtlineare Regression besteht aus dem gleichen Verfahren wie die lineare Regression, welche bei Herr Dr. Andreas Müller im Fach "Wahrscheinlichkeit und Statistik" behandelt wurde. Die Funktion $Y = aX + b$ wird durch eine neue Funktion ersetzt. Der Algorithmus zur Entwicklung der Regression bleibt identisch.

Erklärung an einem einfachen Beispiel

mit Hilfe des Skripts von Andreas Müller [2]

Die Funktion

$$Y = aX + b$$

wird um einen Term erweitert

$$Y = aX^0 + bX^1 + cX^2$$

Der Fehler der Funktion ist definiert durch: $Q(a, b, c) = E((aX^0 + bX^1 + cX^2 - Y)^2)$, mit $E()$ = Erwartungswert

Ausgeschrieben ergibt dies:

$$\begin{aligned} & E(a^2 + 2abX + 2acX^2 - 2aY + b^2X^2 + 2bcX^3 - 2bYX + c^2X^4 - 2cYX^2 + Y^2) \\ & = \\ & a^2 + b^2E(X^2) + c^2E(X^4) + E(Y^2) + 2abE(X) + 2acE(X^2) - 2aE(Y) + 2bcE(X^3) - 2bE(XY) - 2cE(YX^2) \end{aligned}$$

Um den kleinstmöglichen Fehler zu ermitteln, muss die Funktion für alle Variablen abgeleitet und Null gleichgesetzt werden:

$$\begin{aligned} 0 &= \frac{\partial Q}{\partial a} = 2a + 2bE(X) + 2cE(X^2) - 2E(Y) \\ 0 &= \frac{\partial Q}{\partial b} = 2bE(X^2) + 2aE(X) + 2cE(X^3) - 2E(XY) \\ 0 &= \frac{\partial Q}{\partial c} = 2cE(X^4) + 2aE(X^2) + 2bE(X^3) - 2E(YX^2) \end{aligned}$$

Dies resultiert in folgendem Gleichungssystem:

$$\begin{cases} a + bE(X) + cE(X^2) = E(Y) \\ aE(X) + bE(X^2) + cE(X^3) = E(XY) \\ aE(X^2) + bE(X^3) + cE(X^4) = E(X^2Y) \end{cases}$$

Mit dem Gaußalgorithmus können nun die Parameter a, b und c bestimmt werden.

Anmerkung: Auffallend ist das Muster des Gleichungssystems. Nach rechts und unten kommen immer ein X mehr hinzu.

3.5.2 Overfitting

Werden mit der linearen Regression mehr Terme als vorhandene Daten verwendet, ergibt sich das Problem des “Overfittings“ oder “Overtrainings“. Das berechnete Polynom gibt dabei die Testdaten perfekt, also inklusive Ausreißern und Messfehlern, wieder. Ein overfittetes Model approximiert weitere Daten (nicht im Trainings-Set enthalten) mit weitaus geringerer Genauigkeit oder kann für diese gar unnütze Werte annehmen.

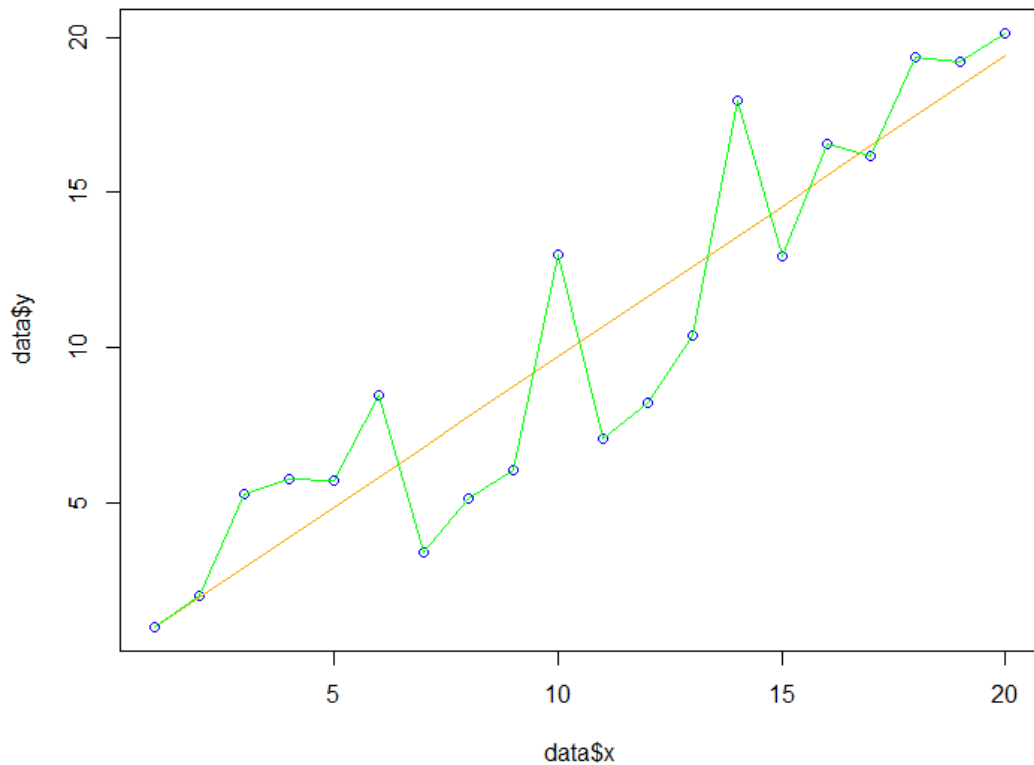


Abbildung 3.10: Grün: Kurve mit Overfit (folgt Noise);
Orange: Regression mit einem Freiheitsgrad

3.5.3 Crossvalidation

Auch wenn ein Overfit der Regression nicht die Ausmasse wie im Kapitel 3.5.2 annimmt, besteht durchaus die Möglichkeit, ein Modell über-trainiert zu haben.

Um feststellen zu können, ob das Problem im aktuell verwendeten Modell besteht, wird auf das Verfahren der “Crossvalidation“ zurückgegriffen. Für das Anwendungsszenario bieten sich besonders die “Leave-One-Out Crossvalidation“ und die “k-Fold Crossvalidation“ an.

k-Fold

Die “k-Fold Crossvalidation“ teilt ein Set von Messdaten in k Teile. Zur Veranschaulichung wird $k = 5$ gesetzt, in der Praxis wird für k meist 10 verwendet.

Unser Messdaten-Set wird in 5 gleich grosse Subsets aufgeteilt. Davon werden 4 Teile zur Erstellung eines Modells mittels linearer Regression verwendet. Das so erhaltene Modell wird mit dem letzten (noch nicht verwendeten) Teil der aufgesplitteten Daten validiert.

Diese Schritte werden insgesamt k (also 5) Mal durchgeführt, womit die Validierung für jede Kombination an Trainings- und Testdaten aus den aufgeteilten Daten einmal durchlaufen wurde.

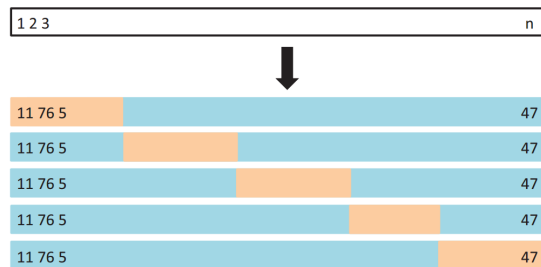


Abbildung 3.11: Datenpermutationen der 5-Fold-Crossvalidation Quelle: ISLR14[1]

Leave-One-Out

Die “Leave-One-Out Crossvalidation“ entspricht im Grunde der “k-Fold Crossvalidation“, bei der $k = \text{Anzahl Messdaten}$ entspricht.

Gegeben sei ein Messdaten-Set der Länge 100. In dem Fall werden insgesamt 100-Crossvalidations durchgeführt. In jeder Iteration werden 99 Messungen für das Trainieren des Modells und die ausgelassene Messung für die Validierung dieses verwendet.

Listing 3.1: Pseudocode zur Veranschaulichung der Leave-One-Out Crossvalidation

```
List<Integer> measurements; // beinhaltet saemtliche Messdaten
for (int i = 0; i < measurements.size(); i++) {
    // Erstelle kopie der Liste
    List<Integer> trainingData = new ArrayList<>(measurements);
    // Messung fuer Validierung aus Liste extrahieren
    Integer validationMeasurement = trainingData.remove(i);

    // Lin. Regr. fuer aktuelle Permutation der Daten durchfuehren
    LinearModel model = performLinearRegression(trainingData);
    // Modell mit ausgelassener Messung validieren
    validateModel(model, validationMeasurement);
}
```



Abbildung 3.12: Datenpermutationen der Leave-One-Out-Crossvalidation Quelle: ISLR14[1]

3.6 Umsetzung

3.6.1 Generierung synthetischer Daten

In der ersten Phase der Studienarbeit sind noch keine realen Maschinendatenabzüge für die Weiterverarbeitung vorhanden. Über diesen Zeitraum soll deshalb die Funktionstüchtigkeit der linearen Regression für Zyklus-periodische Signale mit globalen Trends ermittelt werden, wozu Messdaten mit realitätsnaher Charakteristik erforderlich sind.

Um das Generieren von Messdaten möglichst realistisch zu gestalten, muss die Messdaten-Funktion folgende Anforderungen erfüllen:

1. Periodisches Signal
2. Emulation von Messfehlern
3. Enthält Zyklus- / Periodenübergreifende Trends
4. Intensität der Komponenten konfigurierbar (Periodendauer, globale Trends, maximaler Messfehler)

Messdaten-Generator

Um die lineare Regression auch für komplexere Szenarien testen zu können, wird als Basis für das Synthetisieren der Messdaten eine Kombination aus 3 verschiedenen, periodischen Funktionen (zusammengesetzt aus \cos und \sin) mit jeweils unterschiedlichen Wellenlängen und Amplituden verwendet.

Für die Periodizität wurden 2 Sekunden angenommen. Die Netstal-Maschinen loggen im Optimalfall alle 670 Millisekunden eine Temperaturmessung, was zumeist in 3 Messungen pro Zyklus resultiert. Bei der Generierung der Messdaten kann das Messintervall über die *Sampling-Rate* konfiguriert werden.

Fehlerfunktion

Um in jeden Messpunkt einen zufälligen Fehler einzurechnen, wird ein Random-Number-Generator von R verwendet (Funktion *rnorm*) welche gaussverteilte Zufallszahlen zwischen plus und minus Eins ausgibt. Diese können mit einem Vorfaktor verstärkt werden.

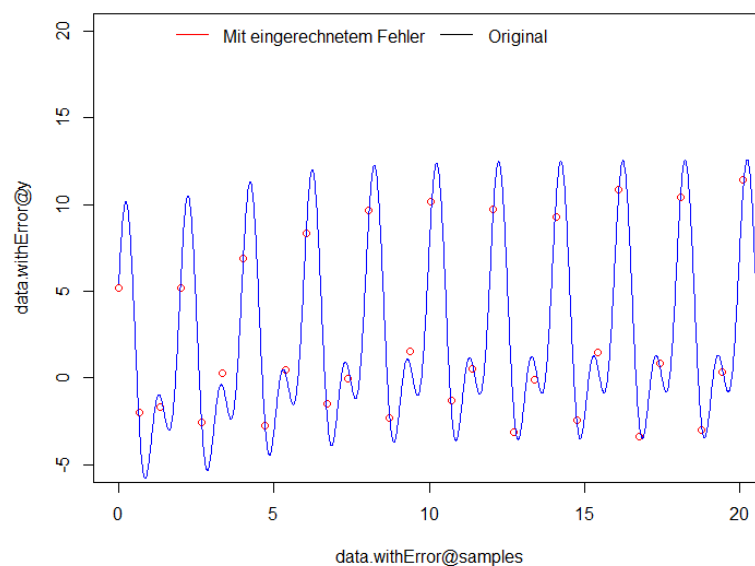


Abbildung 3.13: Messdaten-Funktion (blau), Samples mit eingerechnetem Fehler (rot)

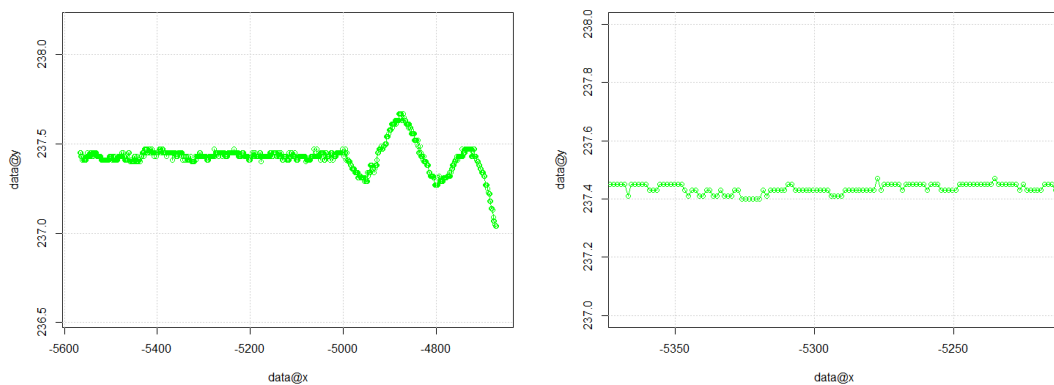
3.6.2 Analyse Realdaten

Mit Erhalt eines Realdaten-Abzugs einer Netstal-Maschine konnten genauere Aussagen über die Maschinen, deren Zonen und den Temperaturverläufen getroffen werden. Wie in Kapitel 3.1.1 beschrieben, sind den Zonen verschiedene Zwecke zuzuordnen und werden in diesem Kapitel danach gruppiert behandelt.

Plastifizierung

Umfasst Zonen 10, 13, 14, 15, 16, 17, 18

Messdaten aus Zonen der Plastifiziereinheit weisen in einfachen Plots weder Regelmässigkeiten noch andere Auffälligkeiten innerhalb eines Zyklus auf.



(a) Temperaturverlauf Zone 10

(b) Ausschnitt für Analyse

Abbildung 3.14: Temperaturverläufe Plastifizierung

Für die genauere Analyse der Zyklus-Charakteristik wurden die Daten auf eine Zykluslänge gruppiert, was jedoch auch keine weiteren Erkenntnisse lieferte.

In einem nächsten Ansatz wurde die von Netstal angesprochene zeitliche Varianz von Zyklen in die Berechnungen miteinbezogen. Aus allen Zyklen, die Messdaten in der ausgewählten Zone beinhalten, wird eine Median-Zykluslänge ermittelt.

Jeder Zyklus wird dann auf diese Median-Länge normiert und die Messzeitpunkte relativ zu der neuen Zykluslänge berechnet.

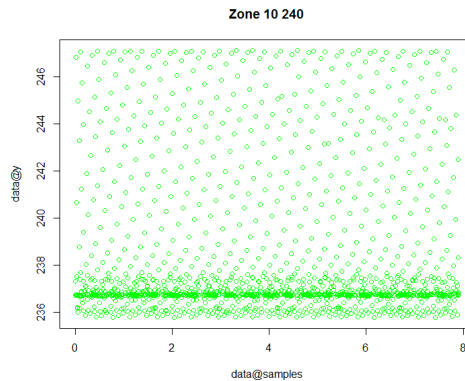


Abbildung 3.15: Zone 10 Zyklus-synchron bei Sollwert 240° Celsius

Weitere Erkenntnisse zur Plastifizierung Nach Besprechung der Befunde machte Betreuer Prof. Oliver Augenstein auf Zusammenhänge von Heizeinstellungen und Temperaturtrends aufmerksam. Die Intensität des Heizstellwerts ist mit einem gewissen zeitlichen Versatz in den Temperaturmessungen wiedererkennbar. Die Möglichkeiten, Daten mit diesem Ansatz zu analysieren und vorherzusagen, werden in Kapitel 3.9.1 erläutert, können jedoch im Rahmen der Studienarbeit nicht im Detail behandelt und umgesetzt werden.

Ebenfalls kann die Temperatur einer benachbarte Zone Einfluss auf Verläufe der beobachteten Zone haben. Verändert sich eine Zone um mehrere Grad Celsius, kann dies in der benachbarten Zone detektiert werden.

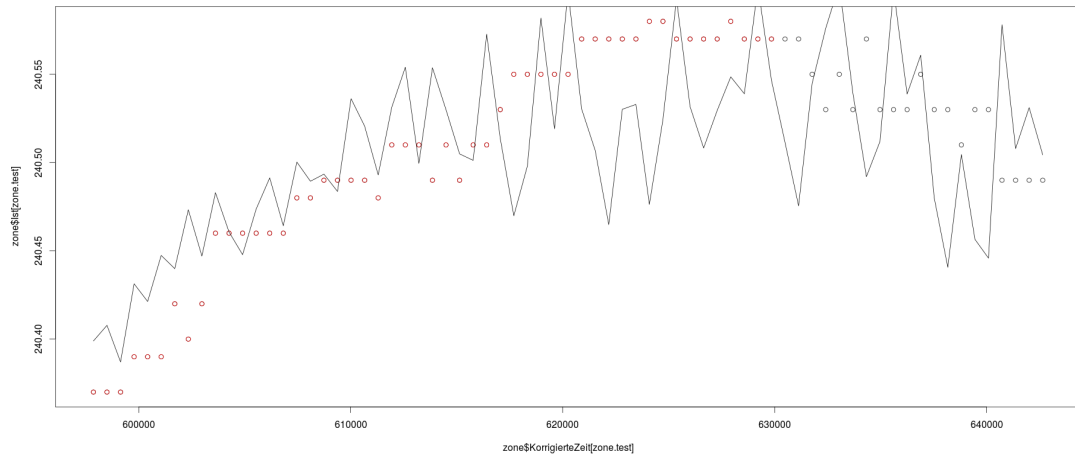


Abbildung 3.16: Vorhersage Temperaturverlauf unter Miteinbeziehen eines vorherigen Heizstellwert; Plot von Prof. Augenstein

Werkzeugzonen

Umfasst Zonen 401, 402, 403, 404

Temperaturschwankungen ergeben sich nicht durch das aktive Erhitzen, sondern finden ihre Ursache im Ausstoss des geschmolzenen Plastikgranulats in eine Gussform¹³.

Der Einfluss dieser Komponente ist für die Werkzeugzonen wesentlich grösser als der des Heizstellwertes¹⁴, weshalb dieser bei der Analyse vernachlässigt werden kann. Dementsprechend existieren keine relevanten Abhängigkeiten zu anderen Daten und es können mit den Auswertungsmethoden, welche bei der Plastifiziereinheit keinen Erfolg gebracht haben, die gewünschten Schwankungen innerhalb von Zyklen ermittelt werden konnten.

¹³Details zu den Werkzeugzonen sind in Kapitel 3.1.1 beschrieben

¹⁴Die Heizstellwerte sind meist automatisch reguliert, um die Materialtemperatur aufrecht zu erhalten

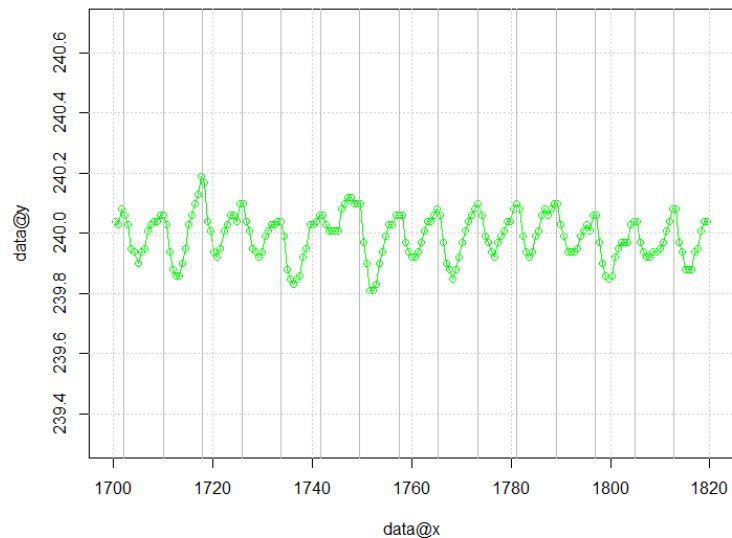


Abbildung 3.17: Ausschnitt Zone 401; Zyklus Start und Ende sind durch graue Linien veranschaulicht

Erkenntnisse und Auswirkungen

Nach Rücksprache mit Prof. Oliver Augenstein und Netstal AG wurde der Fokus der Studienarbeit leicht angepasst. Die Real-Time Analyse macht durch die eingeschränkten Auswertungsmöglichkeiten nur noch begrenzt Sinn.

Die zuvor als “Cloud-Applikation“ bezeichnete Komponente, die für das nachträgliche Auswerten von Daten gebraucht werden sollte, hat deswegen an Signifikanz gewonnen und soll durch Analyse-Möglichkeiten und Benchmarks erweitert werden¹⁵.

3.6.3 Lineare Regression mit Fourier-Komponenten

Wie in Kapitel 3.6.1 angenommen, enthalten die realen Temperaturverläufe folgende Charakteristiken:

- Globale Trends
- Innerzyklische Schwankungen

Saisonales Verhalten

Das Verhalten innerhalb eines Zyklus wiederholt sich in auswertbaren Zonen nach jeweils einer Zykluslänge. Um mit der linearen Regression eine periodische Funktion zu erstellen, wird mit Fourier-Komponenten¹⁶ gearbeitet. Dabei werden Sinus- und Kosinus-Funktionen mit einer Frequenz, die einem Vielfachen der Grundfrequenz entspricht, verwendet. Die Basisfrequenz wird auf

$$F = 1 / \text{Median-Zykluslänge}$$

gesetzt. Da Sinus und Kosinus $2\text{-}\pi$ periodische Funktionen sind, entspricht die Kreisfrequenz

¹⁵Issues: SAZ 29-37 und SAZ 41-44

¹⁶Begriff ist zurückzuführen auf Jean Baptiste Fourier’s Fourier-Transformation

$$\omega = 2 * \pi * F$$

Aus den Komponenten wird durch die lineare Regression eine neue Funktion erstellt, welche dann das Verhalten eines Zyklus wiedergibt und sich nach jeweils einer Zykluslänge (=Grundfrequenz) wiederholt.

Globales Verhalten

Da es für Netstal besonders interessant ist, Temperaturtrends zu ermitteln, werden weitere Terme, deren Wellenlängen länger als ein Zyklus sind, in das lineare Modell miteinbezogen. Dazu werden ebenfalls Sinus und Kosinus Komponenten, wie auch polynomielle Terme verwendet.

Mit den Polynomen sollen konstante Trends abgebildet werden, während die Sinus- und Kosinus-Komponenten mit zyklusübergreifenden Frequenzen für wiederholende Schwankungen¹⁷ zum Einsatz kommen.

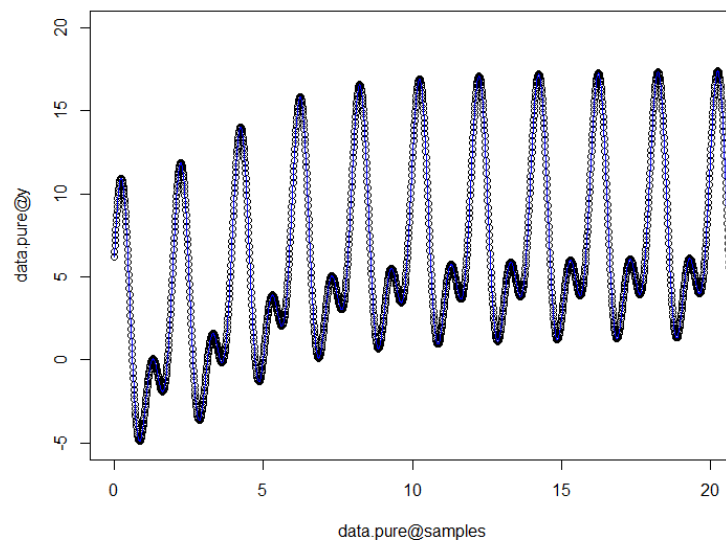


Abbildung 3.18: Beispiel von globalen Trends mit wiedererkennbarem saisonalem Verhalten

In der Abbildung 3.18 ist das zuvor erklärte saisonalen Verhalten in den sich wiederholenden Ausschlägen erkennbar. Dass die einzelnen Ausschläge sich nicht auf genau derselben Höhe befinden ist durch die globalen Trends der Messdaten bedingt.

¹⁷Voreingestellt sind Schwankungen mit einer Länge von 10 - 13 Zyklen

Vorgehen

Während der Durchführung der linearen Regression ist noch nichts über die Charakteristik des aktuellen Datenausschnitts bekannt, weshalb die Regression grundsätzlich mit zu vielen Termen durchgeführt wird. Über den $P(>|t|)$ -Wert kann die Signifikanz einer Komponente ermittelt werden; je mehr sich der Wert an 0 annähert, desto relevanter die Komponente für eine korrekte Nachbildung des Temperaturverlaufs.

Estimate	Std.	Error	t value	Pr(> t)	
(Intercept)	891.67	225.56	3.953	0	***
cos(x * 0.78539816339 * 1)	0.055	0.006	9.466	0	***
cos(x * 0.78539816339 * 2)	0.018	0.006	2.94	0.004	**
cos(x * 0.78539816339 * 3)	0.001	0.014	0.099	0.921	
cos(x * 0.78539816339 * 4)	-0	0.004	-0.063	0.95	
cos(x * 0.78539816339 * 5)	0.003	0.014	0.203	0.839	
sin(x * 0.78539816339 * 1)	-0.064	0.006	-10.617	<0	***
sin(x * 0.78539816339 * 2)	0	0.006	0.079	0.937	
sin(x * 0.78539816339 * 3)	-0.01	0.019	-0.514	0.609	
sin(x * 0.78539816339 * 4)	-0.013	0.027	-0.48	0.633	
sin(x * 0.78539816339 * 5)	-0.004	0.018	-0.215	0.83	
I(x^1)	19.74	9.119	2.165	0.033	*
I(x^2)	-0.807	0.318	-2.533	0.013	*
I(x^3)	0.005	0.002	2.631	0.01	*
cos(x * 0.78539816339 * 0.1)	-1616.497	588.82	-2.745	0.007	**
cos(x * 0.78539816339 * 0.09090)	7943.626	3157.874	2.515	0.014	*
cos(x * 0.78539816339 * 0.08333)	-10512.522	4619.518	-2.276	0.025	*
cos(x * 0.78539816339 * 0.07692)	3533.714	1852.57	1.907	0.06	.
sin(x * 0.78539816339 * 0.1)	296.32	129.66	2.285	0.025	*
sin(x * 0.78539816339 * 0.09090)	-5351.733	1593.378	-3.359	0.001	**
sin(x * 0.78539816339 * 0.08333)	14272.475	4682.986	3.048	0.003	**
sin(x * 0.78539816339 * 0.07692)	-9849.044	3429.078	-2.872	0.005	**

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.042 on 85 degrees of freedom
 Multiple R-squared: 0.7627, Adjusted R-squared: 0.7041
 F-statistic: 13.0115 on 21 and 85 DF, p-value: < 0

Das lineare Modell wird nun angepasst, indem alle Komponenten, deren $P(>|t|)$ -Wert den Signifikanz-Schwellwert¹⁸ überschreiten, weggelassen werden. Mit dem verbleibenden Termen wird erneut eine lineare Regression durchgeführt. Aus Performancegründen werden diese Schritte nur ein Mal durchlaufen.

Es besteht die Möglichkeit, die Anzahl Wiederholungen hinaufzusetzen. In diesem Fall wird der Algorithmus n Mal durchlaufen und stoppt alternativ, falls keine Komponente mehr als irrelevant betrachtet wird.

Zu beachten ist, dass Sinus- und Kosinus-Terme immer in Paaren übernommen werden, d.h. auch wenn die Sinus-Komponente als insignifikant gilt, wird sie trotzdem in das angepasste Modell übernommen, falls die Kosinus-Komponente mit derselben Wellenlänge als relevant erachtet wird.

¹⁸Konfigurierbarer Parameter, Standardwert ist 0.1, bei uns: 0.01

Aus dem vorherigen Summary resultiert das im folgenden Modell:

Estimate	Std.	Error	t value	Pr(> t)	
(Intercept)	880.026	218.248	4.032	0	***
I(x^1)	19.188	8.836	2.172	0.032	*
I(x^2)	-0.789	0.308	-2.557	0.012	*
I(x^3)	0.005	0.002	2.661	0.009	**
cos(x * 0.78539816339 * 1)	0.055	0.006	9.79	0	***
sin(x * 0.78539816339 * 1)	-0.064	0.006	-10.931	<0	***
cos(x * 0.78539816339 * 2)	0.018	0.006	3.141	0.002	**
sin(x * 0.78539816339 * 2)	0.001	0.006	0.164	0.87	
cos(x * 0.78539816339 * 0.1)	-1581.662	570.08	-2.774	0.007	**
sin(x * 0.78539816339 * 0.1)	301.049	125.26	2.403	0.018	*
cos(x * 0.78539816339 * 0.09090)	7748.988	3058.562	2.534	0.013	*
sin(x * 0.78539816339 * 0.09090)	-5291.055	1539.476	-3.437	0.001	***
cos(x * 0.78539816339 * 0.08333)	-10217.367	4475.84	-2.283	0.025	*
sin(x * 0.78539816339 * 0.08333)	14039.869	4529.645	3.1	0.003	**
cos(x * 0.78539816339 * 0.07692)	3410.005	1795.719	1.899	0.061	.
sin(x * 0.78539816339 * 0.07692)	-9665.782	3318.321	-2.913	0.005	**

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.041 on 91 degrees of freedom
 Multiple R-squared: 0.7589, Adjusted R-squared: 0.7191
 F-statistic: 19.0940 on 15 and 91 DF, p-value: < 0

Die Vorhersage-Genauigkeit des optimierten Modells kann über das ANOVA-Kommando in *R* mit der des ursprünglichen Modells verglichen werden.

3.7 Datenanalyse Tools

Genauere Erläuterungen über die mathematischen Vorgänge im Webtool.

3.7.1 Messdatenauswertung

Messlinie

Die Messlinie ist eine Verbindungslinie zwischen den Messpunkten. Jedoch ist es keine sture Gerade, sondern ein kubisch hermetisches “Spline“ des Typs cardinal.

Lineare Regression

Die lineare Regression in der Messdatenauswertung besteht aus allen statistisch relevanten Termen der Regression mit den dazugehörigen Koeffizienten. Genauere Informationen wie die lineare Regression entsteht, ist unter dem Kapitel 3.6.3 beschrieben.

Ergebniskurve

Von jedem Messwert(x_{MP}, y_{MP}) wird dessen innerzyklischer Temperaturwert(y_{Offset}) berechnet und subtrahiert ($y_{MP} - y_{Offset}$). Der innerzyklische Temperaturwert entsteht aus einer Funktion der lokalen Regressionsterme, angewendet auf die x-Koordinate relativ zum Zyklusstart ($x_{MP} - x_{Zyklusstart}$).

Innerzyklische Schwankungen

Die Kurve entsteht durch die lokalen Regressionsterme (d.h. $\sin(i*x*T)$ und $\cos(i*x*T)$, mit $i>0$ und T als $2*\pi/Periodendauer$) angewendet zu einer x-Koordinate relativ zum Zyklusstart. Da sich diese Kurve um den Nullpunkt bewegen würde, wird der eingestellte Temperaturwert der Zone dazu addiert. Da sich dieser verändern kann, wird die eingestellte Temperatur durch den Sollwert des ersten Messpunktes repräsentiert.

3.7.2 Resultate

Ergebniskurve

siehe 3.7.1 Ergebniskurve

Globale Terme

Die lineare Regressionskurve beschränkt auf die globalen Terme. Die globalen Terme sind zyklusübergreifende Fourier- und polynomielle Terme. Dadurch wird der grobe Temperaturverlauf nachgebildet. **Vorsicht:** Die Globalen Terme müssen bei der linearen Regression angegeben werden und unterstehen einem labilen Feintuning. Will man Temperaturschwankungen einer Zeitspanne von 10 Zyklen erfassen, muss dieser Sinus und Kosinus Term im *R*-Skript definiert werden. Dasselbe gilt für Schwankungen anderer Längen.

Werden zu viele globale Terme definiert, läuft man Gefahr, ein Overfitting der linearen Regression zu bekommen, wie in 3.5.2 beschrieben. Aus diesem Grund muss sehr genau abgeschätzt und ausprobiert werden, welche globalen Fourier-Terme man abdecken möchte. In diesem Webtool werden die Terme von 10-13 zyklusübergreifenden Temperaturschwankungen benutzt. Zusammen mit den Polynomialtermen können die meisten (auch sehr grosse) Schwankungen nachgebildet werden.

Moving Average

Bereits ausführlich im Benutzerhandbuch unter 4.5.4 beschrieben.

3.7.3 Weiterführende Informationen

Intervall: Zyklische Schwankung der Messpunkte

Um die zyklische Temperaturschwankung als Offset zu bekommen, wird dem Messwert der dazugehörige Wert aus den globalen Termen abgezogen¹⁹. Damit alle Werte sich innerhalb eines Zyklus befinden, wird auf der Abszisse²⁰ die Zyklus-Startzeit vom Zeitstempel subtrahiert.

Benchmarks: Abweichung vom Sollwert

Dieser Wert entspricht der durchschnittlichen Abweichung des Temperaturmesswertes vom eingestellten Sollwert und wird wie folgt berechnet:

$$\frac{1}{n} * \sqrt{\sum_0^n (Sollwert_i - Messwert_i)^2}$$

Benchmarks: 10-Fold

Das 10-Fold Crossvalidation Resultat gibt Auskunft darüber, wie gut das lineare Modell Daten vorhersagen kann, die nicht im Trainingsset liegen (relevant vorallem für eine Real-Time Analyse). Der Wert gibt die durchschnittliche Abweichung des vorhergesagten Wertes zum tatsächlichen Wert an.

Der Algorithmus ist im Kapitel 3.5.3 genauer beschrieben.

Benchmarks: R^2

Indikator, wie gut der Messwert durch die verwendeten Komponenten der linearen Regression approximiert werden kann. Nimmt einen Wert zwischen 0 und 1 an, wobei bei 0 kein Zusammenhang zwischen den Komponenten und dem Modell existiert, und 1 einer perfekter Repräsentation entspricht.

Benchmarks: Median Residual

Der Median einer Auflistung von Zahlenwerten ist der Wert, der an mittlerer (zentraler) Stelle steht, wenn die Werte der Größe nach sortiert werden. Quelle: Wikipedia[3]

In diesem Fall ist es der mittlere Wert aller Differenzen von Messwerten zu deren Vorhersage (durch die lineare Regression). Falls sich dieser Wert nicht um den Nullpunkt befinden sollte, ist es ein Indiz dafür, dass die lineare Regression einen Offset zu der tatsächlichen Temperaturkurve hat.

¹⁹Wert der "globale Terme"-Kurve mit demselben x-Wert

²⁰x-Achse

3.8 Ergebnisdiskussion

In diesem Kapitel werden Stärken und Schwächen von aus der Studienarbeit resultierenden Befunde analysiert, die für ein weiterführendes Projekt im selben Bereich ausschlaggebend sein können.

3.8.1 Entwickeltes Verfahren

Die entwickelten Analysetools bilden Temperaturverläufe durch folgende Schritte nach:

1. Normierung Zykluslänge auf Median-Wert
2. Synchronisierung Messzeiten auf Median-Zykluslängen
3. Lineare Regression mit Sinus-, Kosinus- und Polynom-Termen

Durch starke Temperatursprünge im Maschinendatenabzug²¹, liefert das Verfahren nur für ausgewählte Teilbereiche, in denen Messungen mit einheitlichem Sollwert gemessen wurden, zufriedenstellende Ergebnisse in Bezug auf Verlaufsrekonstruktion und Vorhersagbarkeit.

Für Netstal AG wäre es interessant, die Applikation mit einem Datenabzug eines produktiven Systems zu verwenden. Liefern die Analysen eines stabil laufendem System einheitliche Resultate für mehrere Teilbereiche, so könnte das lineare Modell zur Vorhersage von Temperaturmessungen in einem Real-Time-System eingesetzt werden. Durch die starke Abhängigkeit der Temperaturen vom Heizstellwert in der Plastifizierung beschränkt sich dieser Anwendungsbereich aber auf die Werkzeugzonen.

Widerspiegeln die während der Studienarbeit verwendeten Maschinendaten aber die realen Umstände und Schwankungen einer Maschine, so ist das Verfahren nicht für den Einsatz in einem Real-Time System tauglich.

3.8.2 Fazit Technologiewahl

Renjin

Im Verlauf der Studienarbeit traten verschiedene Komplikationen bei der Verwendung des Frameworks “Renjin“²² auf. Durch unvollständige Implementation diverser R-Funktionen (Cross-validation, step()-Funktion) musste zusätzlicher Aufwand in deren Implementation investiert werden. Zudem wurden Performance-Unterschiede zwischen der nativen Implementation von R und der JVM-Nachbildung Renjin bemerkt. Genaue Messungen für zwei Kern-Funktionen werden im Folgenden aufgelistet:

	10-Fold	Linear Model step()
R (native)	0.08s	0.02s
Renjin	1s	4-5s

Die Tests wurden mit Temperaturmessungen aus Zone 401 die zwischen “2015-10-30 13:28:33.664“ und “2015-10-30 13:30:32.704“ gemessen, und umfassten insgesamt 745 Datenpunkte.

Während der unvollständige Implementationsstatus von R-Funktionen und die Performancevergleiche deutlich gegen Renjin sprechen, ist die komfortable Parameter-Übergabe und Resultat-Konvertierung von Renjin zu Java und umgekehrt ein Argument für die Verwendung dieses Frameworks.

Um eine Entscheidung zu treffen, ob R über JNI²³ oder Renjin verwendet werden soll, müssen das Zeitbudget für Implementation von Features gegen Performance-Ansprüche abgewogen werden. Alternativen dazu wäre der Einsatz einer anderen Programmiersprache für die Webanalyse-Tools.

²¹Verursacht durch Änderungen an Einstellungen während dem Aufzeichnen

²²R-Implementation in Java

²³Java Native Interface; Erlaubt Aufrufe von Systembefehlen und so eine native Ausführung von R-Skripten

Webapplikation

Die Frameworks, die für die Webapplikation verwendet wurden, erfüllten ihren Zweck ausreichend. Es wurden weder bei den Javascript-Frameworks (Frontend) noch bei den Java- und Spring-Frameworks (Backend) Performance-Einbussen gegenüber anderen gängigen Technologien in den jeweiligen Einsatzbereichen festgestellt.

3.9 Ausblick

Wegen des auf die Werkzeugzonen eingeschränkten Einsatzbereichs des entwickelten Verfahrens werden in diesem Kapitel alternative Ansätze für eine Real-Time Analyse und Vorhersage von Temperaturmessungen beschrieben.

3.9.1 Kalman-Filter

Wie Analysen von Prof. Oliver Augenstein ergeben haben (genauer beschrieben in Kapitel 3.6.2), haben Heizstellwerte einen starken Einfluss auf Temperaturverläufe der Plastifiziereinheitszonen. Das als “Kalman-Filter“ bezeichnete Verfahren bietet eine Möglichkeit zur Vorhersage von Werten unter Berücksichtigung und Gewichtung von anderen, unkorrelierten Variablen²⁴. Der Kalman-Filter hat den Vorteil, dass zusätzliche Informationen aus der Vergangenheit zu den aktuellen Messwerten miteinbezogen werden können.

²⁴meistens physikalische Vorgänge

Kapitel 4

Benutzerhandbuch

Die Verwendung des entwickelten Web-Tools erfordert genauere Kenntnisse über die Funktionsweise und Möglichkeiten, welche es bietet. Im Benutzerhandbuch sind diese genau aufgelistet und beschrieben. Um genauere Informationen über das mathematische Verfahren einzelner Funktionen zu bekommen, kann unter 3.7 nachgeschaut werden.

4.1 Übersicht

Die Webapplikation ist in vier Bereiche unterteilt, welche verschiedenen Zwecken dienen.

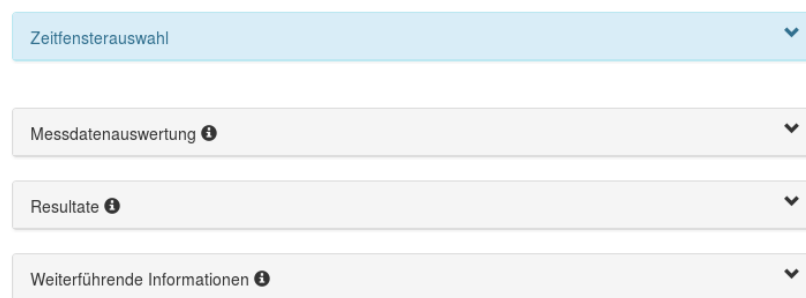


Abbildung 4.1: Illustration der Bereiche im Webtool

- **4.3 Zeitensterauswahl**
Bestimmen des Start- und Endzeitpunkt zwischen denen die Messdaten geladen werden.
- **4.4 Messdaten Auswertung**
Anzeige aller Messdaten inklusive den zugehörigen Auswertungen. Diese beinhalten: Lineare Regression, Prediction Intervall, Zyklus-Starts, Ergebniskurve sowie eine Ergebnis-Hilfskurve.
- **4.5 Resultate**
Auswertung mit den entwickelten Verfahren, die den Temperaturverlauf ohne Zyklus-Temperaturschwankungen darstellen.
- **4.6 Weiterführende Informationen**
Informationen über die Güte der linearen Regression und Verbesserung der Temperaturkurve durch Anwendung der beiden Verfahren.

4.2 Vorgehen

4.2.1 Messung laden

1. Zeigen Sie den Bereich "Zeitfensterauswahl" an, falls dieser nicht bereits geöffnet ist.
2. Wählen Sie die gewünschte Zone.
3. Legen Sie Startdatum inkl. Uhrzeit fest und wählen Sie eine Zeitspanne. Durch die "-2h" und "+2h" Schaltflächen kann der Auswertungsbereich direkt um 2h in die Vergangenheit/Zukunft geschoben werden.
4. Setzen Sie den Schalter "Start- und Endzyklus aus Timeline nehmen?" auf "ja".
5. Schieben Sie den Regler auf den gewünschten Start- und Endzyklus; das ausgewählte Fenster wird in der Timeline dargestellt. Start- und Endwert des Schiebereglers dürfen beim Auswahlvorgang auch gekreuzt werden.
6. Betätigen Sie die Schaltfläche "Messungen Laden".
7. Analysefunktionen können in den angezeigten Plots durch die verschiedenen Schalter ein- und ausgeblendet werden.

Alternative 1: Timeline Auswahl durch klicken

Alt.1-5 Klicken Sie auf den gewünschten Start- und Endzykluspunkt in der Timeline. Das festgelegte Fenster, dargestellt durch rote Balken, verschiebt sich an die gewünschte Position. Info: Dies funktioniert unabhängig von der Position des Schalters "Start- und Endzyklus aus Timeline nehmen?".

Alternative 2: Auswahl durch Datums- und Uhrzeitangabe

Alt.2-4 Setzen Sie den Schalter "Start- und Endzyklus aus Timeline nehmen?" auf "nein". Zwei Datumseingabefenster erscheinen.

Alt.2-5 Klicken Sie unter Startdatum auf die Schaltfläche mit dem Kalendersymbol und wählen Sie das gewünschte Datum inklusive Uhrzeit aus. Wiederholen Sie dies für das Enddatum.

4.3 Zeitfensterauswahl

In der Zeitfensterauswahl wird die Zeitspanne, in welcher Messdaten geladen werden, definiert. **Achtung! Es werden nur Messungen berücksichtigt, die einem Zyklus angehören!**

4.3.1 Übersicht



Abbildung 4.2: Zeitfensterauswahl von Timeline

4.3.2 Zonenauswahl

Die Auswahl enthält alle Zonen, die zu einer Messung (egal zu welchem Zeitpunkt) existieren. Nach einer Zonenselektion beziehen sich alle nachfolgend geladenen Daten auf diese. Die Zonen werden beim Laden der Webapplikation neu bezogen. Werden neue Messwerte samt neuer Zonen in die Datenbank geladen, muss die Seite aktualisiert werden, um die Änderungen sichtbar zu machen.

4.3.3 Timeline

Die Timeline dient als Hilfe zur Zeitfensterauswahl. Es wird der Temperaturverlauf über mehrere Zyklen dargestellt. Die Temperatur ist dabei immer ein Mittelwert eines Zyklus. Messungen ohne Zyklen fehlen ganz¹.

Die Anzeige der Timeline kann verschoben werden, ohne dass die aktuelle Messdatenauswertung beeinflusst wird. Um ein Timeline-Zeitfenster auszuwählen wird als erstes unter "Start" das Startdatum und die Startzeit gewählt, danach kann die Zeitspanne spezifiziert werden. Die Zeitspanne ist auf 5h begrenzt, da es bei längeren Intervallen zu erheblichen Verzögerungen beim Laden der Timeline-Daten kommen kann². Um ein schnelleres Laden von nachfolgenden Daten anzubieten, stehen die 2 Schaltflächen "-2h" und "+2h" zur Verfügung. Beim Betätigen wird das Zeitfenster um 2h verschoben.

Falls der Schalter "Start- und Endzyklus aus Timeline nehmen?" auf "ja" steht, sind zwei rote vertikale Balken in der Timeline zu sehen. Sie markieren den Start- und Endzyklus, der geladen wird, sobald man den "Messung laden" Button betätigt.

Der Schieberegler unterhalb der Timeline ist an diese roten Balken gekoppelt. Er markiert ebenfalls Start- und Endzyklus. Er wird nur angezeigt, falls der "Start- und Endzyklus aus Timeline nehmen?" Schalter auf "ja" steht, kann anders als die roten Balken jedoch verschoben

¹Dadurch kann ein falsches Bild entstehen. Zwei Punkte in der Timeline können im Extremfall auch mehrere Stunden auseinanderliegen und haben trotzdem den gleichen Abstand wie zwei kurz aufeinanderfolgende Zyklen.

²Diese Probleme sind durch die Aufbereitung einer grossen Datenmenge bedingt

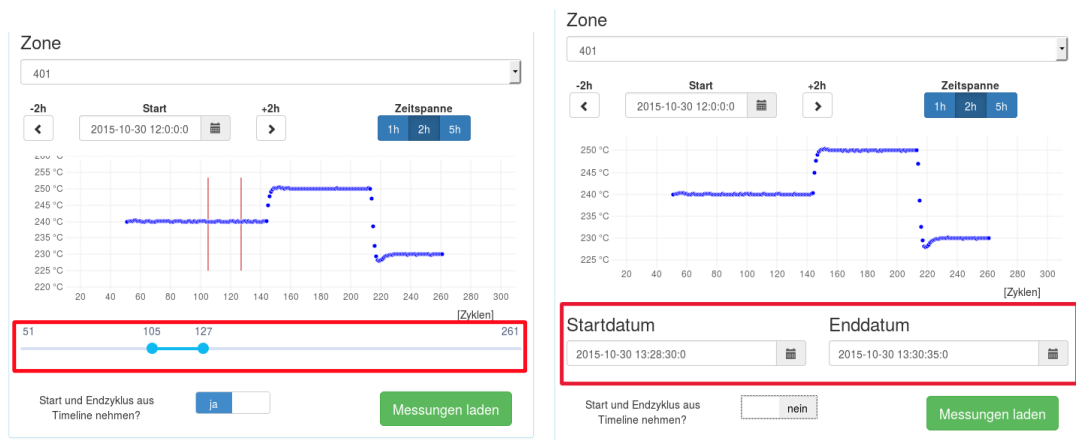
werden. Überkreuzen von Start- und Endzyklus spielt verursacht keine Fehler, da der kleinere Wert immer als Startwert verwendet wird. Bei gleichem Start- und Endzyklus wird der Endzyklus um eins erhöht.

4.3.4 Datepicker

Der Datepicker dient der genauen Definition von Start- und Endzeit. Um die Auswahl anzuzeigen, muss der “Start- und Endzyklus aus Timeline nehmen?” “Schalter auf “nein“ stehen. Wird nun “Messung laden“ betätigt, werden alle Messungen, welche einem Zyklus angehören, der zwischen diesen Daten liegt, geladen. Die Datums- und Zeitauswahl ist intuitiv bedienbar.

4.3.5 Vergleich von Timeline zu Datepicker

Im Unterschied zur Timeline fehlt bei der Auswahl durch genaue Datumsangabe der Schieberegler, dafür erscheinen zwei Datumfelder.



(a) Auswahl mit Timeline

(b) Auswahl durch Datum- und Uhrzeitangabe

Abbildung 4.3: Vergleich der beiden Möglichkeiten zur Zeitauswahl

4.4 Messdaten Auswertung

Mit der Messdatenauswertung kann die selektierte Messreihe visuell analysiert werden. Ebenso kann die dazugehörige lineare Regression und eine Ergebniskurve dargestellt werden. Genauere Informationen zur Umsetzung einzelner Einstellungen finden Sie in Kapitel 3.7.

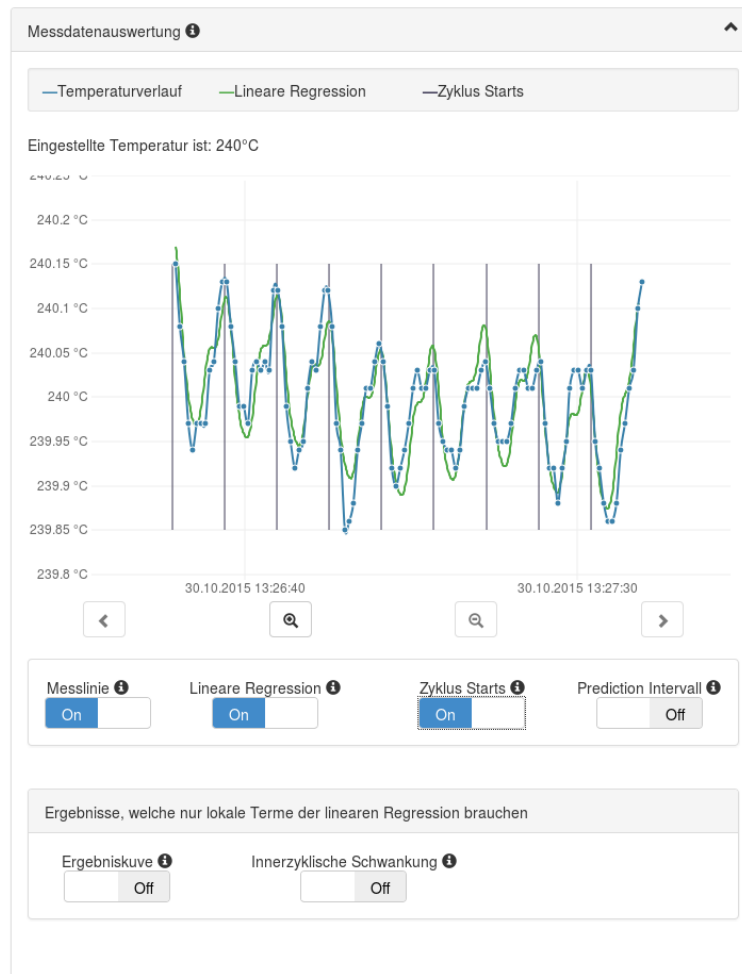


Abbildung 4.4: Übersicht der Messdatenauswertung

4.4.1 Plot

Der Plot ist das Kernelement der Sektion “Messdatenauswertung“ und zeigt die geladenen Messpunkte an. Je nach Auswahl werden weitere Analysefunktionen hinzugefügt oder entfernt.

Zoom

Der Zoom ist intuitiv bedienbar. Die Buttons mit den Lupen dienen zum Hinein- beziehungsweise Heraus-Zoomen. Die Pfeil-Buttons werden für die Navigation im Datenausschnitt verwendet. Der Y-Achsenausschnitt wird durch das Zooming nicht verändert.

4.4.2 Messlinie

Verbindungsline zwischen den Messpunkten. Besonders hilfreich in Kombination mit der Kurve der linearen Regression.

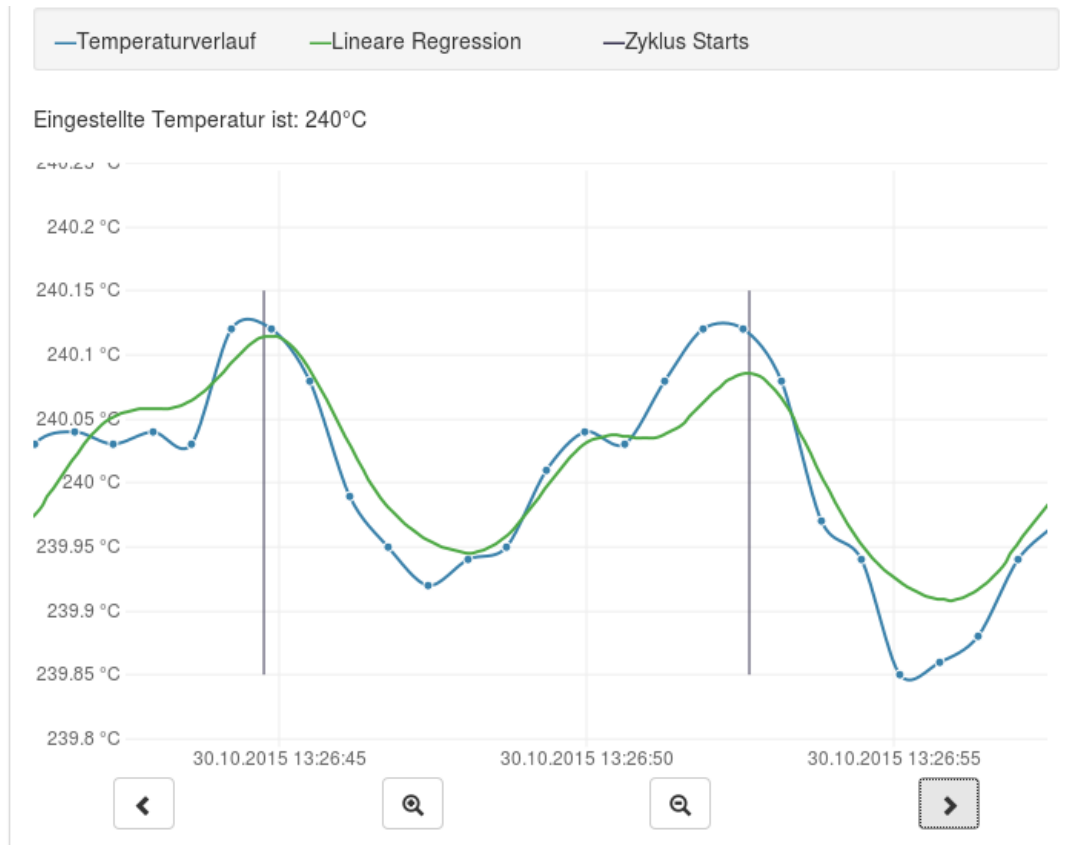


Abbildung 4.5: Hineingezoomte Ansicht der Messdaten (inklusive linearer Regression)

4.4.3 Lineare Regression

Die Regressionskurve sollte möglichst nahe an der Messlinie liegen. Sie ist ein fundamentaler Baustein der Analyse. Um besser über die Nützlichkeit der Regression urteilen zu können, stehen die “Benchmarks“ unter “Weiterführende Informationen“ zur Verfügung³.

4.4.4 Zyklus-Starts

Vertikale Linie die den Start einen Zyklus markiert. Auf das Zyklus-Ende wurde bewusst verzichtet, da dieser im Normalfall mit dem Start des nächsten Zyklus konvergiert.

4.4.5 Prediction Intervall

Hilft bei der Entscheidung, ob die statistische Vorhersage (lineare Regression) gut ist. 95% aller Messwerte sollten innerhalb des Prediction Intervalls liegen.

- Falls augenscheinlich viele Punkte ausserhalb des Intervalls liegen, ist das ein Indikator, dass die lineare Regression nicht stimmt.
- Falls das Intervall sehr gross ist, ist das ein Zeichen für eine schlechte Vorhersagbarkeit der Messwerte durch linearen Regression. Es bedeutet nicht, dass die lineare Regression falsch ist, eher das die Ungewissheit gross ist!

³siehe 4.6.3

4.4.6 Ergebniskurve

Die Ergebniskurve stellt den Temperaturverlauf ohne zyklische Temperaturschwankungen dar. Dazu wird vom Messwert die innerzyklische Temperatur abgezogen, was im Reiter "Weiterführende Informationen" betrachtet werden. Um einen direkten Vergleich dieser Charakteristik mit dem Temperaturverlauf zu ermöglichen, wurde der Schalter "Innerzyklische Schwankungen" eingeführt.

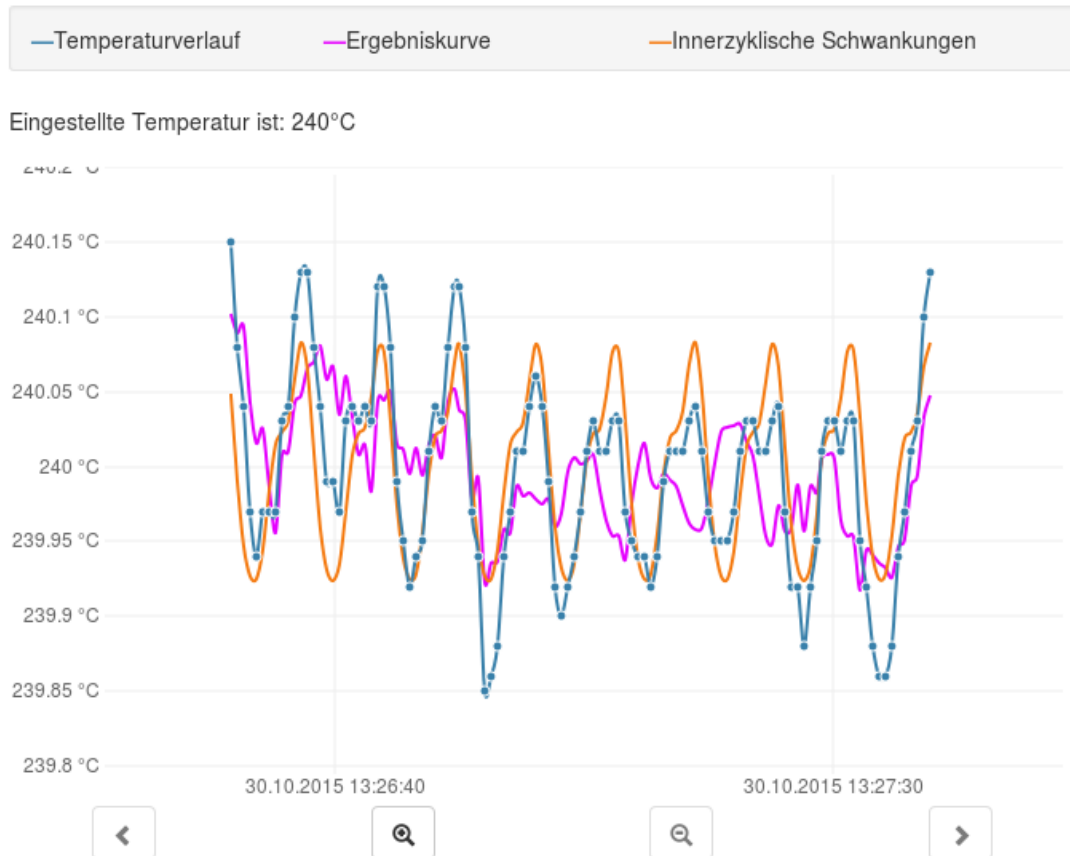


Abbildung 4.6: Temperaturkurve, Ergebniskurve und innerzyklische Schwankungen

4.4.7 Innerzyklische Schwankungen

Hilfskurve zur Verständnisförderung der Ergebniskurve. Entsteht durch die "Innerzyklische Schwankung" addiert mit dem eingestellten Temperaturwert der Zone zum Zeitpunkt der ersten dargestellten Messung⁴.

⁴da die Kurve sonst um den Nullpunkt schwingt

4.5 Resultate

Der Resultate-Abschnitt ist der zentrale Teil des Webtools. Es zeigt diejenigen Kurven auf einem Plot, welche für die Maschinensteuerung zentral sind. Es sind drei Temperaturverläufe ohne die zyklische Temperaturschwankungen. Jede ist ein Resultat eines unterschiedlichen Verfahrens und glättet die Temperaturkurve auf unterschiedlich starke Weise. Je stärker die Kurve geglättet ist, desto mehr Information über die Temperaturschwankungen geht verloren.

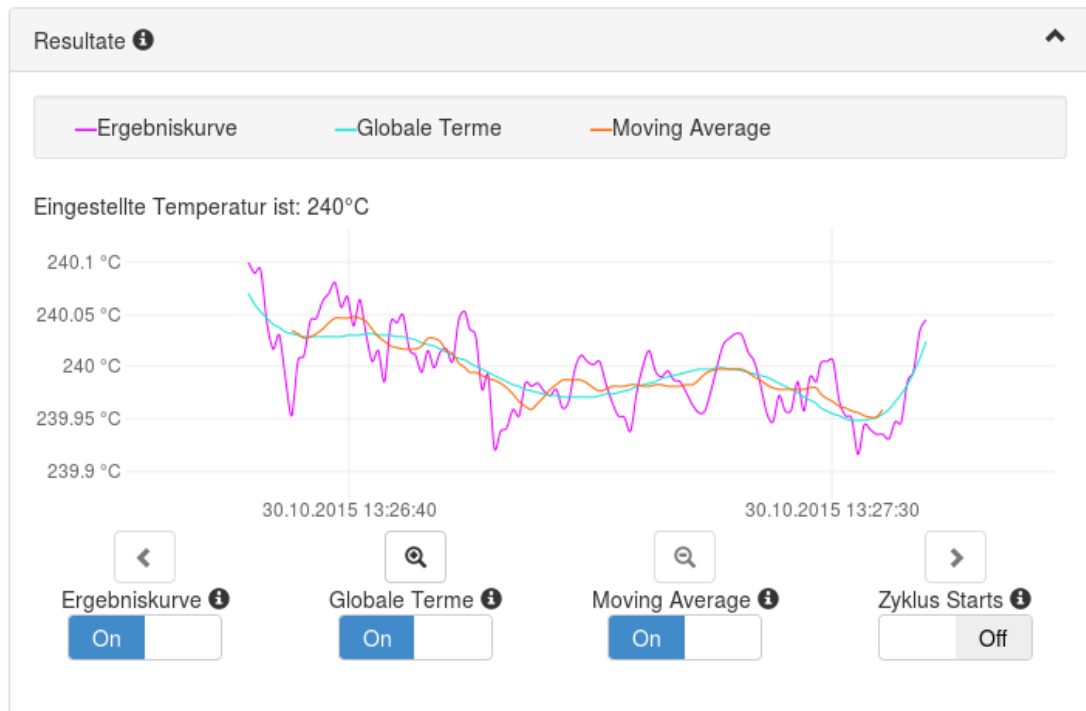


Abbildung 4.7: Übersicht des Resultate-Abschnitts

4.5.1 Plot

Um entscheiden zu können, welche der drei Verfahren zu einem gegebenen Messdaten-Temperaturverlauf am besten ist, können alle drei Kurven unabhängig voneinander angezeigt oder entfernt werden.

4.5.2 Ergebniskurve

Wie bereits unter 4.4.6 beschrieben, stellt die Ergebniskurve den Temperaturverlauf ohne zyklische Temperaturschwankungen dar. Dazu wird vom Messwert die innerzyklische Temperatur abgezogen. Dieses Verfahren behält am meisten Information über den Temperaturverlauf ohne die zyklischen Temperaturschwankungen. Jeder Temperatursprung eines Messwerts, der nicht zyklusbasiert ist, bleibt in dieser Kurve bestehen.

4.5.3 Globale Terme

Entspricht einer auf globale Terme beschränkten linearen Regressionskurve. Die globalen Terme sind zyklusübergreifende Fourier- und polynomielle Terme. Dadurch wird der grobe Temperaturverlauf nachgebildet. Dieses Verfahren behält am wenigsten Information und glättet die Temperaturkurve am stärksten. Dies kann unter Umständen von Interesse sein, da nur signifikante Änderungen in der Temperatur berücksichtigt und so die kleinen Einflüsse entfernt werden. Achtung: Es gibt nicht immer eine Kurve für globale Terme. Falls die Temperatur stabil bleibt oder der Messdatenausschnitt zu klein gewählt wurde, entfallen diese Terme.

4.5.4 Moving Average

Für jeden Messpunkt wird ein arithmetisches Mittel⁵ aller Messpunkte, welche sich bis zu 1/2 Zyklus vor oder nach dem aktuellen Messwert befinden, berechnet. Anders gesagt wird ein Fenster mit der Breite eines Zyklus definiert und über die Messwerte geschoben. Jedes Fenster hat als Ergebnis einen neuen Temperaturwert, der sich aus dem arithmetischen Mittel aller im Fenster befindlichen Messpunkte berechnet.

Die Kurve beginnt erst 1/2 Zyklus nach dem ersten ausgewählten Messpunkt und endet 1/2 Zyklus vor dem Letzten. Dies ist auf die Berechnungsart der Kurve zurückzuführen, da "links" und "rechts" je 1/2 Zyklus mit Messdaten benötigt wird. Unter Anwendung dieses Verfahrens werden Detailinformationen des Temperaturverlauf entfernt. Der Einfluss einzelner Messwerte auf die Temperaturkurve verringert sich, die Information geht trotzdem nicht ganz verloren. Je nach Wunsch kann die Berechnung des moving Average angepasst werden, indem eine andere mathematische Funktion gewählt wird. Damit kann zum Beispiel einem Messpunkt mehr Gewichtung zugesprochen und somit die Glättung der Temperaturkurve beeinflusst werden. Ein weiterer Vorteil des moving Averages ist die Resistenz gegenüber Veränderungen der Zykluszeiten und der Zyklus-Temperaturcharakteristik, womit die lineare Regression ihrerseits ungenau werden kann.

4.5.5 Zyklus-Starts

Als optische Hilfe können die Zyklus-Starts wie unter 4.4.4 beschrieben angezeigt werden. Dies kann für die Entscheidung, ob die zyklischen Temperaturschwankungen restlos entfernt werden können, hilfreich sein⁶.

⁵auch Durchschnitt oder Mittelwert genannt

⁶Achtung, diese Interpretation ist mit dem menschlichen Auge schwierig und sollte nur als geringe Entscheidungshilfe dienen.

4.6 Weiterführende Informationen

In diesem Bereich finden sich Informationen mit dem Zweck, die lineare Regression und die daraus abgeleiteten Kurven in den Bereichen “Messdaten Auswertung“ und “Resultate“ zu verstehen und validieren zu können. Dazu gibt es unter Benchmarks Hinweise und Vergleichswerte, wie viel besser oder schlechter die jeweiligen Verfahren gegenüber der Ausgangslage sind.

4.6.1 Intervall

Der Intervall-Tab enthält Informationen über die Temperaturschwankungen der Zyklen und die Elemente der linearen Regression, welche sich auf diese beziehen⁷.

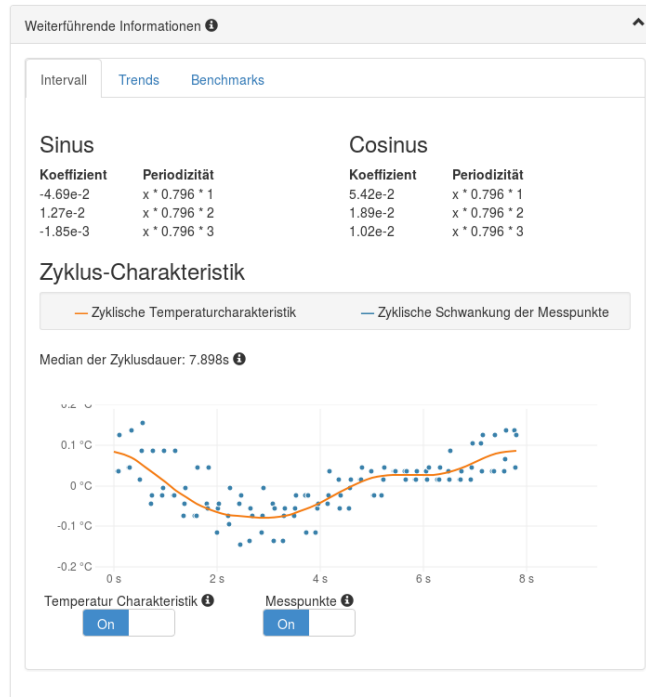


Abbildung 4.8: Weiterführende Informationen auf dem Reiter Intervall

Lokale Terme

Alle Sinus- und Cosinusterme mit Periodizität \leq Zyklusdauer, welche für die Erzeugung der Regressionskurve berücksichtigt wurden. Der Koeffizient ist die Amplitudenhöhe und ein Indikator dafür, wie viel Einfluss dieser Term auf die gesamte Regression hat.

Zyklus Charakteristik

Die Zyklus-Charakteristik zeigt den relativen Temperaturverlauf eines Zyklus an. Dieser besteht aus den oberhalb aufgeführten lokalen Termen der linearen Regression. Zusätzlich wird zu allen ausgewählten Messpunkten der zyklusabhängige Temperaturoffset, geeicht auf einen Zyklus, dargestellt. Damit kann die lineare Regression überprüft werden und klärt, weshalb die innerzyklische Schwankung der linearen Regression eine bestimmte Charakteristik aufweist. Folgen die Punkte keinem eindeutigen Muster oder sind sehr stark verteilt, kann darauf geschlossen werden, dass keine zyklische Charakteristik existiert oder diese innerhalb des ausgewählten Bereichs Änderungen unterworfen ist. In diesem Fall sollte von der linearen Regression kein Gebrauch gemacht werden und das moving Average für Analysezwecke Verwendung finden.

⁷lokalen Terme

4.6.2 Trends

Unter den Trends sind die globalen Terme der linearen Regression mit ihren Koeffizienten aufgelistet. Zu diesen gehören polynomielle (ohne x^0) Terme, sowie Sinus- bzw. Kosinusterme mit Periodizität $>$ Zyklusdauer. Sind solche Terme vorhanden, erscheint ein Warnsymbol rechts neben dem "Trends" Reiter. Die globalen Terme sind ein Hinweis darauf, dass die Temperatur der aktuell ausgewählten Zone trotz herausgefilterten Zyklus-Temperaturschwankungen nicht stabil ist.

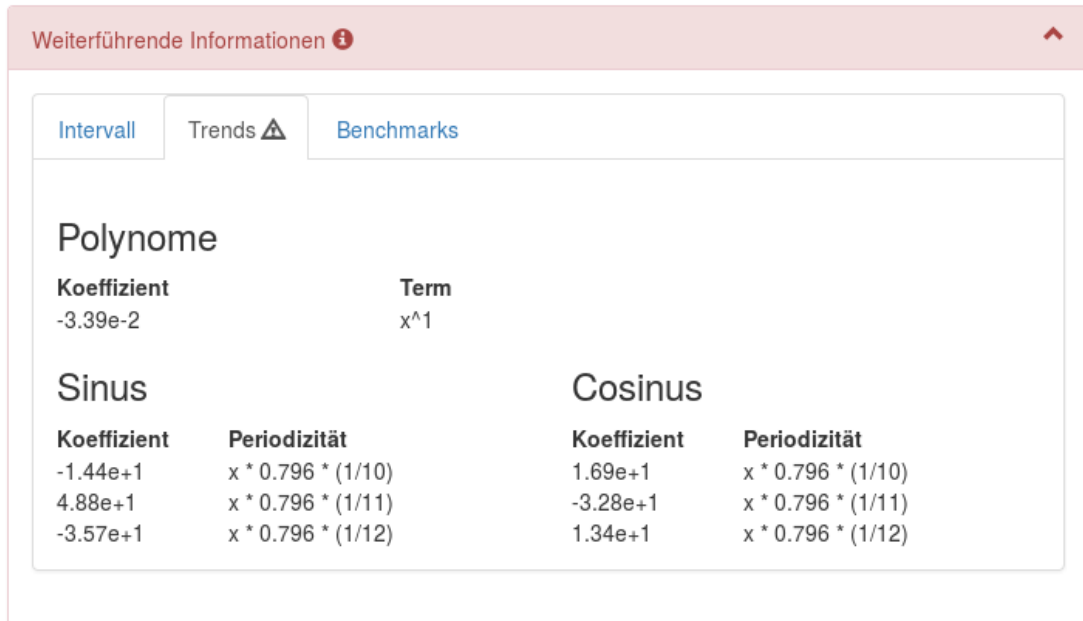


Abbildung 4.9: Weiterführende Informationen auf dem Reiter "Trends"

4.6.3 Benchmarks

Die Benchmarks sind eine Hilfe, um den Mehrwert und die Güte der linearen Regression darzustellen. Dies wird durch Zahlenwerte zwischen 0 und 1 dargestellt. Um eine einfachere Interpretation zu erlauben, korreliert der Wert mit einem farbigen Balken. Die Farbe entspricht dabei einem Ampelstatus.

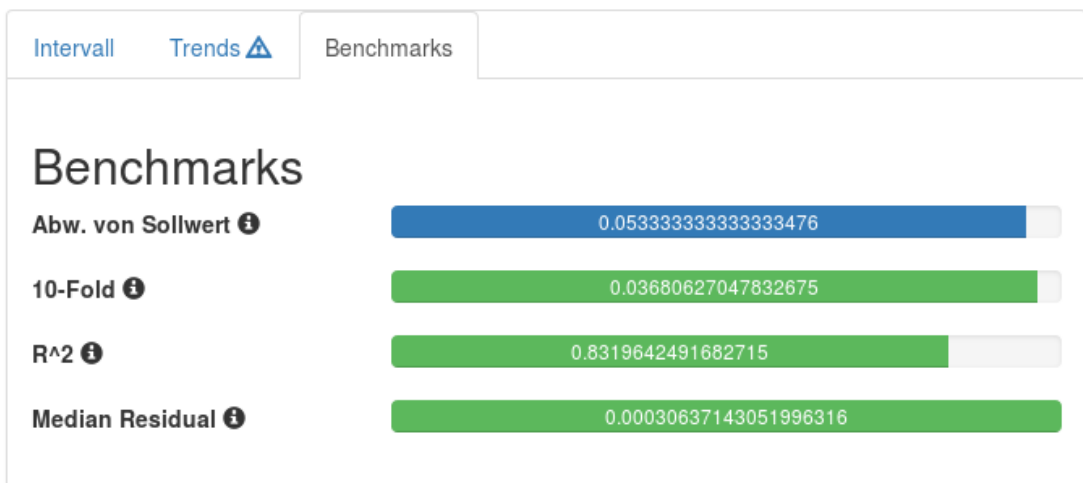


Abbildung 4.10: Weiterführende Informationen auf dem Reiter "Benchmarks"

Abweichung von Sollwert

Entspricht der durchschnittlichen Abweichung einer Temperatur von ihrem Sollwert, genauer definiert in Kapitel 3.7.3. Der Wert dient in der Ansicht zum Vergleich, ob mit der linearen Regression ein präziserer Wert berechnet werden kann, als wenn man ohne statistische Aufbereitung eine Temperaturabweichung berechnet.

10-Fold

Berechnet die durchschnittliche Abweichung einer Vorhersage zum tatsächlichen Messwert, indem der Datenausschnitt 10 Mal auf verschiedene Arten aufgeteilt und getestet wird. Details zur Funktionsweise des Algorithmus in den Kapiteln 3.5.3 k-Fold und 3.7.3 10-Fold.

Die Farbcodierung des Balkens folgt diesen Regeln mit absteigender Priorität:

- Grün: Wert < 0.1
Modell kann Daten gut vorhersagen
- Gelb: Wert $>$ Wert aus "Abweichung von Sollwert" oder Wert ist zwischen 0.1 und 0.5
Modell ist nur bedingt hilfreich
- Rot: Wert > 0.5
Modell ist nicht hilfreich

 R^2

Indikator, wie gut die Daten durch die verwendeten (Sinus-, Kosinus- und Polynom-) Komponenten approximiert werden können, erklärt in Kapitel 3.7.3 R^2 . Der Balken wird nach folgendem Schema eingefärbt:

- Grün: Wert > 0.7
Komponenten nähern sich dem Datenverlauf ausreichend gut an
- Gelb: Wert ist zwischen 0.4 und 0.7
Aussagekraft des Modells ist fraglich
- Rot: Wert < 0.4
Modell ist nicht hilfreich

Median Residual

Entspricht dem Median aller Abweichungen vom linearen Modell zu den tatsächlichen Daten, erklärt in Kapitel 3.7.3 Median Residual. Farbcodierung folgt dem Muster:

- Grün: Wert < 0.1
Kleine Median-Abweichung
- Gelb: Wert ist zwischen 0.1 und 0.5
Präzision des Modells ist fraglich
- Rot: Wert > 0.5
Abweichung sehr gross, Modell approximiert Daten nicht ausreichend genau

Kapitel 5

Anhang

5.1 Installationsanleitung

5.1.1 Workspace: Frontend

Voraussetzungen

Um am Frontend entwickeln zu können, werden folgende Tools benötigt:

- yeoman <http://yeoman.io/>
Erleichtert Erstellung und Verwaltung von Angular-Komponenten
- bower <http://bower.io/>
Dependency-Management von JavaScript-Frameworks
- grunt <http://gruntjs.com/>
Build System für den Workspace

Setup

Sind die Voraussetzungen erfüllt, muss der Workspace zuerst initialisiert werden, indem die JavaScript-Frameworks heruntergeladen und an die korrekten Orte kopiert werden. Dies wird durch folgende Befehle ausgeführt:

```
npm install  
bower install
```

lokale Ausführung

Die Applikation kann lokal gestartet werden, indem

```
grunt serve
```

ausgeführt wird.

5.1.2 Workspace: Backend

Das Setup des Workspaces wird am Beispiel von Tomcat und Eclipse beschrieben.

Voraussetzungen

Um die hier beschriebene Anleitung zu befolgen werden folgende Applikationen benötigt:

- Eclipse (inkl. Maven Dependency-Management)
- Tomcat WebServer
- JDK 7 oder neuer
- Lokale MySQL, MariaDB oder Postgres Installation

Setup

In einem ersten Schritt muss das Projekt in Eclipse importiert werden. Beim Kompilieren des Projekts lädt Maven die verwendeten Libraries automatisch herunter.

Server-Konfiguration

In der Server-View von Eclipse wird in einem ersten Schritt ein Tomcat-Server mit Hilfe des Eclipse Wizards erstellt. Hier soll auch direkt das Projekt als Resource des Servers hinzugefügt werden.

Um die Applikation von der verwendeten Datenbank-Technologie zu entkoppeln, wird die Datenbankverbindung in der WebServer-Konfiguration spezifiziert. Für den eben erstellten Server sollte ein "Servers" Ordner in der Projektstruktur angezeigt werden. Die Konfiguration ist in der Datei "server.xml" vorzunehmen.

Innerhalb des Elements "<Host>" muss dazu ein Context für die Webapplikation mit folgenden Werten erstellt werden:

Listing 5.1: Beispiel-Konfiguration für lokale MySQL Datenbank

```
<Context docBase="zyktemp" path="/zyktemp" reloadable="true" source="
  org.eclipse.jst.jee.server:zyktemp">
  <Parameter name="dataSource.driver" value="com.mysql.jdbc.Driver"
    override="false" />
  <Parameter name="dataSource.url" value="jdbc:mysql://localhost:3306/
    zyktemp" override="false" />
  <Parameter name="hibernate.dialect" value="org.hibernate.dialect.
    MySQLDialect" override="false" />
  <Parameter name="dataSource.username" value="root" override="false" /
    >
  <Parameter name="dataSource.password" value="" override="false" />
  <Parameter name="hibernate.hbm2ddl.auto" value="update" override="
    false" />
</Context>
```

Wurde das Projekt zuvor bereits zu den Ressourcen des Servers hinzugefügt, so existiert ein leerer "<Context>" Eintrag in der Konfiguration. Dieser kann durch die angepasste Konfiguration ersetzt werden.

Listing 5.2: Beispiel-Konfiguration für lokale Postgresql Datenbank

```

<Context docBase="zyktemp" path="/zyktemp" reloadable="true" source=
  "org.eclipse.jst.jee.server:zyktemp">
  <Parameter name="dataSource.driver" value="org.postgresql.Driver"
    override="false" />
  <Parameter name="dataSource.url" value="jdbc:postgresql://localhost/
    zyktemp" override="false" />
  <Parameter name="dataSource.username" value="zykuser" override="false"
    />
  <Parameter name="dataSource.password" value="netstal" override="false"
    />
  <Parameter name="hibernate.dialect" value="org.hibernate.dialect.
    PostgreSQL9Dialect" override="false" />
  <Parameter name="hibernate.hbm2ddl.auto" value="update" override="false
    " />
</Context>

```

Datenbankverbindung

Standardmässig sind Treiber für MySQL/MariaDB und Postgres im Projekt enthalten. Sollen weitere Treiber benötigt werden, so können diese durch Anpassung der “pom.xml“-Datei¹ hinzugefügt werden.

Zur Änderung der unterliegenden Datenbank sind die Werte *dataSource.driver*, *dataSource.url*, *hibernate.dialect*, *dataSource.username* und *dataSource.password* in der Server-Konfiguration anzupassen.

5.1.3 Datenbank

Tabellen

Für die Webapplikation werden zwei Tabellen benötigt. Für MySQL können diese mit folgendem SQL-Skript erstellt werden:

Info: Bei einer Postgresql Datenbank die Zeitstempel nur als **TIMESTAMP** angeben ohne den Klammern und der Zahl.

Listing 5.3: SQL-Skript zur Erstellung der Tabellen

```

CREATE TABLE measurements (
  id BIGINT NOT NULL PRIMARY KEY,
  time TIMESTAMP(6) NOT NULL,
  expected NUMERIC(5,2),
  actual NUMERIC(5,2),
  zone VARCHAR(20),
  heat_setting NUMERIC(6)
);

CREATE TABLE cycles (
  id BIGINT NOT NULL PRIMARY KEY,
  started TIMESTAMP(6) NOT NULL,
  finished TIMESTAMP(6) NOT NULL
);

```

¹Verfügbare Treiber sind auf dem Maven-Repository zu finden

Datenaufbereitung

Nachfolgend ist beschrieben, wie die Mess- und Zyklendaten für einen Import in die Datenbank aufbereitet werden müssen.

measurements.csv:

Spalten-Inhalte: Id, Zeitstempel, Zone, Soll-Temperatur, Ist-Temperatur, Heizstellwert

- *Id*: einzigartige Id eines Messwerts. Auch über verschiedene Files darf diese Nummer nicht doppelt vorkommen²
- *Zeitstempel*: Angabe im Format: yyyy-MM-dd hh:mm:ss.SSS³
- *Zone*: Eine Zeichenfolge der Länge 20
- *Soll-Temperatur*: Die gesetzte Temperatur, auch als Gleitkommazahl möglich. Bis max: 999.99 Auflösung: 2 Stellen nach dem Komma
- *Ist-Temperatur*: Die gemessene Temperatur, auch als Gleitkommazahl möglich. Bis max: 999.99 Auflösung: 2 Stellen nach dem Komma
- *Heizstellwert*: Ganze Zahl, max: 6 Stellig

	A	B	C	D	E	F
1	1	2015-10-30 11:27:25.260	10	240	237.43	0
2	2	2015-10-30 11:27:25.260	17	230	229.81	705
3	3	2015-10-30 11:27:25.340	18	230	229.96	6192
4	4	2015-10-30 11:27:25.420	13	240	240.03	5487
5	5	2015-10-30 11:27:25.500	14	240	239.97	2667
6	6	2015-10-30 11:27:25.580	15	240	240.03	2268
7	7	2015-10-30 11:27:25.660	16	235	235.05	1901
8	8	2015-10-30 11:27:26.540	10	240	237.45	0
9	9	2015-10-30 11:27:26.540	17	230	229.81	705
10	10	2015-10-30 11:27:26.620	18	230	229.96	5818
11	11	2015-10-30 11:27:26.700	13	240	240.03	5810
12	12	2015-10-30 11:27:26.780	14	240	239.97	2677
13	13	2015-10-30 11:27:26.860	15	240	240.04	2254
14	14	2015-10-30 11:27:26.940	16	235	235.05	1906
15	15	2015-10-30 11:27:27.820	10	240	237.45	0

Abbildung 5.1: measurements.csv

²Falls in der DB die Id auf "autoincrement" gesetzt wird, kann diese Spalte weggelassen werden (auch im Datenimport!)

³S = Milisekunde

cycles.csv:

Spalten-Inhalte: Id, Zeitstempel Zyklus-Start, Zeitstempel Zyklus-Ende

- *Id*: einzigartige Id eines Messwerts. Auch über verschiedene Files darf diese Nummer nicht doppelt vorkommen⁴
- *Zeitstempel Zyklus-Start*: Angabe im Format: yyyy-MM-dd hh:mm:ss.SSS⁵
- *Zeitstempel Zyklus-Ende*: Angabe im Format: yyyy-MM-dd hh:mm:ss.SSS

	A	B	C
1	1	2015-10-30 10:49:55.342	2015-10-30 10:50:03.240
2	2	2015-10-30 10:50:03.244	2015-10-30 10:50:11.148
3	3	2015-10-30 10:50:11.152	2015-10-30 10:50:19.050
4	4	2015-10-30 10:50:19.054	2015-10-30 10:50:26.948
5	5	2015-10-30 10:50:26.952	2015-10-30 10:50:34.850
6	6	2015-10-30 10:50:34.854	2015-10-30 10:50:42.750
7	7	2015-10-30 10:50:42.752	2015-10-30 10:50:50.650
8	8	2015-10-30 10:50:50.652	2015-10-30 10:50:58.550
9	9	2015-10-30 10:50:58.554	2015-10-30 10:51:06.450
10	10	2015-10-30 10:54:29.254	2015-10-30 10:54:29.556
11	11	2015-10-30 10:54:34.342	2015-10-30 10:54:47.848
12	12	2015-10-30 10:54:47.850	2015-10-30 10:54:55.758
13	13	2015-10-30 10:55:08.682	2015-10-30 10:55:16.588
14	14	2015-10-30 10:55:17.124	2015-10-30 10:55:25.038
15	15	2015-10-30 11:37:56.034	2015-10-30 11:37:56.336

Abbildung 5.2: cycles.csv

Datenimport

Um die in eine .csv-File aufbereiteten Daten in die Datenbank zu importieren, kann wie folgt vorgegangen werden⁶:

MySQL Falls das Kommandozeilen-Tool “mysqlimport“ vorhanden ist, können die Daten mit diesen Befehlen importiert werden:

```
mysqlimport --local --fields-optionally-enclosed-by= --fields-escaped-by= --fields-terminated-by=,
--columns="id,time,zone,expected,actual,heat_setting" --user="root" --password zyktemp
measurements.csv
```

```
mysqlimport --local --fields-optionally-enclosed-by= --fields-terminated-by=,
--columns="id,started,finished" --user="root" --password zyktemp cycles.csv
```

Postgres Nach Verbinden mit der Datenbank über das Kommandozeilen-Tool können diese Befehle ausgeführt werden:

⁴Falls in der DB die Id auf “AUTO INCREMENT“ gesetzt wird, kann diese Spalte weggelassen werden (auch in der Spaltendefinition des Datenimport-Befehls!)

⁵S = Millisekunde

⁶um eine Beispiel Datei zu laden, existieren im Ordner 05-RealMeasurements/initial-csv-to-install/ auf der Abgabe-CD zwei Beispiel CSVs

```
\copy measurements(id,time,zone,expected,actual,heat_setting) FROM measurements.csv WITH  
DELIMITER ',' CSV
```

```
\copy cycles(id,started,finished) FROM cycles.csv WITH DELIMITER ',' CSV
```

5.1.4 Server: Frontend

Die Installation des Frontends auf einem Server wird am Beispiel eines Linux-Servers⁷, auf dem eine Apache-Instanz läuft, erklärt.

Deployment

Auf einem System, das den Voraussetzungen aus Kapitel 5.1.1 und die Setup-Prozedur aus selbigem Kapitel durchgeführt wurden, kann das Projekt wie folgt gebildet werden:

```
grunt build
```

Im Projekt-Verzeichnis entsteht nun ein “dist“-Ordner. Die Inhalte dieses Ordners werden in das Verzeichnis “/var/www/zyktemp“ des Servers kopiert. Bei korrekter Apache2-Konfiguration ist die Applikation dann unter “http://<IP-Adresse>/zyktemp“ erreichbar.

5.1.5 Server: Backend

Für die Server-Installation des Backends wird ein Server mit einer Tomcat 7 (oder neuer) benötigt.

Deployment

Der Java-Workspace wird mit dem Maven-Befehl

```
mvn clean install
```

in eine .war-Datei kompiliert.

Diese Datei wird auf dem Server in das “webapps“-Verzeichnis der laufenden Tomcat-Instanz kopiert, wo sie automatisch entpackt und ausgeführt wird. Damit die korrekten Parameter bezüglich Datenbankkonfiguration an die Laufzeitumgebung übergeben werden, muss die Context-Datei “zyktemp.xml“⁸ in das Verzeichnis “/etc/tomcat7/Catalina/localhost“ kopiert werden. Gegebenenfalls ist ein Neustart der Tomcat-Instanz notwendig, damit die neuen Umgebungsvariablen gesetzt werden.

Um das Backend über eine Apache-Instanz verfügbar zu machen, ist eine Beispielkonfiguration⁹ für Apache mitgeliefert. Diese Konfiguration soll in das Verzeichnis “/etc/apache2/sites-available“ kopiert und mit dem Kommando

```
a2ensite <Konfigurationsname>
```

aktiviert werden. Das Backend wird mit der Beispielkonfiguration unter der Adresse “http://<IP-Adresse>/api“ verfügbar gemacht. Alle Requests an diese URL werden an die lokale Tomcat-Instanz unter “http://localhost:<Port>/zyktemp“ weitergeleitet.

⁷Für die Studienarbeit wurde Ubuntu 14.04 verwendet

⁸Beispiel-Konfiguration in Anhang, siehe 5.6.2

⁹Siehe 5.6.1

5.2 Semantisches Sprachmodell

Dieser Abschnitt dient der Vereinheitlichung der benutzten Sprache. Fachbegriffe und deren Definitionen werden hier im Stil eines Glossars festgehalten.

F-Test

Statistischer Test zur Ermittlung der Wahrscheinlichkeit einer Hypothese; Im Kontext der Studienarbeit verwendet, um Wahrscheinlichkeit der Existenz eines Trends festzustellen.

Lineare Regression

Statistisches Verfahren um die passendste mathematische Funktion zu einer Datenwolke zu finden. Beschränkt auf Koeffizienten in erster Potenz (meist $a + b \cdot x$)

Nichtlineare Regression

Keine Beschränkung der Regressionskoeffizienten auf die erste Potenz. Ebenfalls sind Fourrierterme möglich (zB. $a + b \cdot x + c \cdot x^2 + d \cdot \sin(x) + e \cdot \cos(x)$).

Regressionskoeffizient

Der Einfluss einer Variable ausgedrückt in einem Faktor (zB $a + b \cdot x + c \cdot x^2$, die Koeffizienten a, b, c).

Zone

Teil einer Spritzgiessmaschine. Enthält ein Temperaturregelelement und ein Temperatursensor zur Temperaturmessung des durchfließenden Materials.

Plastifiziereinheit

Teil einer Spritzgiessmaschine. Enthält bis zu 20 Zonen.

Werkzeug

Ein spezifische Spritzgiessmaschine. Enthält bis zu 200 Zonen.

Trend

Entwicklungstendenz; Im Kontext der Studienarbeit für Temperaturtrends verwendet.

Zyklus

Ein sich wiederholendes Ereignis. In diesem Kontext das fertigen von Werkstücken.

Zyklusdauer

Für die Fertigstellung eines Werkstücks benötigte Zeit.

Innerzyklisch / Saisonal

Verhaltensmuster die sich auf einen Zyklus beschränken. Meistens ausgelöst durch das wiederholte fertigen von Werkstücken.

Zyklus übergreifend / Trend

Verhaltensmuster die sich über mehrere Zyklen erstrecken. Auslöser ist nicht das wiederholte fertigen von Werkstücken.

Lokale Terme

Sinus und Cosinus, dessen Periodendauer gleich der Zyklusdauer oder einer Bruchteil davon entspricht. Bildet innerzyklische / saisonale Verhalten ab.

Globale Terme

Sinus und Cosinus, dessen Periode gleich der Zyklusdauer oder vielfachen davon entspricht. Bildet Zyklus übergreifende Verhalten / Trends ab.

Residuum / Residuen

Abweichung eines Wertes zum gewünschten Ergebnis, Plural auch Residuen. Quelle: Wikipedia [5].

Periodizität

Aussage darüber wie oft sich das Ereignis in einer bestimmten Zeitdauer wiederholt. Quelle: Wikipedia [4]

Resultate

In diesem Kontext, die Ergebnisse der Problemstellung.

Ergebniskurve

Ergebnis der Problemstellung mit Ansatz der linearen Regression.

Benchmark

Hilfestellung um Fragen anhand von einfachen Indizien beantworten zu können. Hier Qualität der Resultate anhand eines Ampelstatus ablesen.

Zyklus Charakteristik

Das Verhalten der Temperatur, ausgelöst durch das wiederholte fertigen von Werkstücken.

Frontend Als "Frontend" wird die Browser-Applikation von ZykJTemp bezeichnet, welche vom Benutzer aufgerufen wird.

Backend Bei der als "Backend" bezeichneten Komponente handelt es sich um einen Java-WebService.

ZykJTemp Abkürzung dieser Arbeit.

5.3 Test-Logs

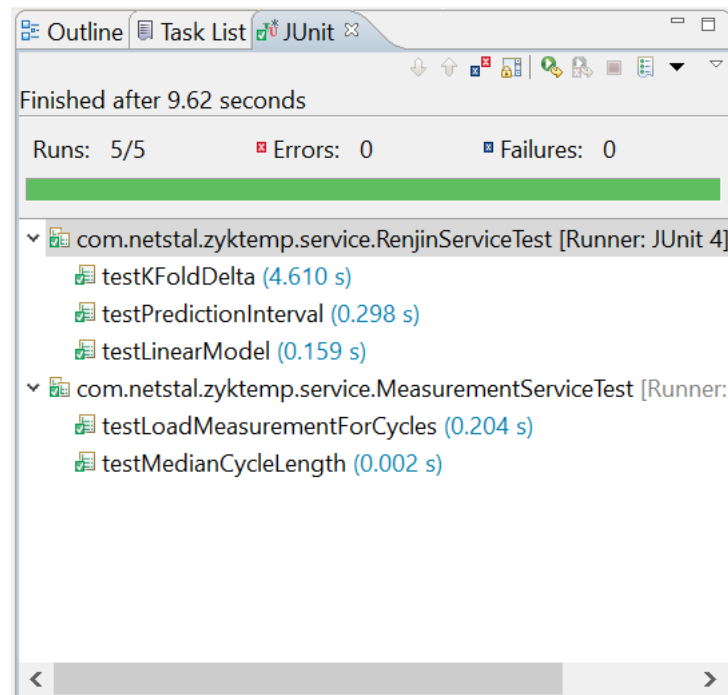


Abbildung 5.3: JUnit Tests

Da sich bei der entwickelten Software um einen Prototypen handelt, wurden nur die Implementation von mathematischen Kernfunktionen getestet.

5.4 Projektplanung

5.4.1 Projektverlauf

Das Team stiess während des gesamten Projektverlaufs auf verschiedene Komplikationen, was besonders zu Beginn der Arbeit durch Unsicherheiten in neuen Technologien wie R oder unbekanntem Vorgehensweisen wie Agile bedingt ist. Durch stetig ändernde Requirements erwies

sich zudem als schwierig, einen geordneten Projektablauf zu erstellen. Grund dafür war eine offen formulierte Zielsetzung, was oft einen zu grossen Interpretationsfreiraum zulies. Beispiel dafür ist die späte Lieferung von realen Messdaten in Projektwoche 8, bis zu welcher man auf Annahmen basierende Messdaten-Emulationen für die Entwicklung des Verfahrens verwendete.

Trotz all den Hindernissen ist uns ein guter Projektverlauf gelungen. Nach einer eher trägen Eingewöhnungszeit konnte man durch Agile während des Projekts auf grössere Änderungen reagieren und sich so grob an die Sprints halten. Trotz der zusätzlichen Flexibilität durch Agile änderten sich meist kleinere Anforderungen zu schnell. Diese wurden an den Meetings mit dem Betreuer oder Auftraggeber festgelegt und in Notizform auf Papier festgehalten. Oftmals handelte es sich dabei um viele eher kleine, wichtige Änderungen, welche möglichst schnell erledigt werden mussten. Daher war es uns unmöglich jeden dieser Changes in Projektmanagementtool einzutragen, geschweige den eine Aufwandsschätzung durchzuführen.

Kennzahlen/Verlauf Issues

- Die Zahl der totalen Issues stieg von 24 auf 42, was einer Vervielfachung von 75% (+18) entspricht.
- 19 Issues von 42 wurden nicht abgearbeitet (aufgrund veränderter Anforderungen).
- 23 Issues wurden abgearbeitet

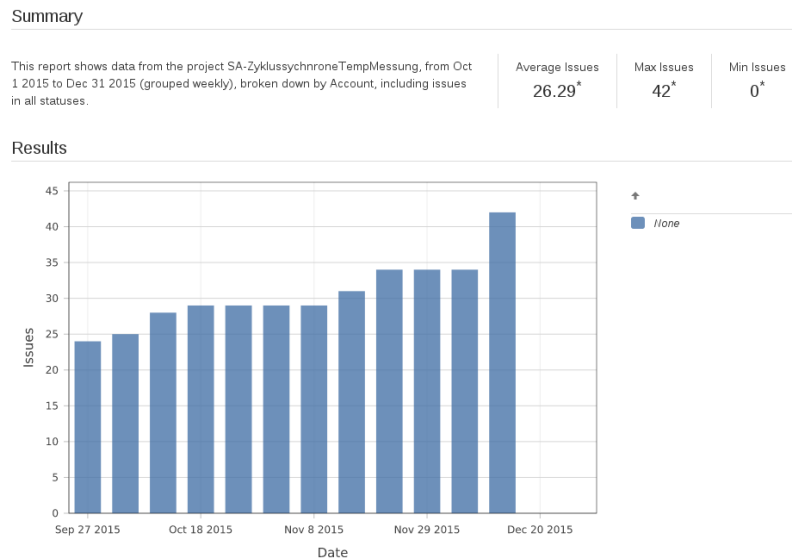


Abbildung 5.4: Veränderung der Issues über das Projekt;

Summary

This report shows data from the project SA-ZyklussynchronTempMessung, from Oct 1 2015 to Dec 31 2015 (grouped weekly), broken down by Account, including issues in the statuses Done.

Average Issues	Max Issues	Min Issues
7.86*	23*	0*

Results

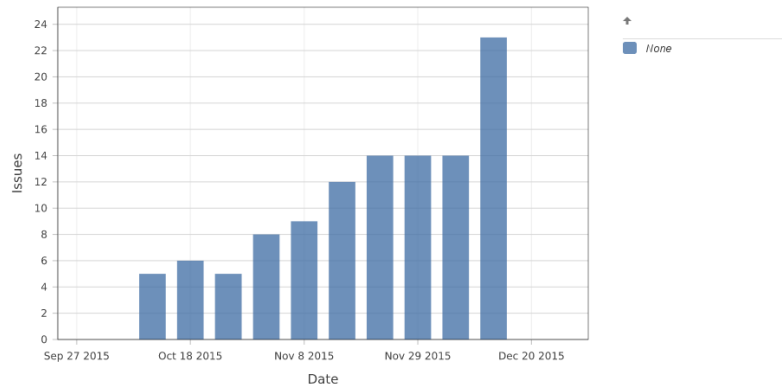


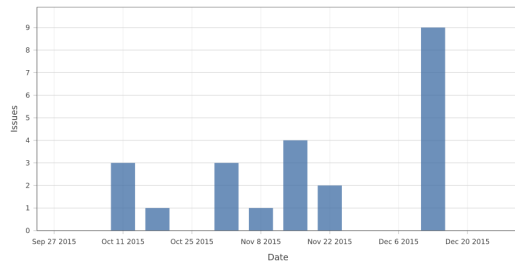
Abbildung 5.5: Burnup-Chart

Summary

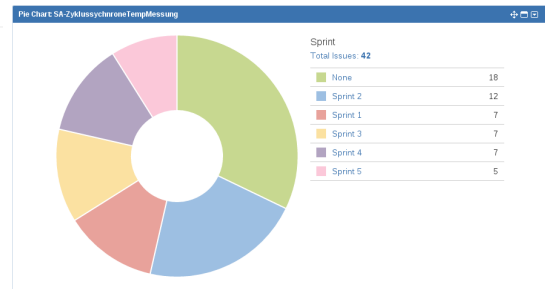
This report shows the count of issues by Resolution Date from the project SA-ZyklussynchronTempMessung, from Oct 1 2015 to Dec 31 2015 (grouped weekly).

Total Issues	Average Issues	Max Issues	Min Issues
23	1.64	9	0

Results



(a) Abgearbeitete Issues pro Woche



(b) Abgearbeitete Issues pro Sprint

Abbildung 5.6: Arbeit geleistet in Issues

5.4.2 Projektaufwand/Zeitabrechnung

Da wir uns während den ersten 2 Wochen in das agilen Vorgehen einarbeiten mussten, wurde erst ab der 3. Woche Zeit eingetragen (dafür Woche 3 mit rückwirkenden Stunden).

Summary

This report shows work logged in the project SA-ZyklussynchroneTempMessung, from Oct 1 2015 to Dec 31 2015 (grouped weekly), broken down by Project.

Total Hours	Average Hours	Max Hours	Min Hours
569.00	40.64	76.00	0.00

Results

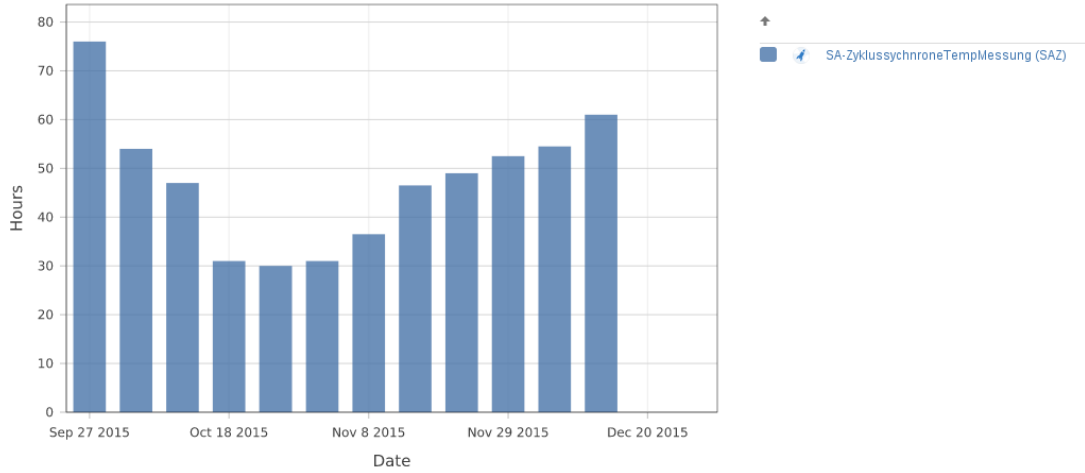


Abbildung 5.7: Total Aufwand in Stunden, pro Woche

Summary

This report shows work logged in the project SA-ZyklussynchroneTempMessung, from Oct 1 2015 to Dec 31 2015 (grouped weekly), broken down by Work Logged By and Epic/Theme.

Total Hours	Average Hours	Max Hours	Min Hours
569.00*	21.88*	50.00*	0.00*

Results

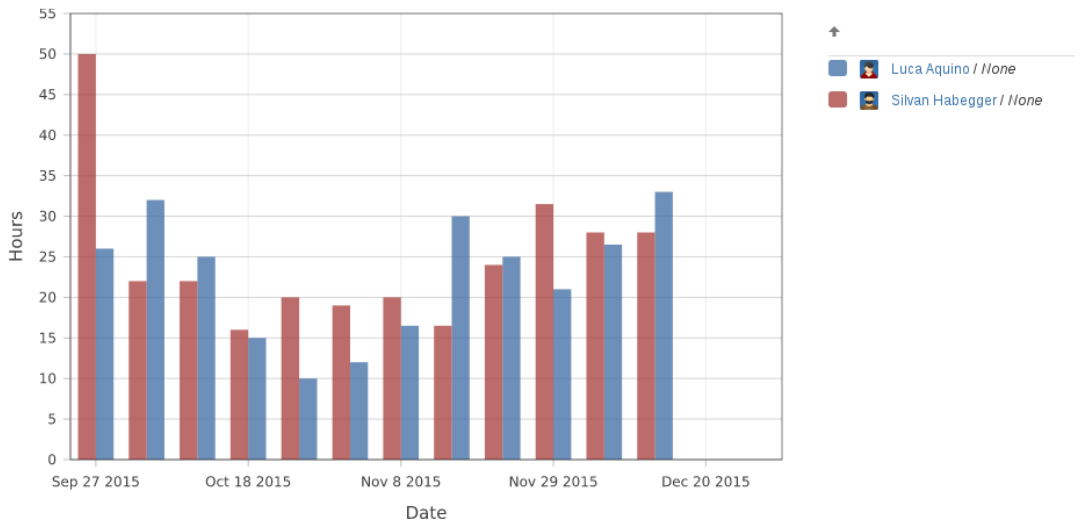


Abbildung 5.8: Aufwand pro Person in Stunden, pro Woche

Unstimmigkeiten Zielsetzung

Die Zielsetzung und damit abgeleitet die Anforderungsanalyse war lange nicht klar und kristallisierte sich erst ab Woche 9, siehe 5.4.2, heraus. Der Grund war, dass Netstal sich gar nicht sicher war, wie die Temperaturverläufe aussehen, und was man genau damit machen will. Die ursprüngliche Idee war es ein viel komplexeres Gebilde an Systemen zu bauen, welche die Maschine simuliert. Dazu war auch geplant eine realtime Analyse der Daten zu machen und diese auf einem maschinenähnlichen Raspberry Pi laufen zu lassen. Je länger wir mit Johannes Heuser von Netstal redeten, desto mehr kristallisierte sich heraus, dass es besser ist ein Tool zu bauen das die Daten analysiert und dafür eignet zu verstehen was in der Maschine vor sich geht und als zweites ein Verfahren zu haben, dass die Zyklus-Schwankungen aus der Temperaturkurve herausrechnet. Konkret konnte man diese Anforderungen erst machen, als man die realen Daten erhalten hatte und sah, was sich damit machen lässt. Des Weiteren gab Herr Professor O. Augenstein uns den Auftrag, eine Validierung der linearen Regression mittels verschieden statistischen Vorgehensweisen zu machen und diese als Benchmark darzustellen.

Unstimmigkeiten User Stories /Anlaufschwierigkeiten mit Agile

Da wir noch nie mit Agile gearbeitet haben, war es für uns neu, dass alle Anforderungen als Story geschrieben werden, dadurch dachten wir, es handelt sich nur um Interaktionen mit dem Benutzer. Wir brauchten insgesamt 3 Wochen und die Hilfe von unserem Betreuer bis Anforderungen in Form von User-Stories standen.

Während des Projekts gab es jedoch so viele Änderungen, dass oftmals auf die Feingranularität der User-Stories verzichtet wurde und viele von Hand geschriebene Anforderungen umgesetzt wurden. Die Stories wurden etwas ungenauer definiert um einen Kompromiss zu finden zwischen dem Agilen vorgehen (Bei dem keine neuen Requirements in einen Sprint kommen, sobald dieser gestartet ist) und den ständig Ändernden Anforderungen.

Projektneuausrichtung

Mit den realen Daten und viel Analyse unsererseits und auf Seiten von unserem Betreuer, sowie vielen Gesprächen mit dem Product-Owner kamen wir zum Schluss dass wir viele von Anfang an definierte Anforderungen nicht umsetzen werden, sondern uns darauf konzentrieren eine Web-Anwendung zu schreiben. Auch mit dem Hintergedanken, dass wir als Informatiker gerne noch etwas in unserer Disziplin machen würden. Da es in der Statistik oft kein richtig und falsch gibt, wurde noch zusätzliche eine Validierung der linearen Regression hinzugenommen.

Anlaufschwierigkeiten mit R

Die statistische Software R war für uns Neuland und wir mussten uns zuerst soweit mal einarbeiten, dass wir unsere eigenen Programme schreiben konnten. Weil sich um viele statistische Ausdrücke drehte von denen wir oftmals noch nichts gehört haben, war es für uns schwierig die richtigen Funktionen in R zu benutzen (da es meist kein eindeutiges richtig oder falsch gibt). Oftmals endete es in einem "try-and-error" Verfahren und erst ein Gespräch mit unserem Betreuer gab weiteren Aufschluss. Das erschwerte ein effizientes Arbeiten.

Renjin

Renjin ist ein JVM basierter Interpreter für R und sollte unsere Leben leichter machen indem Funktionen aus einem bereits geschriebenen R-file in Java genutzt werden können. Es trat jedoch eher das Gegenteil in Kraft. Renjin zeigt viele Schwächen in der Performance und wie wir schmerzlich feststellen mussten, interpretiert es nicht alle Befehle aus R. Die etwas komplizierteren statistischen Funktionen wie eine 10-Fold Analyse sind noch nicht im Repertoire der Renjin. Somit mussten wir diese Funktionen selber in R implementieren, mit Ausdrücken die Renjin kennt.

Produktionsdaten

Wie bereits erwähnt bekamen wir die Produktionsdaten erst in Woche 8, da es für Netstal einen erheblichen Aufwand bedeutete diese zu beschaffen. Ohne diese Daten konnten wir uns nur auf unsere Simulationen verlassen und tappten lange im Dunkeln. Als die Daten endlich zur Verfügung standen mussten wir erst einmal eine Woche investieren um diese für unsere Zwecke gebräuchlich zu machen. Da die Daten als Logfile kamen, war viel Umformatierung und Bereinigung nötig. Weil sich die Zonen sehr stark von der Einigung zu unserem Zweck unterscheiden, mussten wir zuerst herausfinden, wo die Daten zu gebrauchen sind und danach unser Programm so anpassen das die lineare Regression auf die Daten funktioniert.

5.5 Persönliche Berichte

5.5.1 Luca Aquino

Meine Stimmung durchlief während der Studienarbeit mehrere Hochs und Tiefs. Die Gründe dafür fanden sich vor allem im verhältnismässig grossen Neuanteil an Technologien und Verfahren, die im Verlaufe des Projektes zum Einsatz kamen.

Während ich schon im Voraus sehr zufrieden mit Silvan Habegger als Team-Partner und Professor Oliver Augenstein als Betreuer war (was sich über die folgenden Wochen bestätigte), gab es in der ersten Phase viele Probleme mit dem für uns beide noch nie verwendeten agilen Vorgehen und der langen Einarbeitungszeit in R, was die Stimmung zu dieser Zeit etwas trübte. Sichtlich erleichtert waren dann beide Team-Mitglieder, als es schliesslich hiess "jetzt solltet Ihr aber schon mal etwas programmieren".

Trotz Schwierigkeiten - besonders mit der JVM-Implementation von R - war das Team während dieser Phase wieder in seiner Komfortzone und Defizite, welche bis dato gemacht wurden, konnten wieder aufgeholt werden. Die Kommunikation zu Netstal fühlte sich sehr ungezwungen an und wir konnten von den regelmässigen Besprechungen mit Johannes Heusser sehr profitieren. Besonders mit Anlieferung der Maschinendaten von Netstal nahm die Studienarbeit endlich konkretere Züge an. Trotz dem kleinem Dämpfer, dass Zonen der Plastifizierung mit dem gewählten Verfahren nicht sauber analysiert werden konnten, wurden mit Unterstützung von Betreuer Oliver Augenstein und Johannes Heusser Zonen gefunden, in denen die gewählten Auswertungsmethoden anwendbar waren. Eine Möglichkeit, wie Plastifizierungszonen aufbereitet werden können, veranschaulichte uns Prof. Augenstein durch seine Analyse der Messdaten. Sehr spannend war es für mich zu erfahren, dass bei der Bearbeitung von mathematischen/statistischen Problemen eine säuberliche Datenanalyse schon zu Beginn des Projektes unumgänglich ist.

Insgesamt kann ich auf eine in vielen Bereichen sehr lehrreiche Studienarbeit mit einem durchwegs positiven Gefühl zurückblicken. Silvan Habegger und ich ergänzten uns in vielen Bereichen, und da wir oft einer Meinung bezüglich Vorgehen und Prioritäten waren, machte die Zusammenarbeit viel Spass. Auch die Kooperation mit Netstal war, gerade durch Johannes Heussers Interesse am Projekt, sehr ergiebig und half uns trotz oft ändernder Anforderungen auf Kurs zu bleiben. Betreuer Oliver Augenstein stand stets mit gutem Rate zur Seite und wies uns einige Male freundlich und kompetent den Weg, wenn es unsererseits wieder etwas holpriger vorwärts ging.

5.5.2 Silvan Habegger

Nach erhaltenen Zusage für die Studienarbeit freute ich mich besonders auf die Zusammenarbeit mit Herr Professor Oliver Augenstein. Da er mir in früheren Semestern bereits Analysis unterrichtete, wusste ich, wie freundlich und kompetent er ist (was sich auch bestätigte). Ebenfalls freute ich mich auf das Kennenlernen von neuen Technologien.

Zu Beginn war es für mich schwierig ins Projekt zu kommen. Die kurze Theorie über agile Softwareentwicklung in der SE2 Vorlesung reichte nicht aus um mir einen genauen Plan zu machen, wie ich vorzugehen habe. In Verbindung mit den nicht ganz so klaren Anforderungen

löste das ein eher ungutes Gefühl aus. Der Plan, wie das Projekt etwa ablaufen soll, fehlte irgendwie. Einen nicht unwesentlichen Anteil daran hat sicher auch die Tatsache, dass ich vor dem SE2 Projekt noch nie ein Software-Projekt gemacht habe. Als wir in Woche 9 wussten, was wir genau zu tun hatten, löste sich das unguete Gefühl und ich freute mich richtig auf das Entwickeln des Webtools.

Mit R habe ich vor der Studienarbeit erst bei Herr Andreas Müller etwas zu tun gehabt, gab dazumal jedoch nur Befehle aus einem kleinen Tutorial ein, ohne wirklich genau zu wissen, wie man R benützt. Zu Beginn hat es mir extrem Spass gemacht R und seine Möglichkeiten zu entdecken. Mit der Zeit verlor ich jedoch die Freude daran, da der grosse Fortschritt in der SA ausblieb und oftmals die Hilfe von Oliver Augenstein nötig war. Dieses Unselbständig-sein frustrierte mich zwischendurch extrem und somit war ich auch sehr froh, dass sich später Luca um dies kümmerte und ich mich voll motiviert aufs Frontend stürzen konnte.

Die Zusammenarbeit mit Netstal und besonders Johannes Heusser war sehr angenehm und der wöchentliche Austausch mit ihm über Skype empfand ich als super Unterstützung. Am meisten geholfen hat mir der Besuch bei Netstal, um mir ein Bild davon machen zu können, wie so eine Spritzgiess-Maschine funktioniert.

Sehr hilfreich waren wie bereits erwähnt die Meetings mit Oliver Augenstein. Er half uns immer sehr kompetent und wusste immer, wo der Schuh drückte und wie dies zu lösen war. Für die Hilfe und Unterstützung war ich jedes mal sehr dankbar.

Die Zusammenarbeit mit Luca war wie immer super und ohne Probleme. Da wir uns auch sonst oft an der Schule sehen, war die Kommunikation einwandfrei und wir zogen immer am gleichen Strang. Dadurch (und auch dank Oliver Augenstein und Johannes Heusser) machte es trotz vielen Hürden sehr viel Spass, die Studienarbeit durchzuführen und meiner Meinung nach ist uns zum Schluss ein gutes Ergebnis gelungen, auf das ich stolz bin.

5.6 Konfigurationen

5.6.1 Apache2

Listing 5.4: Beispiel Apache-Serverkonfiguration

```

<VirtualHost *:80>
# DocumentRoot fuer FrontEnd Ressourcen
DocumentRoot /var/www

LogLevel warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>

# Weiterleitung aller Requests auf /api an Tomcat-Instanz
ProxyPass /api http://localhost:8081/zyktemp
ProxyPassReverse /api http://localhost:8081/zyktemp
ProxyRequests Off

</VirtualHost>

```

5.6.2 Tomcat

Listing 5.5: Beispiel Tomcat-Serverkonfiguration

```

<?xml version="1.0" encoding="UTF-8" ?>

<Context docBase="/var/lib/tomcat7/webapps/zyktemp" path="/
zyktemp">
<Parameter name="dataSource.driver" value="com.mysql.jdbc.
Driver"
override="false" />
<Parameter name="dataSource.url" value="jdbc:mysql://
localhost:3306/zyktemp"
override="false" />
<Parameter name="dataSource.username" value="zykuser" override=
"false" />
<Parameter name="dataSource.password" value="netstal" override=
"false" />
<Parameter name="hibernate.dialect" value="org.hibernate.
dialect.MySQLDialect" override="false" />
<Parameter name="hibernate.hbm2ddl.auto" value="update"
override="false" />
</Context>

```

5.7 Formulare

5.7.1 Eigenständigkeitserklärung

Wir erklären hiermit:

1. Dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt sind oder mit dem Betreuer schriftlich vereinbart wurden.
2. Dass wir sämtliche verwendete Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.
3. Dass wir keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt haben.

Ort, Datum:

Namen, Unterschriften:

Literaturverzeichnis

- [1] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning. pages 179–181, 2013.
- [2] Herr Dr. Prof Andreas Müller. Skript wahrscheinlichkeit und statistik satz 6.4. *hsr*, page 115, 2015.
- [3] die freie Enzyklopädie Wikipedia. Median (Begriffsklärung). <https://de.wikipedia.org/wiki/Median>. Accessed: 2015-12-12.
- [4] die freie Enzyklopädie Wikipedia. Peridoizität. <https://de.wikipedia.org/wiki/Peridoizität>. Accessed: 2015-12-12.
- [5] die freie Enzyklopädie Wikipedia. Residuum (Numerische Mathematik). [https://de.wikipedia.org/wiki/Residuum_\(Numerische_Mathematik\)](https://de.wikipedia.org/wiki/Residuum_(Numerische_Mathematik)). Accessed: 2015-12-12.

Abbildungsverzeichnis

3.1	Abstrahierung der Maschine	9
3.2	Abstrahierung im Kontext der Maschine	9
3.3	Innerzyklische Temperaturschwankung, Plot von Prof. Augenstein	10
3.4	Ist- und Heizstellwert korrelieren umgekehrt zu einander, Plot von Prof. Augenstein	10
3.5	Aufteilung User-Stories nach Epics	11
3.6	Übersichtsdiagramm ZykJTemp (stark abstrahiert)	17
3.7	Übersichtsdiagramm Backend (abstrahiert)	18
3.8	Packagestruktur ZykJTemp	18
3.9	Datenklassen ZykJTemp	20
3.10	Grün: Kurve mit Overfit (folgt Noise); Orange: Regression mit einem Freiheitsgrad	23
3.11	Datenpermutationen der 5-Fold-Crossvalidation Quelle: ISLR14[1]	24
3.12	Datenpermutationen der Leave-One-Out-Crossvalidation Quelle: ISLR14[1]	25
3.13	Messdaten-Funktion (blau), Samples mit eingerechnetem Fehler (rot)	26
3.14	Temperaturverläufe Plastifizierung	27
3.15	Zone 10 Zyklus-synchron bei Sollwert 240° Celsius	27
3.16	Vorhersage Temperaturverlauf unter Miteinbeziehen eines vorherigen Heizstellwert; Plot von Prof. Augenstein	28
3.17	Ausschnitt Zone 401; Zyklus Start und Ende sind durch graue Linien veranschaulicht	29
3.18	Beispiel von globalen Trends mit wiedererkennbarem saisonalem Verhalten	30
4.1	Illustration der Bereiche im Webtool	37
4.2	Zeitfensterauswahl von Timeline	39
4.3	Vergleich der beiden Möglichkeiten zur Zeitauswahl	40
4.4	Übersicht der Messdatenauswertung	41
4.5	Hineingezoomte Ansicht der Messdaten (inklusive linearer Regression)	42
4.6	Temperaturkurve, Ergebniskurve und innerzyklische Schwankungen	43
4.7	Übersicht des Resultate-Abschnitts	44
4.8	Weiterführende Informationen auf dem Reiter Intervall	46
4.9	Weiterführende Informationen auf dem Reiter “Trends“	47
4.10	Weiterführende Informationen auf dem Reiter “Benchmarks“	47
5.1	measurements.csv	52
5.2	cycles.csv	53
5.3	JUnit Tests	56
5.4	Veränderung der Issues über das Projekt;	57
5.5	Burnup-Chart	58
5.6	Arbeit geleistet in Issues	58
5.7	Total Aufwand in Stunden, pro Woche	59
5.8	Aufwand pro Person in Stunden, pro Woche	59