

# DIPLOMARBEIT

zur Erlangung  
des Grades eines Diplom Ingenieurs  
der technischen Wissenschaften

über das Thema

## Analyse von gemessenen Haushalts- und Gerätelastgängen

eingereicht bei:

**Univ.Prof. Dr.-Ing. Wolfgang Gawlik**  
E370 - Institut für Energiesysteme und Elektrische Antriebe  
Fakultät für Elektrotechnik und Informationstechnik  
Technische Universität Wien

Eingereicht von:

**Michael Koderhold, B.Sc.**  
Matrikelnummer 0825737  
Eichenstraße 36, 2640 Hart

Wien am 26. Februar 2014

## Abstract

At the moment the demand of households is projected with standard load-profiles, so called H0-profiles. These profiles are for example published yearly by E.ON for different areas and are then used by the smaller energy suppliers for demand prognosis. This profile has an resolution of 15 minutes, because the planning in energy economics is also done in this time span. Because of that, devices with shorter usage duration are considered with their mean power over 15 minutes.

During the research project „ADRES“ a very extensive measurement series was recorded. In this thesis it should be compared on basis of the measurement series, if the daily power consumption for devices used in the creation of synthetic load profiles are similar to derived characteristics. After an short introduction to the topic of this thesis, the steps during the programming are described. As a result of this programming work, a graphical user interface for easy adjustment of the desired analysis is created.

As a next step, some literature will be briefly cited, that was recently published on this topic.

The actual analysis are separated into three main sections. The first section evaluates the usage of single devices, the second one investigates the whole households and the third and last one is a trial to extract some of the non detailed measured devices from the sum measurement and describe them.

The creation of the analysis is done in respect with in literature found trend curves. Extensive comparison with the H0-profile is especially done for the sum measurement to evaluate a connection between actually used devices and the widely used H0-profile and thereby verify the extensive usage of this relatively simple model for households on basis of this measurement series.

## Kurzfassung

Im Moment wird der Verbrauch von Haushalten mit einem Standard-Lastprofil, dem so genannten H0-Profil ermittelt. Dieses Profil wird zum Beispiel von E.ON für jedes Jahr für unterschiedliche Gebiete herausgegeben und dann von den kleineren Energielieferanten zur Bedarfsermittlung herangezogen. Dieses Profil hat auf Grund der Viertelstunden-Planung in der Energiewirtschaft auch eine Auflösung von 15 Minuten und erfasst daher kürzer betriebene Verbraucher mit ihrer mittleren Leistung über diese Zeit betrachtet.

Da im Zuge des Forschungsprojektes „ADRES“ eine sehr umfangreiche Messreihe aufgenommen wurde, soll in dieser Arbeit basierend auf dieser Messreihe untersucht werden, in wie weit die verwendeten Tageslastgänge bei der Erstellung von synthetischen Haushaltslastprofilen mit den aus dieser Messreihe abgeleiteten Aussagen übereinstimmen. Dafür wird nach einer Einleitung beschrieben, wie bei der Programmerstellung vorgegangen wurde, worauf zu achten war und als Ergebnis dieser Programmierung wird kurz das graphische User-Interface zur Steuerung der Analysen erklärt.

Im Anschluss wird in aller Kürze auf die Literatur eingegangen, die auf dem Gebiete dieser Arbeit bereits veröffentlicht wurde.

Die Auswertungen gliedern sich auf drei Abschnitte. Im ersten Abschnitt wird der Einsatz von Einzelgeräten untersucht, im zweiten Abschnitt wird der gesamte Haushalt beleuchtet und im dritten Abschnitt wird ein Versuch unternommen, die nicht separat vermessenen Geräte aus dem gesamten Haushaltsverbrauch zu extrahieren und danach zu beschreiben.

Dabei wird bei der Erstellung der Auswertungen immer wieder der direkte Vergleich mit in der Literatur angegebenen Verläufen angeführt und besonders bei der Betrachtung der Summenmessung wird ein Vergleich mit dem Standardlastprofil angestellt und es soll somit untersucht werden, ob ein Zusammenhang zwischen den tatsächlich eingesetzten Geräten in Haushalten und dem meist verwendeten H0-Profil feststellbar ist und ob damit verbunden die weitreichende Verwendung dieses doch recht einfachen Modells für Haushalte auf Basis dieser Messreihe bestätigt werden kann.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Die Messreihe . . . . .	2
1.2. Geräteklassen . . . . .	3
1.3. Methodik . . . . .	4
<b>2. Programmbeschreibung</b>	<b>5</b>
2.1. Informationsdatenstruktur erstellen . . . . .	5
2.2. Geräteeinsatzzeitpunkte suchen . . . . .	13
2.3. Funktionen für die Auswertungen . . . . .	14
2.3.1. Einschaltzeitpunkt . . . . .	14
2.3.2. Einschaltdauer . . . . .	15
2.3.3. Einschalthäufigkeit / Tag . . . . .	16
2.3.4. Nennleistung (Durchschnittswert) . . . . .	16
2.3.5. Nennleistung (Maximalwert) . . . . .	17
2.3.6. Energieverbrauch . . . . .	17
2.3.7. Leistungsverlauf . . . . .	17
2.3.8. Tagesleistungsverlauf . . . . .	18
2.3.9. Vergleich der Einschaltzeitpunkt . . . . .	18
2.3.10. Vergleich der Einschalthäufigkeit / Tag . . . . .	19
2.3.11. Haushaltlastgang . . . . .	19
2.3.12. Restprofil . . . . .	20
2.4. Grafisches User-Interface (GUI) . . . . .	20
<b>3. Literaturrecherche</b>	<b>24</b>
<b>4. Auswertungen</b>	<b>26</b>
4.1. Geräteeinsatz . . . . .	27
4.1.1. zyklische Geräte . . . . .	28
4.1.2. Waschmaschinen . . . . .	44
4.1.3. Wäschetrockner . . . . .	55
4.1.4. Geschirrspüler . . . . .	62
4.1.5. Küchengeräte . . . . .	69
4.1.6. Multimedia . . . . .	83
4.1.7. weitere vermessene Geräte . . . . .	89
4.1.8. Sonstige Geräte . . . . .	98
4.1.9. Zusammenfassung der wichtigsten Geräteparameter . . . . .	100
4.2. Haushaltlastprofile . . . . .	101

## *Inhaltsverzeichnis*

4.3. Restprofil . . . . .	112
4.3.1. Ermittlung des Restprofils . . . . .	112
4.3.2. Betrachtungen für 4 Beispielhaushalte . . . . .	122
4.3.3. prozentuell vermessene Leistung . . . . .	130
<b>5. Schlussfolgerung</b>	<b>133</b>
<b>Abbildungsverzeichnis</b>	<b>135</b>
<b>Tabellenverzeichnis</b>	<b>140</b>
<b>Literaturverzeichnis</b>	<b>142</b>
<b>A. Übersicht über die vermessenen Geräte</b>	<b>143</b>
<b>B. MATLAB Quellcode</b>	<b>148</b>
B.1. build_infostructure() . . . . .	148
B.2. find_relevant_areas(...) . . . . .	152
B.3. show_usage_histogram(...) . . . . .	158
B.4. show_usage_duration(...) . . . . .	161
B.5. show_usage_count(...) . . . . .	164
B.6. show_power_consumption(...) . . . . .	167
B.7. show_energy_consumption(...) . . . . .	169
B.8. show_cycles(...) . . . . .	172
B.9. show_mean_power(...) . . . . .	175
B.10. show_usage_comparison(...) . . . . .	178
B.11. show_usage_comparison_day(...) . . . . .	180
B.12. show_load_profile(...) . . . . .	181
B.13. show_restprofil(...) . . . . .	187

# 1. Einleitung

Der steigende Energieverbrauch und die vermehrte Erzeugung elektrischer Energie aus regenerativen Quellen, die sehr schwer prognostizierbar sind, stellt die Energielieferanten vor immer größere Herausforderungen. Der Bedarf an Regelausgleichsenergie wird immer größer, da die Erzeugung aus beispielsweise Wind- und Photovoltaikanlagen in Sekundenbruchteilen schwanken können und sehr stark von der momentanen Wetterlage abhängen und damit schwer voraussagbar sind.

Die exakte Modellierung des Verbrauches ist daher genauso wichtig wie die sehr genaue Prognostizierung der Erzeugung, um den Bedarf an teurer Regelenergie so gering als möglich zu halten.

Der Haushaltsstromverbrauch hat im Moment einen Anteil von rund 27 Prozent (Quelle [4]) am Gesamtstromverbrauch und daher ist eine genaue Modellierung sehr wichtig. Die Ermittlung des Haushaltsstromverbrauch stellt eine besondere Herausforderung dar, da der Stromverbrauch eines Industriebetriebes, dessen Prozessschritte relativ genau bekannt sind, leichter planbar ist als der momentane Verbrauch eines Haushaltes, der sehr stark vom Benutzerverhalten abhängt und somit großen statistischen Schwankungen ausgesetzt ist.

Diese Modellierung erfolgt mit sogenannten Haushaltslastprofilen oder auch H0-Profilen, die jährlich von unterschiedlichen großen Energielieferanten, wie beispielsweise E.On ([1]) für verschiedene Regionen veröffentlicht werden. Die zeitliche Auflösung dieser Profile ist auf Grund des Planungsintervalls von 15 Minuten in der Energiewirtschaft ebenfalls auf 15 Minuten festgelegt. Geräte mit kürzerer Betriebsdauer werden daher in diesem Intervall mit ihrer mittleren Leistung erfasst.

Im einleitenden Kapitel dieser Arbeit wird auf die hier verwendete Messreihe eingegangen und im Anschluss daran wird die Unterteilung der vermessenen Geräte in die hier verwendeten Geräteklassen vorgenommen. Danach folgt ein kurzer Abschnitt über die weitere Vorgehensweise.

Nach dieser kurzen Beschreibung der erstellten Programme und einem kurzen Abriss über die bereits auf diesem Gebiet veröffentlichte Literatur wird daher in dieser Arbeit zuerst auf einzelne Geräte eingegangen und deren Verhalten mit verschiedenen charakteristischen Werten, wie der Nutzungsdauer und der mittleren Leistung sehr genau beschrieben.

Danach wird der Gesamtverbrauch der Haushalte untersucht und mit einem Standardlastprofil verglichen. Weiters wird in diesem Abschnitt der Blindleistungsverbrauch von

## 1. Einleitung

Haushalten betrachtet und damit verbunden der Verlauf des Leistungsfaktors über einen Tag dargestellt.

In einem abschließenden Kapitel sollen die nicht vermessenen Geräte aus der Summenmessung extrahiert werden. Der Verlauf dieser nicht detailliert vermessenen Geräte wird hier als Restprofil bezeichnet und soll eine Einschätzung ermöglichen, wie viel Prozent des Gesamtverbrauches auf die detailliert gemessenen Geräte entfallen. Weiters soll versucht werden, charakteristische Verläufe im Restprofil zu einzelnen Geräten zuzuordnen.

### 1.1. Die Messreihe

In diesem Kapitel wird kurz auf die Messreihe eingegangen, deren Daten die Grundlage für die Programmerstellung sowie die Auswertungen ist.

Im Prinzip ist die Verwendung der hier erstellten Funktionen zwar für jede Messreihe denkbar, aber um möglichst viele Tests und Auswertungen zu ermöglichen, wurde von der sehr umfangreichen, am Institut vorhandenen Messreihe des Forschungsprojektes ADRES unter der Leitung der TU Wien ausgegangen.

Diese Messreihe wurde in den Jahren 2010/11 von der *Energie AG OÖ Netz GmbH* in insgesamt 42 Haushalten durchgeführt. Zum Einsatz kamen zwei verschiedene Messsysteme für die beiden jeweils gemessenen Daten pro Haushalt. Ein Messsystem wurde zur Erfassung des Gesamtverbrauchs des jeweiligen Haushaltes eingesetzt und das zweite Messsystem wurde zur Detailmessung einzelner Geräte verwendet. Für die Einzelgerätemessung standen pro Haushalt maximal 32 Zwischenstecker mit integrierter Messung und eine Messzentrale zur Aufzeichnung der Messwerte zur Verfügung. Daher wurden in den einzelnen Haushalten je nach Zugänglichkeit der zu vermessenden Geräte immer wieder andere Gerätetypen vermessen. Des weiteren schwankt die Anzahl der vermessenen Geräte zwischen 20 und 32, je nach technischer Möglichkeit zur Vermessung. Fix angeschlossene Verbraucher wie E-Herd, Warmwasserboiler oder Durchlauferhitzer, die für die Analysen sehr interessant wären, konnten wegen des Fehlens eines Steckers dieser Gerätetypen nicht vermessen werden. Der notwendige Umbau wäre zu aufwändig gewesen und hätte zusätzlich eine Elektrofachkraft bei Start und Beendigung der Messreihe erfordert. Pro Haushalt wurde im Normalfall eine Sommer-, sowie eine Wintermessreihe über mehrere Tage durchgeführt. Bei manchen Haushalten fand aber auch nur eine der genannten Messreihen statt. Es wurden dabei im Sommer 35 von 42 Haushalten an insgesamt 366 Tagen vermessen. Im Winter wurden 40 Haushalte an insgesamt 421 Tagen vermessen.

Die gemessenen Daten wurden bereits in vorangegangenen Arbeiten aufbereitet und stehen als MATLAB-Binärdateien zur Verfügung. Dabei ist zu beachten, dass pro Haushalt eine eigene Datei vorhanden ist. Für jede Tagesmessreihe gibt es einen entsprechenden Eintrag, wie in Abbildung 1.1 als Baumdiagramm dargestellt. Die Detailmessung

## 1. Einleitung

und die Summenmessung pro Tag wurde auf Grund der vorher beschriebenen verschiedenen Messsysteme in zwei eigenen Einträgen abgelegt, die daher auch verschiedene Indizierungen der beiden Messungen erfordern. Innerhalb des Eintrages einer Tagesmessung gibt es drei Felder. Das erste Feld enthält die Spaltenüberschriften, das zweite Feld Datum und Uhrzeit und im dritten Feld werden in einem Array die Messwerte zu den Spaltenüberschriften für jeden Zeitpunkt abgelegt. In zukünftigen Beschreibungen wird ein Tagesindex verwendet, der einen Eintrag einer bestimmten Tagesmessreihe indiziert. Innerhalb dieser Tagesmessreihe wird weiters ein Spaltenindex angegeben, der auf eine Spalte im Überschriftenfeld sowie in den Messwerten verweist.

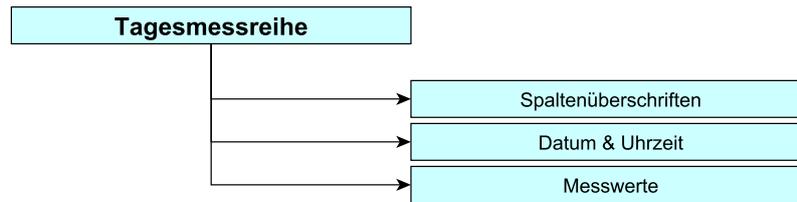


Abbildung 1.1.: Baumdiagramm der Rohdaten

## 1.2. Geräteklassen

In diesem Abschnitt wird auf die Unterteilung der vermessenen Geräte in verschiedene Klassen eingegangen. In der folgenden Aufzählung sind die entsprechend implementierten Geräteklassen mit ihrem in weiterer Folge verwendeten Geräteindex aufgelistet und jene Geräte angegeben, die in diese Klassen fallen. Aus Gründen der einfacheren Lesbarkeit der erstellten Geräteübersicht sind die Indices 1 und 2 nicht für Geräteklassen sondern für allgemeine Informationen zur jeweiligen Zeile der folgenden Tabelle bzw. der im nächsten Abschnitt beschriebenen Informationsdatenstruktur verwendet. Tabelle A.1 im Anhang gibt einen Überblick, in welchen Haushalten die entsprechenden Geräte gemessen wurden. In der letzten Zeile dieser Tabelle wird die Gesamtanzahl der vermessenen Geräte angegeben, wobei hier Sommer-, und Wintermessung extra gewertet wurden, da in manchen Haushalten bei den beiden Messreihen für Sommer und Winter unterschiedliche Geräte vermessen wurden.

Diese hier beschriebenen Klassen bilden auch die Grundlage der weiteren Auswertungen, da jede Klasse für sich auf ihre charakteristischen Eigenschaften untersucht wird. Weiters gibt es für jede Klasse verschiedene Analysen, die von höherem Interesse sind als andere.

1. (Haushaltsname)
2. (Sommer-, oder Wintermessung)

## 1. Einleitung

3. Kühlschrank: Kühlschränke und Kühl-, Gefrierkombigeräte
4. Waschmaschine
5. Gefrierschrank
6. Wäschetrockner
7. Föhn
8. Multimedia: TV, HiFi, SAT-Receiver, DVD-Player, Switch, Router, Radio, Laptop, PC, Drucker,...
9. Geschirrspüler
10. Toaster
11. Mikrowelle
12. Wasserkocher
13. Sonstiges: alle übrigen Geräte
14. Gesamt
15. Gesamt L1
16. Gesamt L2
17. Gesamt L3

### 1.3. Methodik

In dieser Arbeit werden die Rohdaten, die als MATLAB-Binärdaten zur Verfügung stehen in einem ersten Schritt analysiert und die relevanten Informationen in einer Informationsdatenstruktur zusammengefasst, um einen schnelleren und einfacheren Zugriff auf die benötigten Daten zu erlauben. Sobald das Grundgerüst dieser Struktur vorhanden ist, kann es mit den entsprechenden Daten gefüllt werden, die für die nachher durchgeführten Auswertungen benötigt werden.

## 2. Programmbeschreibung

In diesem Kapitel wird auf die Erstellung der benötigten Funktionalitäten in MATLAB eingegangen. Im Anhang B ist der Quellcode der wichtigsten Funktionen angeführt und diese sind durch entsprechende Kommentare selbsterklärend gestaltet.

Da bei den unterschiedlichen Auswertungen, die im Abschnitt 2.3 genauer beschrieben sind, jeweils nur ein Teil der gesamten verfügbaren Daten benötigt wird, wird wie im folgenden Abschnitt beschrieben, eine Informationsdatenstruktur erstellt, die einen einfacheren und vor allem schnelleren Zugriff auf die aktuell benötigten Daten bieten soll. Daher wird in diesem Kapitel zuerst auf die Erstellung dieser Datenstruktur eingegangen und im Anschluss werden die Auswertungen beschrieben. Bei der Erstellung der Informationsdatenstruktur wird bereits nach den Geräteeinsatzpunkten gesucht und die Informationen über den Geräteeinsatz gespeichert. Dabei wird darauf geachtet, dass einerseits alle benötigten Werte für die folgenden Auswertungen bereits gespeichert sind aber andererseits die gesamte Datenmenge in der Struktur so gering wie möglich gehalten wird, um ein schnelles Laden der Daten zu ermöglichen. Ohne vorherige Erstellung der Informationsdatenstruktur würde ein Aufruf einer einzelnen Auswertung sehr lange Rechenzeiten mit sich bringen.

Nachdem die Informationsdatenstruktur erstellt und mit den Informationen über die vermessenen Geräte gefüllt ist, können die gewünschten Auswertungen in schneller und einfacher Weise durchgeführt werden.

Um die Anpassung der gewünschten Auswertung an die eigenen Bedürfnisse so einfach wie möglich zu halten, wird weiters ein graphisches User Interface erstellt. Die graphische Ausgaben der entsprechenden Auswertungen bilden die Grundlage des Kapitels 4.

### 2.1. Informationsdatenstruktur erstellen

In einem ersten Schritt wird eine neue Informationsdatenstruktur erstellt, die später einen schnelleren Zugriff auf relevante Daten liefern soll. Daraus ergibt sich auch die Notwendigkeit, diese Informationsdatenstruktur laufend zu erweitern bzw. zu ergänzen. Die aktuell gespeicherten Informationen sind im Folgenden dargestellt. Die MATLAB-Funktion `build_infostructure()` (siehe B.1) erstellt genau die beschriebene Informationsdatenstruktur unter Verwendung der folgenden Hilfsfunktionen:

- `find_relevant_areas(...)`: Suche nach Ein-, und Ausschaltzeitpunkten der Geräte; die gefundenen Ein-, und Ausschaltzeitpunkte werden jeweils paarweise zu

## 2. Programmbeschreibung

einem so genannten Zyklus zusammengefasst und paarweise gespeichert. (siehe 2.2 sowie B.2)

- `calculate_overall_data(...)`: Berechnung der Summendaten pro Tag für ein bestimmtes Gerät
- `find_...(...)`: Suche nach diversen Strings im Spaltennamen um eine automatische Gerätezuordnung in die im Abschnitt 1.2 beschriebenen Gerätekategorien zu treffen.

Um bei Fehlern im Programm oder fehlerhaft erkannten Zyklen in einer Messreihe nur die fehlenden Daten berechnen zu müssen, werden die gefundenen Zyklen zuerst für jeden Haushalt einzeln mit allen nötigen Informationen gespeichert. Falls eine Haushaltsmessreihe neu berechnet werden muss, kann dies einfach durch Löschen der zugehörigen Datei und Aufruf der Funktion `build_infostructure()` geschehen. Die Funktion erkennt automatisch, dass für einen Haushalt die Zwischenspeicherung fehlt und ergänzt sie. Diese Vorgehensweise spart vor allem bei der Entwicklung der Funktionen Zeit beim Testen, da auch bei Fehlern während der Ausführung alle bereits vollständig berechneten Haushalte zwischengespeichert sind und nicht erneut berechnet werden müssen.

Die ermittelten Daten werden in einem MATLAB cell-array gespeichert, damit sie einerseits schnell und effizient mit MATLAB verarbeitet werden können und andererseits auch das händische Betrachten der Daten sehr einfach gehalten bleibt. Dabei werden pro Haushalt zwei Zeilen im Array erstellt, also eine für die Sommermessreihe und eine für die Wintermessreihe. Die Spalten des Arrays sind die in Abschnitt 1.2 beschriebenen Geräteklassen. In Abbildung 2.1 wird ein Eintrag in dieser Struktur mit allen enthaltenen Feldern dargestellt.

In dieser Struktur werden die folgenden relevanten Informationen gespeichert. Bei dieser Aufstellung gibt der Name in Klammer den korrespondierenden MATLAB Variablennamen in der Struktur selbst an. Für den Aufbau der Rohdaten als Grundlage einzelner Felder dieser Struktur siehe 1.1.

- Dateiname der Rohdaten (`filename`)
- Tagesindex innerhalb der Rohdaten (`day_index`)
- Spaltenindex innerhalb der Rohdaten (`column_index`)
- Geräteklasse (siehe zur Einteilung 1.2) (`geraete_type`)
- Phasenanschluss (`phase`)
- maximale Leistung (`power_max`): maximale Leistung über den gesamten Tag betrachtet
- durchschnittliche Leistung (`power_mean`): durchschnittliche Leistung über den gesamten Tag betrachtet

## 2. Programmbeschreibung

- Einschaltdauer (`on_time`): gesamte ermittelte Betriebsdauer des Gerätes
- Startindex (`start`): Zeitindizes des Beginns der Betriebszyklen des Gerätes
- Startzeitpunkte (`start_time`): Uhrzeiten des Beginns der Betriebszyklen des Gerätes
- Endindex (`ende`): Zeitindizes des Endes der Betriebszyklen des Gerätes
- Endzeitpunkte (`end_time`): Uhrzeiten des Endes der Betriebszyklen des Gerätes
- Einschaltdauer pro Zyklus (`on_times`)
- maximale EIN-Leistung (`on_power_max`): maximale Leistung des Gerätes während der Zyklen
- durchschnittliche EIN-Leistung (`on_power_mean`): durchschnittliche Leistung des Gerätes während der Zyklen
- minimale EIN-Leistung (`on_power_min`): minimale Leistung des Gerätes während der Zyklen
- Einschaltzyklen (`cycles`): Anzahl an Betriebszyklen
- Ausschaltdauer (`off_time`): Zeit, die das Gerät im Standby-Zustand oder ausgeschaltet verbringt
- durchschnittliche Einschaltdauer (`on_time_mean`)
- maximale Einschaltdauer (`on_time_max`)
- minimale Einschaltdauer (`on_time_min`)
- maximale Stand-By Leistung (`standby_max`)
- durchschnittliche Stand-By Leistung (`standby_mean`)

## 2. Programmbeschreibung

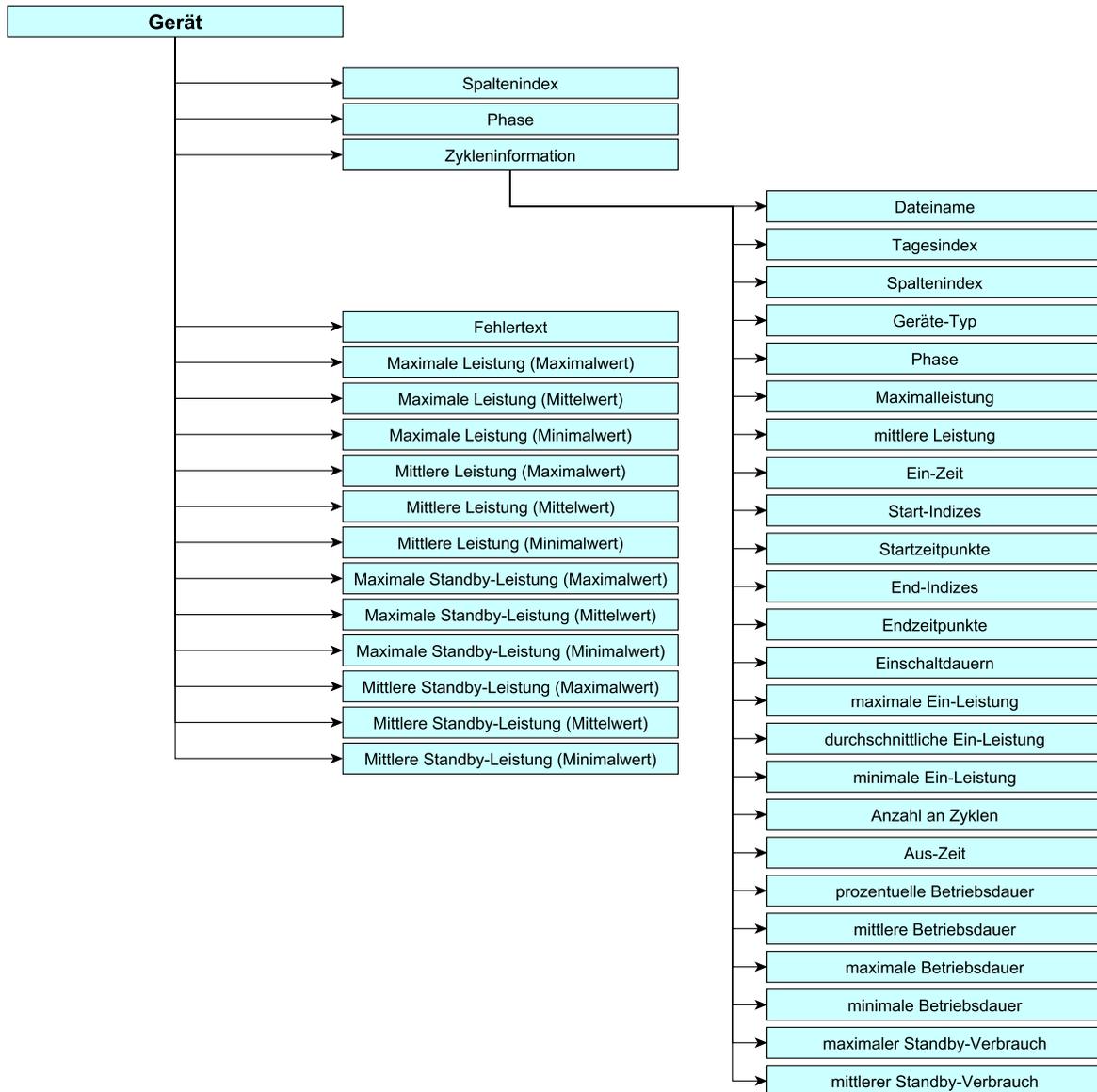


Abbildung 2.1.: Baumdiagramm der Informationsdatenstruktur geräte

## 2. Programmbeschreibung

Für die Auswertung des Energieverbrauchs sowie eine detaillierte Betrachtung der Leistungsverläufe während der Zyklen werden auch die Informationen über die Leistung zu jedem Zeitpunkt während des Betriebs des Gerätes benötigt. Da diese Daten zu einer großen Menge an zusätzlichen Werten führen und somit das Laden der Informationsdatenstruktur sehr lange brauchen würde, werden diese Informationen sinnvollerweise in einer eigenen Struktur abgespeichert. Diese Struktur wird auf mehrere Dateien, also eine für jede vorher beschriebene Geräteklasse, aufgeteilt. In dieser eigenen Struktur sind dann zusätzlich zu den relevanten Informationen wie oben beschrieben die Leistungsverläufe von allen einzelnen Zyklen enthalten. Hierbei wird im erstellten Array jede Zeile einem spezifischen Gerät zugeordnet und jede Spalte einem Tag innerhalb des Messzeitraumes an dem es entsprechende Zyklen für dieses Gerät gibt. Die Abbildung 2.2 zeigt die Informationen, die in den einzelnen Einträgen dieser Struktur abgelegt sind. Diese erweiterte Struktur wird ausgehend von der vorher beschriebenen Struktur `geraete` mit Hilfe der MATLAB Funktion `save_cycle_info(...)` für jedes Gerät einzeln erstellt. Diese Struktur wird weiters um Informationen zum Blindleistungsverlauf ergänzt.

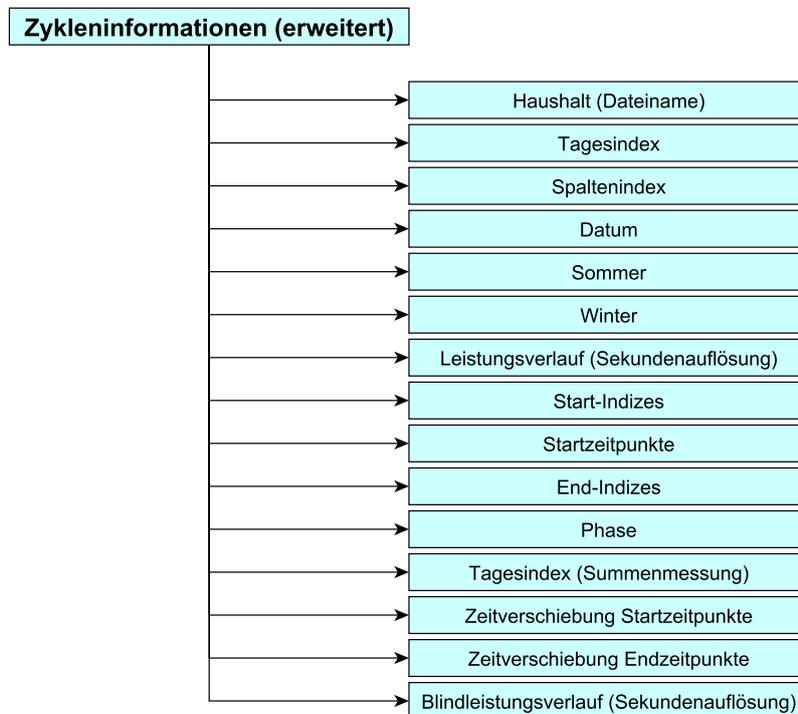


Abbildung 2.2.: Baumdiagramm der Informationsdatenstruktur `cycle_info`

In dieser zweiten Informationsdatenstruktur werden folgende Felder mit Daten bestückt:

- Dateiname der Rohdaten (`hh`)
- Tagesindex innerhalb der Rohdaten (`day_index`)

## 2. Programmbeschreibung

- Spaltenindex innerhalb der Rohdaten (`column_index`)
- Datum (`date`)
- Sommermessung (`sommer`): Gibt an, ob es sich um eine Sommermessreihe handelt
- Wintermessung (`winter`): Gibt an, ob es sich um eine Wintermessreihe handelt
- Leistungsverlauf während der Betriebszyklen (`power`)
- Startindex (`start`): Zeitindizes des Beginns der Betriebszyklen des Gerätes
- Startzeitpunkte (`start_time`): Uhrzeiten des Beginns der Betriebszyklen des Gerätes
- Endindex (`ende`): Zeitindizes des Endes der Betriebszyklen des Gerätes
- Endzeitpunkte (`end_time`): Uhrzeiten des Endes der Betriebszyklen des Gerätes
- Phasenanschluss (`phase`)
- Tagesindex der zugehörigen Summenmessung innerhalb der Rohdaten (`day_summe`)
- Zeitverschiebung zwischen Einzel-, und Summenmessung der Startzeitpunkte (siehe 4.3 für weitere Beschreibungen) (`time_offset_start`)
- Zeitverschiebung zwischen Einzel-, und Summenmessung der Endzeitpunkte (siehe 4.3 für weitere Beschreibungen) (`time_offset_ende`)
- Blindleistungsverlauf während der Betriebszyklen (`reactive_power`)

## 2. Programmbeschreibung

Um auch für die Haushaltsmessungen die benötigten Informationen schnell verfügbar zu haben, wurde hier eine weitere Struktur angelegt, in der diese Informationen abgespeichert sind. Die gespeicherten Werte hierbei sind neben dem Haushaltsnamen für die Rohdaten auch die Personengröße, die aus den zusätzlich zu den Messdaten verfügbaren Fragebögen herausgeht. Für die effiziente Speicherung der Daten wird wieder ein cell-Array verwendet. Jeder Eintrag entspricht dabei einem Haushalt und für einen Beispielhaushalt sind die gespeicherten Daten in Abbildung 2.3 zu sehen. Dabei wird in dem cell-Array `days` (Tagesinformationen) eine Zuordnung der Indizes zwischen Detailmessung und Summenmessung getroffen. Weiters wird das Datum des jeweiligen Messtages sowie einige charakteristische Werte für diesen Tag gespeichert. Es werden beispielsweise jeweils die minimalen, durchschnittlichen und maximalen Werte der Spannungs-, und Leistungsverläufe aller drei Phasen berechnet und zwischengespeichert. Diese Werte sollen einen Überblick über den Leistungsverbrauch eines Haushaltes geben. Weiters ist es so möglich, die Spannungsqualität über den Tag verteilt leicht zu betrachten.

## 2. Programmbeschreibung

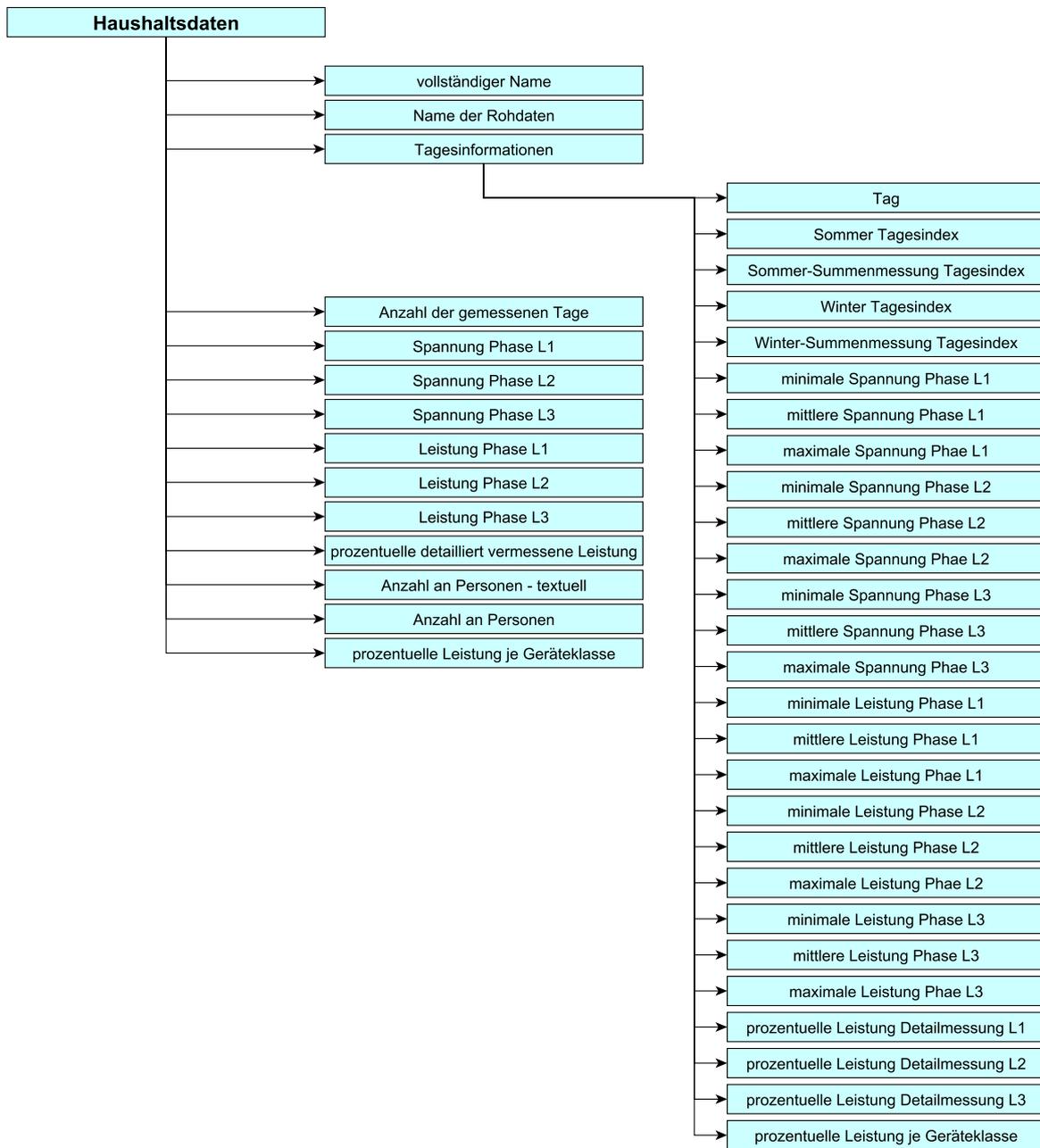


Abbildung 2.3.: Baumdiagramm der Informationsdatenstruktur hh\_data

## 2.2. Geräteeinsatzzeitpunkte suchen

Sobald das Grundgerüst der Informationsdatenstruktur erstellt ist, gilt es, dieses mit Informationen zu befüllen. Die wichtigste Unterscheidung hierbei ist, ob sich ein Gerät in Betrieb befindet oder ausgeschaltet bzw. im Standby-Zustand ist. Daher wird im nächsten Schritt nach den Einsatzzeitpunkten, also Beginn und Ende des Betriebes aller Geräte gesucht und diese Informationen dann in der Informationsdatenstruktur abgelegt. Dabei wurden je nach Gerät unterschiedliche Herangehensweisen gewählt, um optimale Suchergebnisse zu erhalten und möglichst wenig Fehlerkorrekturen zu benötigen.

Als Suchalgorithmus für die Einschaltzeit wurde eine einfache Funktion verwendet, die den aktuellen Leistungswert mit dem Wert vom vorherigen Zeitpunkt vergleicht und ab einem Unterschied vom Faktor 1.2 diesen Zeitpunkt als Beginn verwendet. Dies erlaubt es, kleine Leistungsunterschiede im Stand-By Betrieb nicht als Beginn des regulären Betriebs zu werten.

Für die Ausschaltzeit wurde je nach Geräteklasse ein anderer Algorithmus verwendet, da manche Geräte während des Betriebes keine konstante Leistungsaufnahme bzw relativ große Leistungssprünge aufweisen. Hier sei als Beispiel eine Waschmaschine genannt, die nach einer Aufheizphase mit durchschnittlich 2kW eine Schleuderphase mit deutlich geringerer Leistungsaufnahme von etwa 500W im Anschluss eine Abpumpphase mit ca 100W,... aufweist und daher während eines Waschganges Leistungssprünge aufweist, die als Ausschaltzeit fehlinterpretiert werden könnten.

Für die folgenden Geräte mit alternierender Leistungsaufnahme während des Betriebes wurde eine eigens entworfene Suchfunktion verwendet. Diese sucht ab dem Einschaltzeitpunkt + 3h (maximale Laufzeit der betrachteten Geräte) in Richtung Einschaltzeitpunkt nach einem Leistungssprung nach unten und nimmt diesen Sprung als Ausschaltzeitpunkt. Geräte mit dieser Suchfunktion sind Geschirrspüler, Waschmaschine, Wäschetrockner.

Für Bügeleisen und Toaster gilt eine deutlich kürzere Betriebszeit von maximal 30 Minuten. Um daher knapp aufeinander folgende Zyklen unterscheiden zu können, wurde die oben verwendete Funktion angepasst, und zwar so dass sie nur 0,5h zum Einschaltzeitpunkt addiert, aber sonst den gleichen Algorithmus verwendet.

Für alle anderen Geräte wurde vereinfacht der gleiche Algorithmus verwendet, der alle Messzeitpunkte durchgeht und nach einem Einschaltzeitpunkt den nächsten Leistungssprung nach unten als Ausschaltzeitpunkt verwendet. Dies liefert vor allem bei zyklischen Geräten oder Geräten mit annähernd konstanter Leistungsaufnahme während des Betriebes brauchbare Ergebnisse.

Während der Suche nach Einschaltzyklen sind folgende Probleme aufgetreten, für die eine geeignete Abhilfe gesucht werden musste:

- Einschaltzeitpunkt am Beginn der Messreihe; **Abhilfe:** Falls der erste Messwert größer als 5W ist, wird dies bereits als EIN interpretiert
- Ausschaltzeitpunkt liegt eigentlich erst am nächsten Tag; **Abhilfe:** Falls am Ende des Tages noch kein Ausschaltzeitpunkt gefunden wurde, setze den letzten Zeit-

punkt als AUS

- Zyklen diverser Geräte, die sehr lange dauern ( $> 15h$ ) und daher nicht realistisch sind; **Abhilfe:** Mit der Funktion `find_too_long_cycles(...)` wird nach solchen überlangen Zyklen gesucht und ein entsprechendes Flag in der Informationsdatenstruktur gesetzt, das die weitere Verwendung dieser Daten ausschließt.

Nach eingehender Untersuchung der Ergebnisse des oben beschriebenen Algorithmus für Waschmaschinen wurde festgestellt, dass hier doch eine hohe Anzahl an fehlerhaft erkannten Zyklen auftritt. Daher wurde ein neuer Algorithmus entwickelt, der für Waschmaschinen eingesetzt wurde. So wurde mit diesem Algorithmus nach dem Einschalten nach einer Pause im Leistungsverbrauch von mindestens 200 Sekunden gesucht und der Beginn dieser Pause als Endzeitpunkt gesetzt. Falls die Zyklendauer unter 50 Minuten liegt, wurde der Zyklus als falsch erkannt gewertet und nicht weiter betrachtet. Dies lieferte viel bessere Ergebnisse als die vorher verwendete Suchfunktion.

### 2.3. Funktionen für die Auswertungen

In diesem Abschnitt werden in aller Kürze die bei den Auswertungen aufgerufenen Funktionen mit den wichtigsten Eigenschaften beschrieben. Es wird nicht näher auf die tatsächliche Programmierung eingegangen, da vorausgesetzt wird, dass der Leser mit den hier beschriebenen Punkten in der Lage sein sollte, die durchgeführten Auswertungen auf einfache Weise nachzuvollziehen, allerdings das detaillierte Verständnis des Codes von zweitrangigem Interesse ist. Bei genauerer Betrachtung der jeweiligen Funktionen sollte es mit Erfahrungen mit MATLAB innerhalb kürzester Zeit möglich sein, den detaillierten Code nachzuvollziehen und bei Bedarf anpassen zu können. Daher wurde auch auf ausführliche Kommentare im Code geachtet. Der gesamte Quellcode der beschriebenen Funktionen ist im Anhang B angeführt.

#### 2.3.1. Einschaltzeitpunkt

Auf Basis der Informationen in der Informationsdatenstruktur `geraete` werden bei dieser Auswertung die Einschaltzeitpunkte aller Geräte der gewählten Geräteklasse ermittelt und die Anzahl der Geräte zu den einzelnen Zeitpunkten zusammengefasst und graphisch dargestellt.

Da alle benötigten Informationen bereits bei der Erstellung der Struktur ermittelt worden sind, ist hier nur mehr eine Filterung nach diversen Kriterien notwendig, wodurch die Rechenzeit dieser Auswertung sehr gering gehalten werden kann. Dabei kann neben der gewünschten Messreihe (also Sommer oder Winter) auch die gewünschten Haushalte, Wochentage sowie die zeitliche Auflösung der Darstellung frei gewählt werden.

## 2. Programmbeschreibung

Bei der Auswertung des Einschaltzeitpunktes für Einzelgeräte wird die Funktion `function ret = show_usage_histogram(geraete, hh, sommer, winter, geraete_type, weekdays, aufloesung, do_plots)` verwendet. Wie schon an der Deklaration der Funktion ersichtlich, werden hier als erster Parameter die Gerätedaten übergeben. Als zweiter Parameter wird eine Liste der gewünschten Haushalte übergeben. Die nächsten beiden Parameter sind Boolean-Variablen für die Auswahl von Sommer-, und Wintermessreihen, der gewünschte Geräte-Typ wird als nächster Parameter als Integer übergeben. Die Zuordnung zwischen Geräteklassen und Index ist im Abschnitt 1.2 zu finden. Die gewünschten Wochentage werden in einer Liste mit den englischsprachigen Abkürzungen (3 Buchstaben) übergeben. Der Parameter `aufloesung` gibt die zeitliche Auflösung der gewünschten Analyse an, wobei hier der Wert 1 für 1 Tag steht und somit zum Beispiel eine Auflösung von einer Stunde mit dem Wert  $1/24$  angegeben werden kann. Der letzte Parameter `do_plots` erlaubt es auf einfache Weise, die grafische Ausgabe dieser Funktion zu unterdrücken. Dies ist besonders bei der Weiterverwendung der Rückgabewerte interessant. Die Funktion geht alle gefundenen Zyklen in der Informationsdatenstruktur durch und filtert gemäß den übergebenen Parametern. Weiters werden Zyklen, die mit dem `exclude`-Flag gekennzeichnet sind nicht beachtet. Für alle Zyklen werden dann die Beginnzeiten in ein Array gespeichert und als Rückgabewert der Funktion erhält man zusätzlich zu dieser Liste auch die Rückgabewerte der MATLAB-Funktion `hist()`, die eine einfache Anpassung der Ausgabegrafiken an die eigenen Anforderungen erlauben. Außerdem plottet die Funktion ein Histogramm mit der gewünschten zeitlichen Auflösung inklusive aller Achsbeschriftungen. Es wird weiters versucht eine Wahrscheinlichkeitsdichtefunktion zu schätzen, die dieser Verteilung entspricht. Das Ergebnis dieser Schätzung wird in einem normierten Plot dargestellt und als Rückgabewert einerseits als MATLAB-eigener Variablentyp `prob.KernelDistribution` und andererseits als Parameterliste zur Verfügung gestellt. Um den Aufruf der Funktion ohne dem GUI zu erleichtern, wird die Informationsdatenstruktur automatisch nachgeladen, falls der erste Parameter mit 0 übergeben wird. Der gesamte Quellcode dieser Funktion ist im Anhang B.3 angegeben.

### 2.3.2. Einschaltdauer

Auch hier basiert die Auswertung auf der Informationsdatenstruktur `geraete`. Es werden für das gewählte Gerät die Einschaltdauern der jeweiligen Zyklen ermittelt und wieder zusammengefasst und graphisch dargestellt. Dabei sind wieder die gleichen Auswahlmöglichkeiten wie vorher beschrieben vorhanden.

Bei der Auswertung der Einschaltdauer der Geräteklassen wird die Funktion `function ret = show_usage_duration(geraete, hh, sommer, winter, geraete_type, weekdays, aufloesung, xmin, xmax)` verwendet. Die ersten Parameter haben wieder die gleiche Reihenfolge und Bedeutung wie im Abschnitt 2.3.1 beschrieben. Die beiden zusätzlichen

## 2. Programmbeschreibung

Parameter `xmin` und `xmax` geben den Bereich an, in dem der Plot dargestellt werden soll. Weiters wird bei der Auswertung die durchschnittliche und die maximale Einschaltdauer ermittelt. Auf Grund vieler langer Zyklen bei der Auswertung wird eine zusätzliche Begrenzung eingebaut, die alle Zyklen über 6 Stunden Laufzeit ausklammert und erneut einen Plot generiert und die Durchschnitts-, und Maximalwerte berechnet. Der gesamte Quellcode ist im Anhang B.4 abgedruckt.

### 2.3.3. Einschalthäufigkeit / Tag

Auch hier wird die Informationsdatenstruktur `geraete` zur Auswertung herangezogen. Es wird für jeden Tag einzeln ermittelt, wie oft sich das jeweilige Gerät an diesem Tag eingeschaltet hat. Die ermittelten Daten werden wieder gruppiert und graphisch dargestellt. Auch hier gibt es wieder die Auswahlmöglichkeiten wie vorher beschrieben.

Hier kommt die Funktion `function ret = show_usage_count(geraete, hh, sommer, winter, geraete_type, weekdays, do_plots)` Auch diese Funktion hat die gleiche Parametergestaltung und Reihenfolge wie die im Abschnitt 2.3.1 beschriebene. Einzig der Parameter `aufloesung` ist hier nicht verfügbar, da er nicht weiter relevant ist. Der Parameter `do_plots` unterdrückt die grafische Ausgabe und die Funktion liefert somit die berechneten Werte nur als Rückgabewert. Auch hier wurde wieder ein zweiter Plot mit entsprechender Begrenzung und zwar auf 100 Zyklen eingeführt. Rückgabewerte sind wieder alle berechneten Werte. Der vollständige Code ist im Anhang B.5 abgedruckt.

### 2.3.4. Nennleistung (Durchschnittswert)

Auch die Nennleistung ist bereits in der Struktur `geraete` gespeichert. Es wird daher der vorher ermittelte Durchschnittswert der Leistungsaufnahmen zur Auswertung herangezogen.

Hier kommt die Funktion `function ret = show_power_consumption(geraete, hh, sommer, winter, geraete_type, weekdays, aufloesung, max_mean)` zum Einsatz. Auch diese Funktion hat die gleiche Parametergestaltung und Reihenfolge wie die im Abschnitt 2.3.1 beschriebene. Der Parameter `aufloesung` gibt die entsprechende Klasseneinteilung der Nennleistung für das Histogramm an. Rückgabewerte sind wieder alle berechneten Werte. Der zusätzliche Parameter `max_mean` lässt eine Auswahl der Berechnungsmethode für die Nennleistung zu. Es wird hierbei entweder die maximale oder die durchschnittliche Leistung während des Zyklus verwendet. Je nach untersuchtem Gerät liefert eine der beiden Methoden bessere Ergebnisse. Bei dieser Auswertung wird `max_mean` auf `'mean'` gesetzt. Der vollständige Code ist im Anhang B.6 abgedruckt.

### 2.3.5. Nennleistung (Maximalwert)

Diese Auswertung verwendet die gleiche Funktion wie im Abschnitt 2.3.4 beschrieben. Hier wird nur der Parameter `max_mean` auf `'max'` gesetzt.

### 2.3.6. Energieverbrauch

Um den Energieverbrauch des jeweiligen Gerätes ermitteln zu können, muss der Leistungsverlauf während des Zyklus bekannt sein. Mit dieser Information kann dann der Energieverbrauch pro Zyklus ermittelt werden, indem während eines Zyklus die Summe der Leistung berechnet wird. Da die zeitliche Auflösung der Messreihe in Sekunden ist, erhält man damit den Energieverbrauch in Ws. Je nach Wunsch wird im Anschluss der mittlere Energieverbrauch entweder pro Zyklus, pro Tag oder über pro Jahr berechnet.

Diese Auswertung verwendet die Funktion `function ret = show_energy_consumption(~, hh, sommer, winter, geraete_type, weekdays, type)`, wobei hier der normalerweise erste Parameter `geraete` nicht benötigt wird und nur aus dem Grund eingeführt wird, damit die restliche Parameterreihenfolge ident mit den vorher beschriebenen Funktionen bleibt. Der zusätzliche Parameter `type` lässt eine Auswahl des Betrachtungszeitraumes für den Energieverbrauch zu, da für verschiedene Geräte unterschiedliche Betrachtungen in der Literatur gebräuchlich sind. So ist die Übergabe des Wertes 1 beispielsweise für eine zyklweise Betrachtung nötig, wie bei Waschmaschinen oder Geschirrspülern üblich. Die Angabe von 2 erlaubt eine tageweise Betrachtung, der Wert 3 eine Betrachtung über ein ganzes Jahr, wie es für Kühlgeräte üblich ist. Die gesamte Funktion ist im Abschnitt B.7 abgedruckt.

### 2.3.7. Leistungsverlauf

Hier soll der Leistungsverlauf während eines Zyklus betrachtet werden. Es wird entweder die Wirk-, die Blindleistung oder beide dargestellt.

Bei dieser Auswertung wird die Funktion `function ret = show_cycles(~, hh, sommer, winter, geraete_type, weekdays, cycle_count, pq, filename)` verwendet. Die Parameter `hh, sommer, winter, geraete_type` sowie `weekdays` wurden in den vorher beschriebenen Funktionen ebenfalls verwendet. Neu ist hier der Parameter `cycle_count`, mit dem sich die Anzahl der gewünschten Zyklen im erstellten Plot auswählen lässt. Übergibt man hier eine Zahl, so wird diese Anzahl zufällig aus allen Zyklen gewählt. Übergibt man den String `'all'`, so werden alle Zyklen in der Grafik dargestellt. Mit dem Parameter `pq` kann man wählen, ob der Wirkleistungsverlauf (Übergabe von `'p'`),

## 2. Programmbeschreibung

der Blindleistungsverlauf ('q') oder beide Verläufe ('pq') dargestellt werden sollen. Der Parameter `filename` wird für das aufgerufene GUI `comparison` verwendet, das ein einfaches Auswählen der gewünschten Linien sowie eine Eingrenzung des Anzeigebereiches erlaubt. Weiters gibt es einen Button, der den Graphen mit dem übergebenen Namen und einer fortlaufenden Nummerierung abspeichert. Der komplette Quellcode dieser Funktion ist im Anhang B.8 zu finden.

### 2.3.8. Tagesleistungsverlauf

Diese Auswertung berechnet den mittleren Leistungsverlauf pro Person über einen Tag verteilt. Dazu werden die Verläufe der einzelnen Zyklen mit der entsprechenden Beginnzeit summiert und dann zuerst für jeden Haushalt durch die Personengröße dividiert. Da in der Literatur meist eine zeitliche Auflösung von einer Viertelstunde oder einer Stunde dargestellt wird, wird im Anschluss noch eine Glättungsfunktion angewandt.

Für diese Auswertung wird die Funktion `ret = show_mean_power(~, hh, sommer, winter, geraete_type, weekdays)` verwendet. Da die Informationsdatenstruktur `geraete` in dieser Funktion nicht benötigt wird, der Aufruf aber von der Reihenfolge der Parameter weiterhin konsistent sein soll, wird der erste Parameter nicht verwendet. Die restlichen Parameter sind wie schon vorher beschrieben zu verwenden. Diese Funktion lädt automatisch die Informationsdatenstruktur mit den Informationen zu den einzelnen Zyklen für das jeweilige Gerät nach, was einen Augenblick dauern kann. Danach werden die Mittelwerte aller gewünschten Haushalte, Tage und Messreihen ermittelt und grafisch dargestellt. Hierbei wird in der ersten Grafik auf Sekundenbasis, in der zweiten Grafik auf Minutenbasis und in der dritten Grafik auf Viertelstundenbasis geglättet. Dafür wird die MATLAB-eigene Funktion `decimate` verwendet. Der gesamte Quellcode ist wieder im Anhang B.9 zu finden.

### 2.3.9. Vergleich der Einschaltzeitpunkt

Diese Auswertung stellt die Einschalthäufigkeit für ein gewähltes Gerät für die gewünschten Wochentage und Messreihen gegenüber und erlaubt somit einen einfachen Vergleich im Benutzungsverhalten von verschiedenen Tagestypen. Es wird hier zur Anzeige wieder das GUI `comparison` verwendet. Alle nicht benötigten Linien können wieder über die eigens angebrachten Checkboxen am Rand des Plots ausgeschaltet werden.

Die verwendete Funktion `function ret=show_usage_comparison(geraete, hh, sommer, winter, geraete_type, weekdays, aufloesung)` greift auf die im Abschnitt 2.3.1 beschriebene Funktion zurück, wobei hier die grafische Ausgabe dieser Funktion unter-

drückt wird. Daher werden bei dieser Funktion auch die gleichen Parameter verwendet. Der gesamte Code ist im Anhang B.10 angeführt.

### 2.3.10. Vergleich der Einschalthäufigkeit / Tag

Diese Auswertung betrachtet die Verwendungshäufigkeit des gewählten Gerätes für verschiedene Tagestypen und die beiden Messreihen getrennt und stellt die Ergebnisse für jeden Tag einzeln dar. Weiters wird der Mittelwert der Wochentage und Wochenenden sowie der gesamte Mittelwert für beide Messreihen separat gebildet und ebenfalls dargestellt.

Die verwendete Funktion `function ret=show_usage_comparison_day(geraete, hh, sommer, winter, geraete_type, weekdays)` greift auf die im Abschnitt 2.3.3 beschriebene Funktion zurück, wobei hier die grafische Ausgabe dieser Funktion unterdrückt wird. Daher werden bei dieser Funktion auch die gleichen Parameter verwendet. Der gesamte Code ist im Anhang B.11 angeführt.

### 2.3.11. Haushaltslastgang

Diese Auswertung betrachtet im Unterschied zu den vorher beschriebenen keine Einzelgeräte sondern die Summenmessungen der jeweiligen Haushalte. Es wird der mittlere Lastgang der Wirk-, Blindleistung oder beider berechnet und daraus der mittlere Leistungsfaktor, falls gewünscht. Ebenfalls wird ein Standardlastprofil als Vergleich dargestellt.

Die Funktion `function show_load_profile(hh, sommer, winter, weekdays, detail_plot, mean_plot, sum_plot, pq)` wertet die entsprechenden Daten über die Haushalte aus. Die Parameter `hh, sommer, winter, weekdays` haben wieder die gleiche Bedeutung wie im Abschnitt 2.3.1 beschrieben. Der Parameter `detail_plot` erlaubt es, die einzelnen Lastgänge in einem Diagramm darzustellen. Der Parameter `mean_plot` stellt den Mittelwert der Lastgänge der gewählten Haushalte dar, der Parameter `sum_plot` die Summe der Lastgänge. Da sich diese beiden Plots nur in der Skalierung unterscheiden, ist jeweils nur einer von ihnen relevant. Der letzte Parameter `pq` erlaubt eine Auswahl der betrachteten Leistung. Die Übergabe von `'p'` betrachtet die Wirkleistung, wie beim H0 Profil üblich, `'q'` betrachtet die Blindleistung. Der Verlauf der Blindleistung ist in der Literatur nur selten zu finden und daher von besonderem Interesse. Die Übergabe von `'pf'` berechnet den Leistungsfaktor und stellt diesen grafisch dar. Weiters wird, falls gewünscht auch ein Vergleichs-H0-Profil dargestellt. Der gesamte Code dieser Funktion ist im Anhang B.12

### 2.3.12. Restprofil

Bei dieser Auswertung wird der hier als Restprofil bezeichnete Tagesleistungsverlauf untersucht. Das Restprofil bezeichnet hierbei das Haushaltslastprofil abzüglich aller im Detail vermessenen Geräte. Dabei wird jeweils der Verlauf des Restprofils über einen Tag dargestellt. Zur Ermittlung des Restprofils siehe Abschnitt 4.3.

Die Funktion `function ret = show_restprofil2(hh,sommer,winter,weekdays,num_days)` zeigt für den gewünschten Haushalt eine vorher festgelegte Anzahl an Tageslastgängen an. Diese Anzahl kann mit dem Parameter `num_days` angegeben werden. Hierbei gibt der String `'all'` an, dass alle verfügbaren Tage geplottet werden. Anderenfalls wird die gewünschte Anzahl zufällig aus allen Tagen gewählt. Zu Beachten ist hierbei, dass hier erstmals nur ein Haushalt übergeben werden darf, da auch eine Darstellung des Mittelwertes mehrerer Haushalte nicht sinnvoll erscheint. Der gesamte Code dieser Funktion ist im Abschnitt B.13 angeführt.

## 2.4. Grafisches User-Interface (GUI)

Um die Generierung von gewünschten Auswertungen auf Basis des ADRES-Datensatzes zu erleichtern, wurde in MATLAB ein graphisches User-Interface (kurz GUI) entwickelt. Dieses GUI bietet Zugriff auf die wichtigsten Auswertungen und erleichtert es somit, die im Abschnitt 4 beschriebenen Grafiken nachzubilden und natürlich auch, eigene Auswertungen zu erstellen. Im GUI Hauptfenster ist es möglich, diverse Parameter der Auswertungen einzustellen. Dieses Fenster ist in Abbildung 2.4 dargestellt und die blau nummerierten Bereiche korrespondieren mit den unten beschriebenen Einstellmöglichkeiten. Weiters geben sie die optimale Reihenfolge beim Einstellen diverser Parameter an.

Im obersten Auswahlfeld (1) wählt man aus, welche Auswertung man machen möchte. Je nach Auswahl stehen weitere Parameter bereit. Zur Verfügung stehen folgende Auswertungen:

- **Einschaltzeitpunkt:** Wertet die Einschaltzeitpunkte der einzelnen Geräte der gewählten Geräteklasse aus.
- **Einschaltdauer:** Wertet die Einschaltdauer der einzelnen Zyklen der gewählten Geräteklasse aus.
- **Einschalhäufigkeit / Tag:** Wertet die Häufigkeit des Einschaltens pro Tag der gewählten Geräteklasse aus.

## 2. Programmbeschreibung

- **Nennleistung (Durchschnittswert):** Wertet die durchschnittlich verbrauchte Leistung der gewählten Geräteklasse aus.
- **Nennleistung (Maximalwert):** Wertet die maximal verbrauchte Leistung der gewählten Geräteklasse aus.
- **Energieverbrauch:** Wertet den Energieverbrauch der gewählten Geräteklasse aus. Hierbei ist weiters eine Auswahl des Betrachtungszeitraumes vorzunehmen.
- **Leistungsverlauf:** Wertet den Leistungsverlauf während eines Zyklus aus.
- **Tagesleistungsverlauf:** Wertet die mittlere Tagesleistung der gewählten Geräteklasse pro Person aus.
- **Vergleich der Einschaltzeitpunkte:** Wertet die Einschaltzeitpunkte für jeden Tag getrennt aus.
- **Vergleich der Einschalthäufigkeit:** Wertet die Einschalthäufigkeit pro Tag für jeden Tag getrennt aus.
- **Haushaltlastgang:** Wertet den Haushaltlastgang der gewählten Haushalte im Vergleich zum H0-Profil aus.
- **Restprofil:** Wertet das Restprofil für einen gewählten Haushalt aus.

Im darunter befindlichen Bereich (2) legt man die gewünschten Haushalte für die folgende Auswertung fest. Man kann dies auf zwei Weisen erledigen. Entweder durch Setzen des dazugehörigen Häkchens oder durch Auswahl in der darüber befindlichen Liste. So kann man zum Beispiel auch 5 zufällige Haushalte wählen.

Im Bereich daneben (3) wählt man die gewünschten Messreihen (Sommer und Winter) mit Checkboxen aus.

Auch für die Wochentage gibt es zwei Möglichkeiten der Auswahl (4). Einerseits kann man wieder die gewünschten Wochentage durch Setzen des Häkchens wählen oder in der Liste oberhalb zum Beispiel „Wochenende“ oder „Werkstage“ auswählen.

Weitere Auswahlmöglichkeiten wie das zu untersuchende Gerät (5) oder die gewünschte Auflösung der Grafiken (6) sind je nach gewählter Analyse verfügbar. Auch die Auswahl der Abstufungen für das Histogramm sind bei Bedarf im Bereich (6) verfügbar. Dieser Bereich ändert je nach gewählter Analyse seine Auswahlmöglichkeiten

Der Bereich darunter (7) erlaubt eine Speicherung der Messwerte als MATLAB-Binärdatei sowie bei Vorhandensein der MATLAB-Funktion `matlabfrag` (Beziehbar unter [16] und optimal geeignet zum Erstellen von Grafiken im EPS-Format und anschließender Weiterverarbeitung in  $\text{\LaTeX}$ ) auch die Speicherung der produzierten Grafiken

## *2. Programmbeschreibung*

mit dem angegebenen Dateinamen. Weiters wird ein Export der dargestellten Daten nach Excel angeboten. Vorhandene Dateien werden ohne Nachfrage überschrieben.

Sobald alle Einstellungen vorgenommen sind, kann die Auswertung durch Drücken des Buttons (8) gestartet werden. Abhängig vom verwendeten Computersystem und der gewählten Analyse kann der Rechengang einige Zeit in Anspruch nehmen.

## 2. Programmbeschreibung

The image shows a graphical user interface (GUI) for data evaluation. It consists of several panels and controls:

- 1:** A red header bar with the text "Auswertung" and a dropdown menu labeled "Einschaltzeitpunkt".
- 2:** A panel titled "Haushalte" with a dropdown menu set to "Alle". Below it is a list of 43 addresses (ADRES001 to ADRES043) arranged in two columns, each with a checked checkbox.
- 3:** A panel titled "Messungen" with two checked checkboxes: "Sommer" and "Winter".
- 4:** A panel titled "Wochentage" with a dropdown menu set to "Alle" and a list of seven days (Montag to Sonntag), each with a checked checkbox.
- 5:** A panel titled "Gerät" with a dropdown menu set to "Kühlschrank".
- 6:** A panel titled "Anzahl Zyklen" with a dropdown menu set to "1 Zyklus" and a panel titled "Wirk-, Blindleistung" with a dropdown menu set to "P".
- 7:** A panel titled "Ergebnisse speichern?" with a text input field containing "result.mat" and three checkboxes: "Matlab (.mat)" (checked), "Grafiken (.eps)" (checked), and "Excel (.xls)" (unchecked).
- 8:** A large button labeled "Auswertung starten".

Abbildung 2.4.: Grafisches User-Interface (GUI)

### 3. Literaturrecherche

In diesem Kapitel wird auf vorangegangene Untersuchungen auf dem Gebiet dieser Arbeit eingegangen und weiters werden einige bereits mit dem hier verwendeten Datensatz erstellten Auswertungen und Arbeiten beschrieben.

Unter [6] ist der publizierte Endbericht des Forschungsprojektes „ADRES“ zu finden, in dem bereits auf die statistische Auswertung einiger ermittelter Größen eingegangen wird. Hier wird allerdings im Gegensatz zu dieser Arbeit ein Einsparungspotential von elektrischer Energie ermittelt und in Richtung Demand Side Management gearbeitet.

In [9] wird für verschiedene Geräteklassen, die auch hier verwendet werden, der mittlere Leistungsverlauf über einen Tag pro Person beschrieben. So wird in dieser Arbeit beispielsweise der Tageslastgang für den Sektor TV-Audio, der hier näherungsweise der Gerätekategorie Multimedia entspricht, aus verschiedenen statistisch ermittelten Parametern berechnet. Ein wesentlicher Unterschied hierbei ist, dass in dieser Arbeit der Sektor Büro extra behandelt wurde, die Geräte in diesem Sektor aber hier ebenfalls der Klasse Multimedia zugeordnet wurden. Dabei wird der Tagesverlauf der TV- und Radionutzung (ermittelt von der ORF Medienforschung, siehe [14] sowie [15]) und die durchschnittliche Leistung dieser Geräte zur Ermittlung dieses Verlaufes herangezogen.

Auch in anderen Arbeiten auf dem Gebiet des Demand Side Management, wie etwa [17], werden Tageslastganglinien für beispielsweise die Sektoren Waschen sowie Trocknen, welche hier den Geräteklassen Waschmaschine sowie Wäschetrockner entsprechen, angegeben.

Die Arbeit unter [7] beschäftigt sich unter anderem beispielsweise mit der Auswertung der Fragebögen, die im Zuge der Messkampagne ebenfalls erhoben wurden. Daraus lässt sich die Verteilung einzelner Geräte auf die entsprechenden Energieeffizienzklassen ableiten, was die Validierung des hier berechneten Energieverbrauchs erleichtert.

Unter [2] wird eine andere Herangehensweise zur Ermittlung der Betriebszyklen sowie der Geräteunterteilung beschrieben. In dieser Arbeit wird ausgehend von einem Summenleistungsverlauf eine Zuordnung der Verbrauchsänderungen zu einzelnen Geräten untersucht. Dabei wird der hier beschriebene Datensatz zur Validierung der Funktionalitäten verwendet. Es werden einzelne Geräte aus der Summenmessung erkannt und dann die erkannten Geräte mit den dazugehörigen Einzelgerätemessungen verglichen, um einen die Richtigkeit der erkannten Zyklen zu überprüfen. Die Schwierigkeiten in dieser Arbeit waren unter anderem, dass über den Leistungsverlauf von einzelnen Geräten, wie

### *3. Literaturrecherche*

Waschmaschinen, relativ wenig bekannt ist und daher ein Algorithmus zur Erkennung eines solchen Verlaufes aus der Summenmessung sehr schwierig zu implementieren ist. Bei Geräten, dessen Verlauf relativ genau bekannt ist, wie Kühl-, oder Gefriergeräte im Gegensatz führt diese Herangehensweise mit vertretbarem Aufwand zu brauchbaren Ergebnissen.

## 4. Auswertungen

In diesem Kapitel werden einige repräsentative Auswertungen basierend auf dem ADRES-Datensatz beschrieben, weitere Auswertungen können mit dem im Kapitel 2.4 beschriebenen GUI und den zugrunde liegenden Funktionen jederzeit auf benutzerfreundlichem Wege erstellt werden.

Zu Beginn wird der Einsatz einzelner Geräte an verschiedenen Tagen (Werktag, Wochenende) untersucht und die Ergebnisse mit den erwarteten beziehungsweise in anderer Literatur beschriebenen Merkmalen verglichen und die eventuellen Unterschiede erläutert.

Im zweiten Abschnitt werden die gesamten Haushaltsmessungen untersucht und auch hier wird wieder ein Vergleich mit einem Standardlastprofil durchgeführt und auf die Gemeinsamkeiten bzw. Unterschiede eingegangen. Der Unterschied zwischen Werktagen und Wochenenden soll ebenso untersucht werden wie der Einfluss von Sommer bzw. Winter. Ein weiteres Augenmerk wird auch auf die Untersuchung des Blindleistungsverbrauchs gelegt, da dieser Aspekt bis jetzt in der Literatur kaum beschrieben wurde, aber die Veränderung der Lastimpedanz und damit des Leistungsfaktors einen Einfluss auf die Stabilität von Inselnetzen haben kann.

Im dritten Abschnitt wird versucht, aus den Gesamtmessungen einzelne Geräte auszunehmen bzw. aus dem Restprofil (also die Gesamtmessung abzüglich aller Detailmessungen) weitere Erkenntnisse über nicht vermessene Geräte in den einzelnen Haushalten zu gewinnen. Dieser Punkt stellt eine besondere Herausforderung dar, da kaum Informationen über weitere Geräte, insbesondere leistungsstarke Verbraucher wie Elektroheizungen, Warmwasserboiler oder elektrische Durchlauferhitzer in den Haushalten vorliegen, die ein Auffinden dieser Geräte erleichtern würden.

### 4.1. Geräteinsatz

In diesem Abschnitt wird der Einsatz von diversen charakteristischen Geräteklassen untersucht und die Ergebnisse kritisch auf Plausibilität hinterfragt. So wurde bei dieser Analyse zum Beispiel festgestellt, dass einzelne Messungen über mehrere Tage eine konstante Leistungsaufnahme angeben. Diese Messungen wurden dann in der Informationsstruktur mit einem Flag gekennzeichnet und werden bei der weiteren Auswertung nicht mehr herangezogen.

Die Gliederung ergibt sich aus den im Abschnitt 1.2 beschriebenen Unterteilungen, wobei je nach Gerät auf unterschiedliche Gesichtspunkte geachtet wurde. Es werden für die betrachteten Geräte jeweils eine Auswahl der vorher beschriebenen tatsächlichen Analysemöglichkeiten herangezogen und diese dargestellt und beschrieben. So ist bei einigen Geräten zum Beispiel der Leistungsverlauf während des Betriebes von untergeordnetem Interesse, da er ohnehin während der Betriebsdauer als konstant angenommen werden kann. Genau diese Auswertung kann allerdings für andere Geräte von sehr hohem Interesse sein, da über diesen Verlauf gerade bei Geräten mit vordefinierten Programmen recht wenig Informationen bekannt und verfügbar sind.

Obwohl die Messreihe sehr umfangreich ist, aber für detaillierte Auswertungen einzelner Geräte auf Grund der unterschiedlichen vermessenen Geräte in den Haushalten der Stichprobenumfang leider nicht sehr groß ist, können manche Auswertungen durchaus Ergebnisse liefern, die auf den ersten Blick fragwürdig erscheinen. Hier sei vor allem der Unterschied von Einsatzzeitpunkten an den verschiedenen Tagen zu nennen. Gerade bei Geräten, wo dieser Unterschied interessant wäre, sind die Auswertungen sehr schwierig zu deuten, da nur mehr sehr wenige Messreihen an den jeweiligen Tagen zur Verfügung stehen.

## 4. Auswertungen

### 4.1.1. zyklische Geräte

Hier wird vor allem auf Geräte mit zyklischem Betrieb eingegangen. In diese Kategorie fallen vor allem Kühl-, und Gefriergeräte sowie Geräte zur Raumheizung und Warmwasserbereitung wobei die letzten beiden Gerätetypen auf Grund des meist fixen Anschlusses in dieser Messreihe nicht vermessen werden konnten.

In Abbildung 4.1 wird die Einsatzhäufigkeit für Kühlschränke dargestellt. Man erwartet von solchen Geräten eine annähernde Gleichverteilung der Einschalthäufigkeit, da diese Geräte temperaturgesteuert und somit unabhängig voneinander je nach Innentemperatur einschalten. Allerdings ist die Einschalthäufigkeit natürlich auch davon abhängig, ob die Kühlschranktür geöffnet wird oder nicht und daher erkennt man in den Morgenstunden (ca. 07:00 Uhr) sowie um die Mittagszeit (zwischen 12:00 und 13:00 Uhr) und abends (um ca. 18:00) einen leichten Anstieg der Einsatzhäufigkeit.

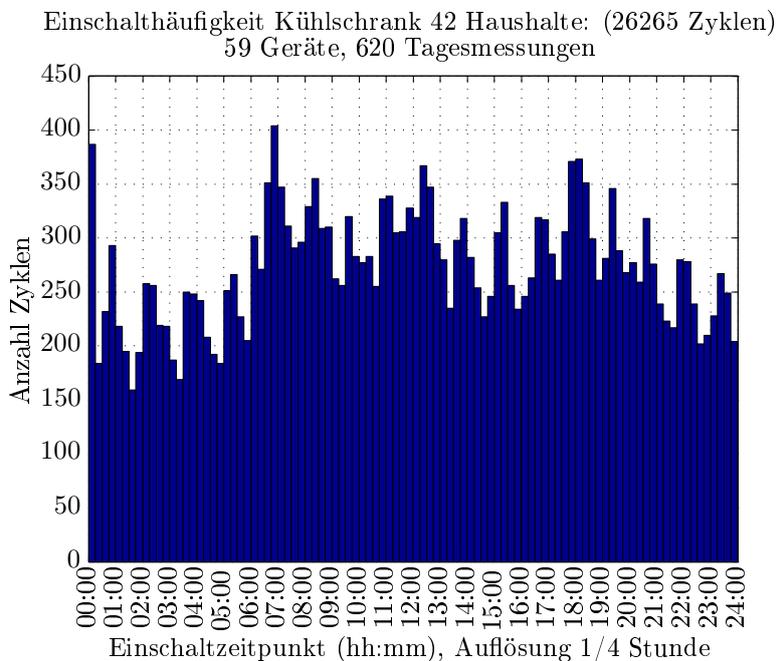


Abbildung 4.1.: Einschalthäufigkeit für Kühlschränke

Wie aus Abbildung 4.2 ersichtlich, hat die Einschaltzeit bei Kühlschränken ein deutliches Maximum bei 10 Minuten und variiert nur geringfügig um diesem Wert. Die durchschnittliche Einschaltzeit aller Zyklen berechnet sich zu 10 Minuten und 19 Sekunden, die maximale Einschaltzeit ist mit etwas über 15 Stunden noch deutlich zu hoch und deutet auf eine falsche Messung hin, die durch die automatische Fehlerkorrektur nicht ausgeschlossen wurde.

## 4. Auswertungen

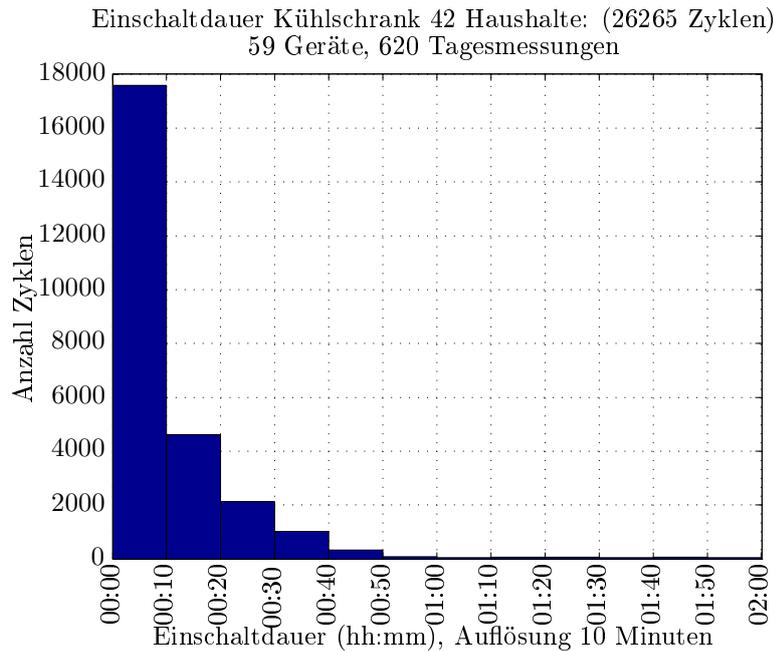


Abbildung 4.2.: Einschaltdauer für Kühlschränke

Für die Leistungsaufnahme erkennbar in Abbildung 4.3 ergibt sich ein Mittelwert von 88.8 W, ein Maximalwert von 1131.2 W, der auf einen Fehler in der Messung hindeutet, da ein so großer Leistungswert für Kühlschränke eindeutig zu hoch ist und eine minimale Leistung von 29.2 W, welche selbst für ein neues Gerät der Energieeffizienzklasse A++ zu niedrig scheint.

In der folgenden Darstellung (Abbildung 4.4) wird zusätzlich zum Histogramm der Leistungen ein sogenannter Boxplot erstellt. In Abbildung 4.5 ist nochmal der interessante Ausschnitt aus dem Boxplot dargestellt.

„Der Boxplot (auch Box-Whisker-Plot oder deutsch Kastengrafik) ist ein Diagramm, das zur grafischen Darstellung der Verteilung kardinalskaliertter Daten verwendet wird. Es fasst dabei verschiedene robuste Streuungs- und Lagemaße in einer Darstellung zusammen. Ein Boxplot soll schnell einen Eindruck darüber vermitteln, in welchem Bereich die Daten liegen und wie sie sich über diesen Bereich verteilen. Deshalb werden alle Werte der sogenannten Fünf-Punkte-Zusammenfassung, also der Median, die zwei Quartile und die beiden Extremwerte, dargestellt.“ (Quelle: [18])

In der hier gewählten Darstellung ist die rote Linie in der Box der Medianwert, die beiden Quartile sind die äußeren Grenzen der Box, die beiden Extremwerte werden durch die so genannten Antennen angegeben. Dabei dürfen diese Extremwerte maximal das 1,5-Fache des Interquartilsabstands entfernt sein. Anderenfalls werden sie als Ausreißer bezeichnet und in der Darstellung mit einem roten + markiert.

## 4. Auswertungen

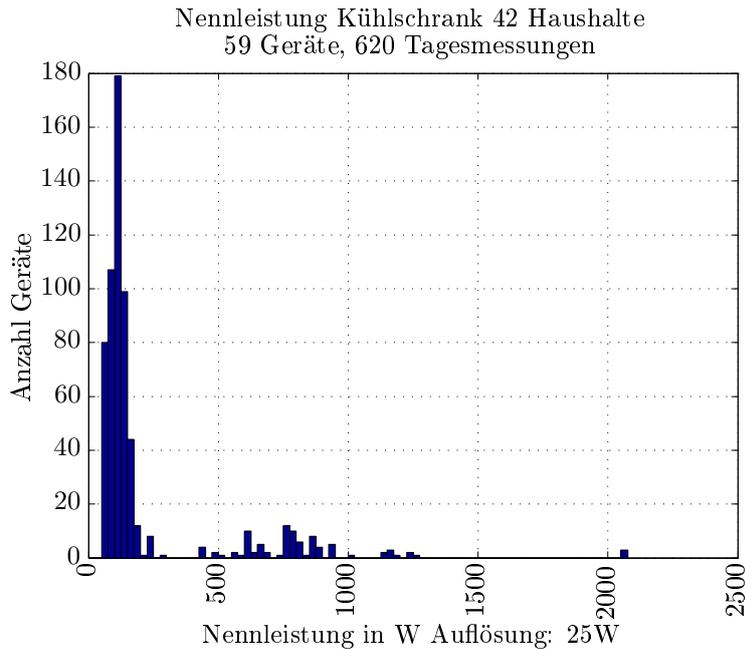


Abbildung 4.3.: Nennleistung für Kühlschränke

Hierbei wird zwischen der Maximalleistung und der Durchschnittsleistung während des Zyklus unterschieden.

In der Abbildung 4.6 ist der Energieverbrauch aller hier gemessenen Kühlschränke dargestellt. Weiters sind in dieser Abbildung die Energieeffizienzklassen in etwa wiedergegeben. Der genaue Leistungswert der jeweiligen Klasse hängt von der Größe des Kühlschranks, der Ausführungsform (ob mit oder ohne integriertem Gefrierfach oder als Kühl- und Gefrierkombinationsgerät). Für weitere Informationen zu diesem Thema auch bei anderen Geräten siehe [13]. Die Werte für die verwendeten Klassen in der Abbildung stammen aus [12] und entsprechen jenen eines Kühlschranks ohne Gefrierfach. Da weitere Informationen über die tatsächlich in Betrieb befindlichen Geräte fehlen, ist eine Einteilung in eine bestimmte Klasse mit äußerster Vorsicht zu genießen, da die Größe des Kühlschranks einen großen Einfluss auf die Leistungswerte der jeweiligen Klasse besitzt.

Weiters ist es interessant, wie oft am Tag sich ein Kühlschrank im Schnitt einschaltet und die Verteilung dieser Einschalthäufigkeit pro Tag ist in Abbildung 4.7 ersichtlich. Wie auf den ersten Blick erkennbar, ist eine Zyklenzahl von 500 oder 600 nicht wahrscheinlich und deutet auch hier wieder auf einen Fehler in den Messdaten hin. Als Maximalwert an Zyklen pro Tag ergibt sich hier 613 Zyklen, was in etwa einem Einschalten alle 2 Sekunden entspricht. Der Mittelwert wäre 48 Zyklen. Um eine genauere Schlussfolgerung zu ermöglichen, wurde der Bereich bis 100 Zyklen vergrößert dargestellt und alle Werte über 100 aus der Auswertung ausgeklammert. Das Ergebnis ist in

## 4. Auswertungen

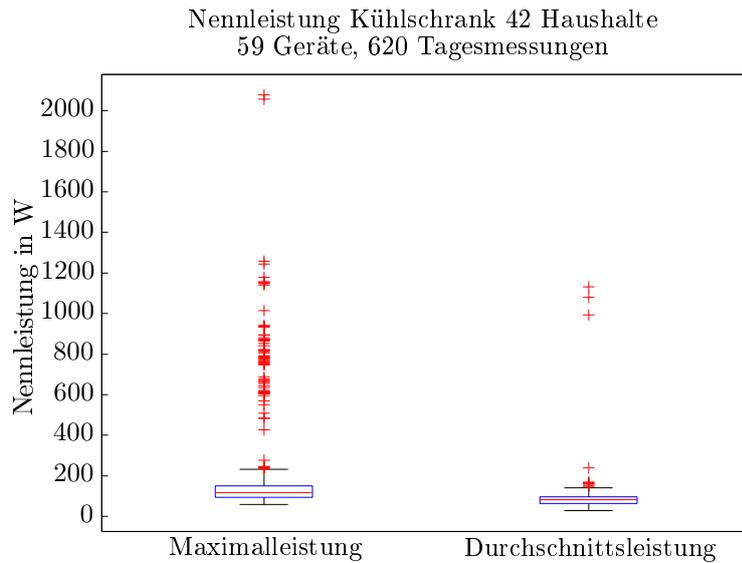


Abbildung 4.4.: Nennleistung für Kühlschränke - Boxplot

Abbildung 4.8 zu sehen. Klar erkenntlich ist ein Mittelwert bei 40 Zyklen und rechnerisch ergibt sich 32.8 Zyklen pro Tag. Dies erscheint schon glaubwürdiger, da es einem Einschalten etwa alle 3/4 Stunden entspricht. Der Maximalwert hier ist noch immer 3 Mal so hoch wie der Mittelwert und erscheint auch noch etwas hoch, da sich hier der Kühlschrank alle Viertelstunden einschalten würde.

Weiters findet man in der Literatur des Öfteren eine Darstellung mit dem Leistungsverbrauch pro Person diverser Geräte. Hier erwartet man sich auf Grund der annähernden Gleichverteilung der Einschaltzeitpunkte und der geringen Streuung der Laufzeit einen konstanten Verlauf des mittleren Leistungsverbrauches aller Haushalte über einen Tag verteilt. In Abbildung 4.9 sieht man, dass die Annahme eines gleichmäßigen mittleren Verbrauches durchaus gerechtfertigt ist und man nur einen kleinen Fehler macht, wenn man den Verlauf als tatsächlich gleichmäßig annimmt.

Als nächstes wurde untersucht, ob sich ein Unterschied in der Einschalthäufigkeit zwischen Sommer-, und Wintertag erkennen lässt. Vermuten würde man, dass sich ein Kühlschrank im Winter weniger oft einschalten muss, da die Raumtemperatur tendenziell geringer ist als im Sommer. Wider Erwarten sieht man in Abbildung 4.10 genau das gegenteilige Verhalten. Im Sommer sind Kühlschränke weniger oft in Betrieb als im Winter. Dieses Ergebnis ist sehr überraschend und der genaue Grund für dieses Verhalten wäre sehr interessant und eine erste Vermutung wäre, dass im Winter auf Grund des vermehrten Einsatzes von Heizungen die Raumtemperatur im Schnitt höher ist als

## 4. Auswertungen

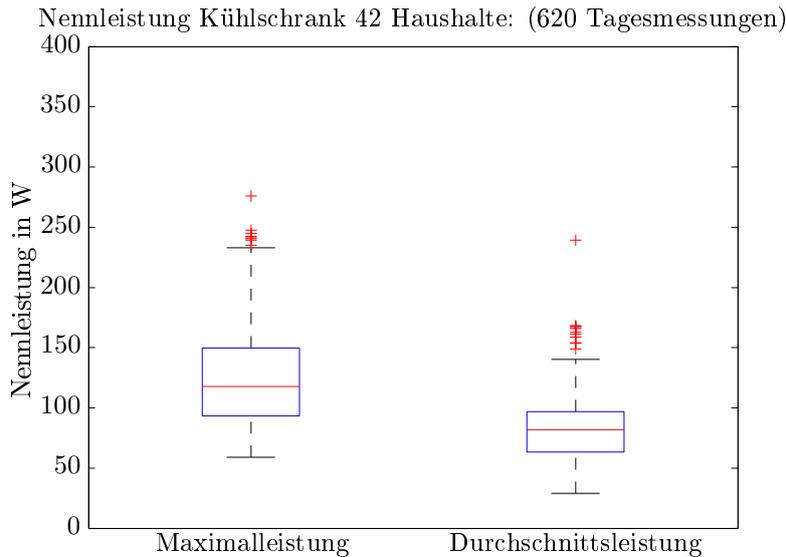


Abbildung 4.5.: Nennleistung für Kühlschränke - Ausschnitt aus dem Boxplot

im Sommer, da viele Haushalte im Sommer ihre Heizungen außer Betrieb nehmen und es daher kühler im Raum ist. Zu Beachten ist auch, dass bei dieser Messreihe Tage im Frühjahr zum Sommer und Tage im Herbst zum Winter gezählt wurden und daher das Ergebnis auch verfälscht ist.

Der Wirkleistungsverlauf wird hier nur am Rande betrachtet, da man während des Betriebes eine konstante Leistung vermuten kann und nur am Anfang des Zyklus mit einer Einschaltspitze zu rechnen ist. Der typische Verlauf ist in Abbildung 4.11 gemeinsam mit der Blindleistung dargestellt.

Da in dieser Messreihe auch der Blindleistungsverbrauch des gesamten Haushaltes vermessen wurde, wird nun noch ein Versuch unternommen, den Blindleistungsverlauf eines Kühlschranks zu bestimmen. Das größte Problem dabei ist, dass mehr als ein Gerät in der Summenmessung auftauchen kann und daher eine eindeutige Zuordnung zum betrachteten Gerät schwer möglich ist. Ein Beispiel für einen Blindleistungsverlauf sieht man in Abbildung 4.11. Man erkennt, dass sich während des betrachteten Zyklus ein weiteres Gerät (in diesem Fall noch ein Kühl-, oder Gefrierschrank) einschaltet und somit zu einer Veränderung der Blindleistungsbilanz führt. Betrachtet man aber einen Punkt, in dem der Verlauf der Detailmessung und der Summenmessung konstant bleiben, so kann man in diesem Punkt die Werte für Blind-, und Wirkleistung ablesen und daraus den Leistungsfaktor bestimmen. Es ergeben sich zum Beispiel im Punkt mit dem Zeitindex 2005 folgende Werte und das in Abbildung 4.12 dargestellte Zeigerdiagramm. Man erkennt, dass in diesem Betriebspunkt der Kühlschrank überkompensiert und somit kapazitiv ist.

#### 4. Auswertungen

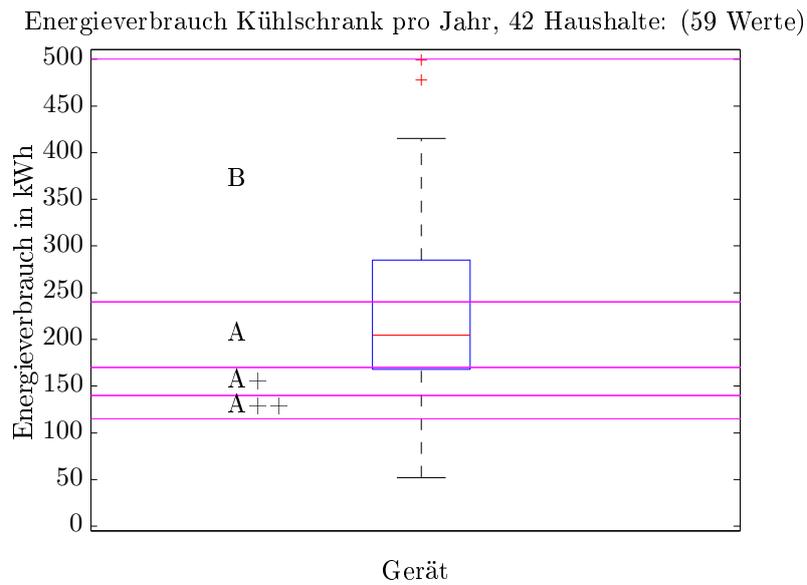


Abbildung 4.6.: Energieverbrauch der untersuchten Kühlschränke

$$P = 84.5W \quad Q = -37.14VA \Rightarrow S = \sqrt{P^2 + Q^2} = 92.3VA$$
$$\cos(\varphi) = \frac{P}{S} = 0.92\text{kap.} \Rightarrow \varphi = -22.9^\circ$$

#### 4. Auswertungen

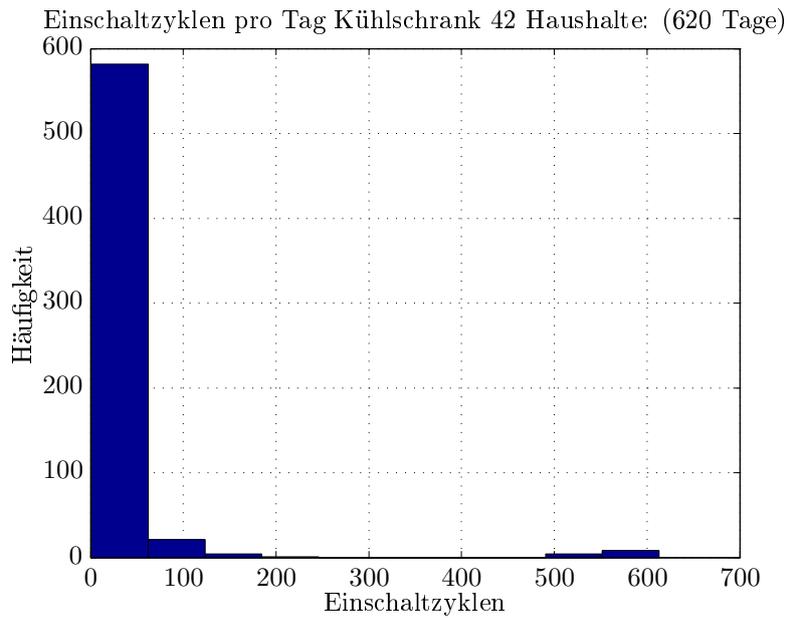


Abbildung 4.7.: Zyklen pro Tag für Kühlschränke

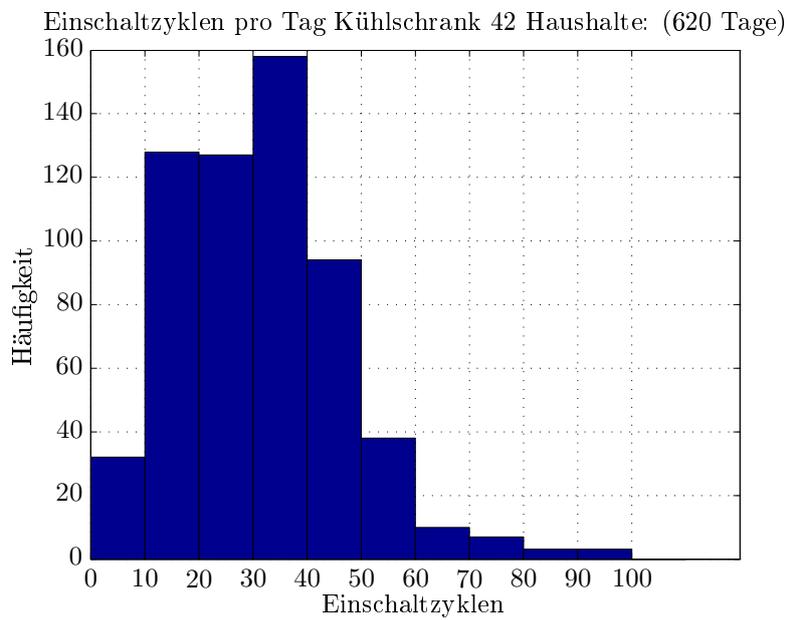


Abbildung 4.8.: Zyklen pro Tag für Kühlschränke (korrigiert)

#### 4. Auswertungen

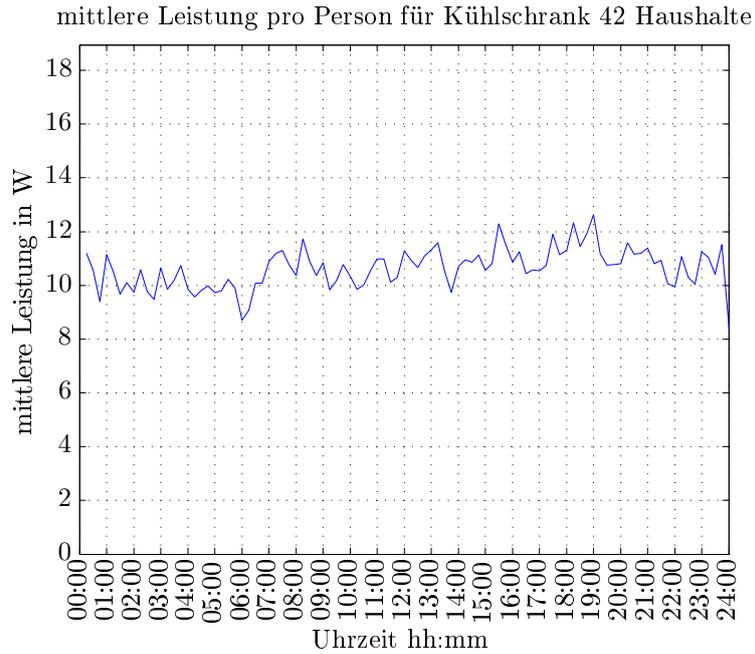


Abbildung 4.9.: mittlerer Leistungsverlauf eines Kühlschranks über den Tag verteilt

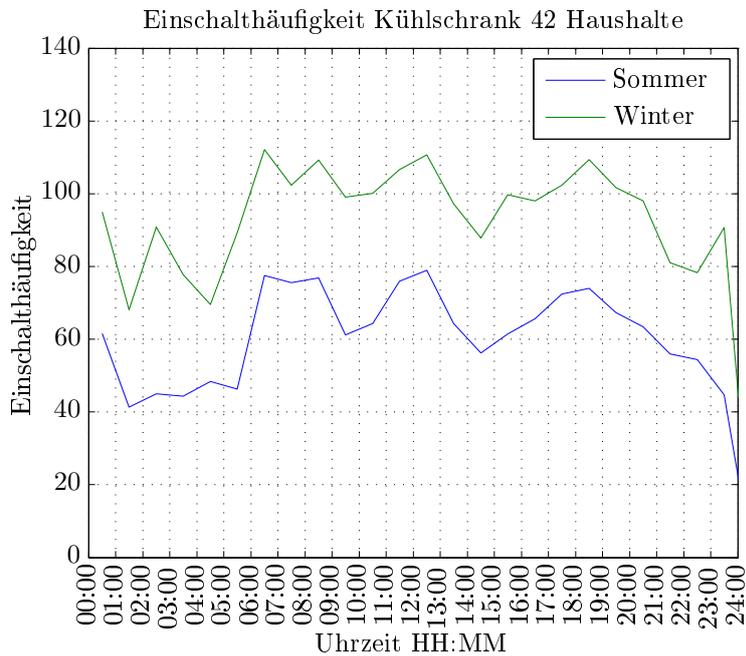


Abbildung 4.10.: Vergleich der Einschalthäufigkeit eines Kühlschranks für verschiedene Tagestypen

#### 4. Auswertungen

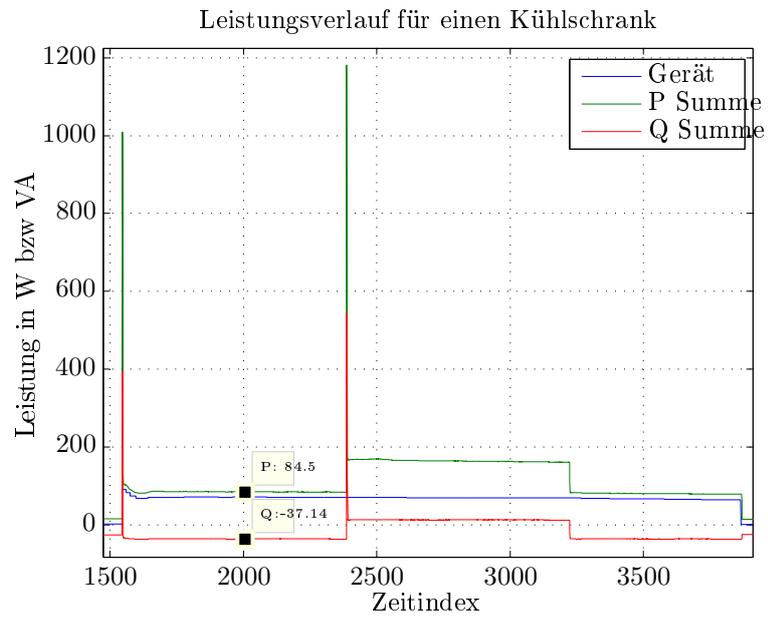


Abbildung 4.11.: Blindleistungsverlauf eines Kühlschranks

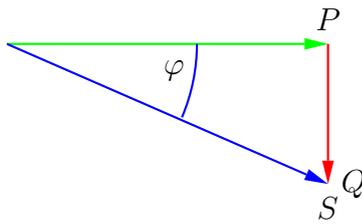


Abbildung 4.12.: Zeigerdiagramm für den betrachteten Zeitpunkt eines Kühlschranks

#### 4. Auswertungen

Ähnliche Ergebnisse, wie oben für Kühlschränke dargestellt, erwartet man auch bei der Auswertung von Gefriergeräten, wobei hier die Gleichverteilung des Einschaltzeitpunktes auf Grund des deutlich geringeren Öffnens der Geräte noch besser zu sehen ist (siehe Abbildung 4.13). Das Maximum um Mitternacht ist auf fehlerhaft erkannte Einschaltzeiten am Beginn der jeweiligen Tagesmessreihe zurückzuführen.

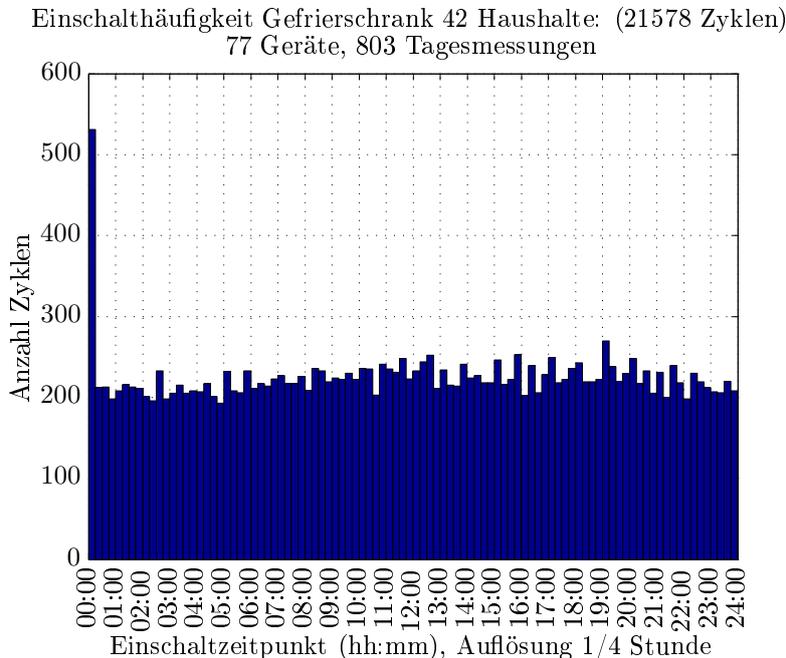


Abbildung 4.13.: Einschalthäufigkeit für Gefrierschränke

Auch die Einschaltdauer in Abbildung 4.14 zeigt ein deutliches Maximum im Bereich von 10-20 Minuten mit einem Mittelwert von 16 Minuten und 17 Sekunden.

Die Leistungsaufnahme, dargestellt in Abbildung 4.15 zeigt eine deutliche Streuung je nach Gerätealter und hat einen Mittelwert von 193W, einen Maximalwert von 1154W, der wieder zu hoch erscheint, und einen Minimalwert von 43.3W, der selbst für ein modernes Gerät der Energieeffizienzklasse A++ sehr niedrig erscheint. Die meisten dieser Geräte haben eine Anschlussleistung von rund 90 W. In Abbildung 4.16 sieht man den deutlichen Unterschied zwischen der Betrachtung der mittleren bzw. maximalen Nennleistung. Dieser deutliche Unterschied ist auf teilweise gemessene Einschaltspitzen am Beginn des Zyklus zurückzuführen, die durch die Berechnung des Durchschnittswertes nicht mehr so stark ins Gewicht fallen. Aus diesem Grund ist hier die Betrachtung des Durchschnittswertes besser geeignet. Manche Einschaltspitzen werden vom recht langsamen Messsystem erfasst, andere wiederum nicht, weshalb die Maximalleistung nicht geeignet ist, um Aussagen über die Verteilung zu treffen.

## 4. Auswertungen

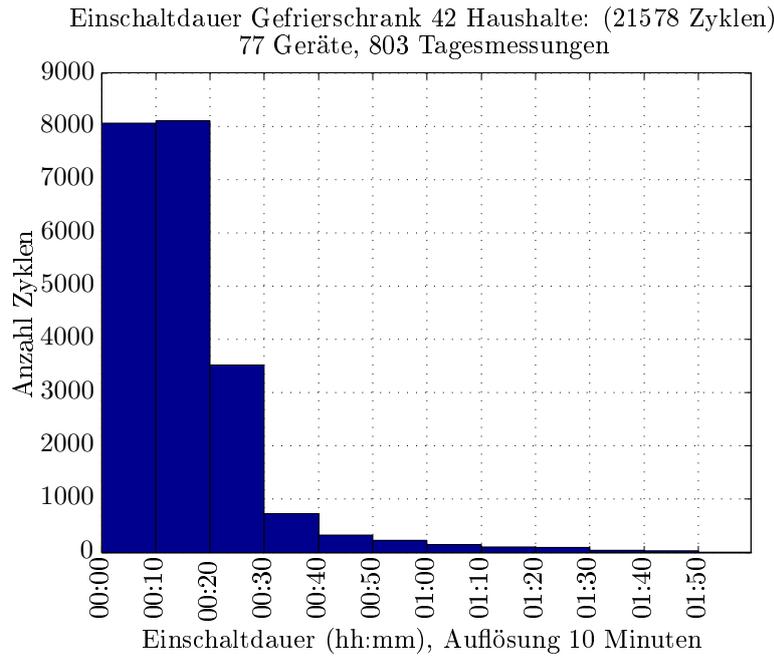


Abbildung 4.14.: Einschaltdauer für Gefrierschränke

Bei Gefriergeräten ist wie bei Kühlschränken auf den Energielabels der Jahresenergieverbrauch angegeben. Daher wird auch hier der Energieverbrauch in einem Jahr dargestellt. Weiters wurden wieder die Energieeffizienzklassen wie in [12] beschrieben, dargestellt. Hier ergibt sich die Darstellung nach Abbildung 4.17, in der man erkennt, dass ein Großteil der Geräte in die Klassen A und A+ fallen. Einige wenige Geräte liegen in den Klassen B und A++. Diese Streuung entspricht recht gut dem, was man sich erwartet, wenn man in Betracht zieht, dass Gefriergeräte eine durchaus lange Lebensdauer haben und die neueren und somit auch energieeffizienteren Geräte erst in den letzten Jahren auf dem Markt gekommen sind. Für eine Verteilung der beiden Geräte Kühlschrank und Gefrierschrank nach Energieklasse aufgeschlüsselt siehe beispielsweise für Österreich [7]. Dort erkennt man, dass ein Großteil der Geräte laut dieser Arbeit in noch höhere Klassen fallen, als hier in der Abbildung erkennbar. Ein möglicher Grund hierfür wäre, dass die Gerätegröße nicht jener entspricht, die bei der Berechnung der Energieklassen in der zitierten Quelle entsprechen.

Für Gefrierschränke wurde wieder die Einschalthäufigkeit pro Tag ausgewertet und ist in Abbildung 4.18 dargestellt. Erkennbar ist ein Mittelwert im Bereich von 30 Zyklen, rechnerisch ergeben sich 26.7 Zyklen pro Tag. Ein Maximalwert von 98 Zyklen / Tag erscheint wieder recht hoch, ist aber für ältere Geräte durchaus möglich.

Die mittlere Leistung über einen Tag ausgewertet, ergibt wieder erwartungsgemäß eine sehr geringe Streuung und einen deutlichen Mittelwert von rund 22W, zu erkennen in Abbildung 4.19.

## 4. Auswertungen

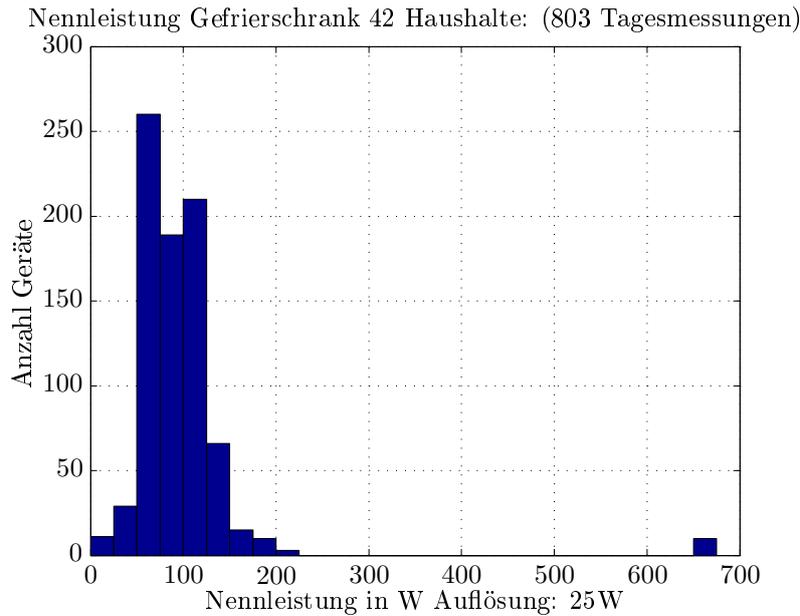


Abbildung 4.15.: Nennleistung für Gefrierschränke

Vergleicht man wieder die Einschalthäufigkeit im Winter und Sommer miteinander, so ergibt sich auf Grund des häufig im Keller aufgestellten Gefrierschranks und der damit verbundenen relativ konstanten Umgebungstemperatur kaum ein Unterschied in dieser Verteilung, wie in Abbildung 4.20 erkennbar.

Wie bei einem Kühlschrank ist der typische Leistungsverlauf während des Betriebes annähernd konstant. Auch die Blindleistungsbilanz ist mit jener eines Kühlschranks vergleichbar. In Abbildung 4.22 ist so ein typischer Verlauf dargestellt. In blau wird wieder die Wirk-, und in grün die Blindleistung dargestellt. Es wird für einen Betriebspunkt der Leistungsfaktor berechnet und das Zeigerdiagramm in Abbildung 4.21 dargestellt. Hier ergibt sich ein leicht induktives Verhalten, was auf eine Unterkompensierung im Nennpunkt schließen lässt.

$$P = 81W \quad Q = 24VA \Rightarrow S = \sqrt{P^2 + Q^2} = 84.5VA$$

$$\cos(\varphi) = \frac{P}{S} = 0.959 \text{ind.} \Rightarrow \varphi = 16.5^\circ$$

In Tabelle 4.1 werden die wichtigsten ermittelten Parameter nochmals für die beiden hier beschriebenen Geräte zusammengefasst.

#### 4. Auswertungen

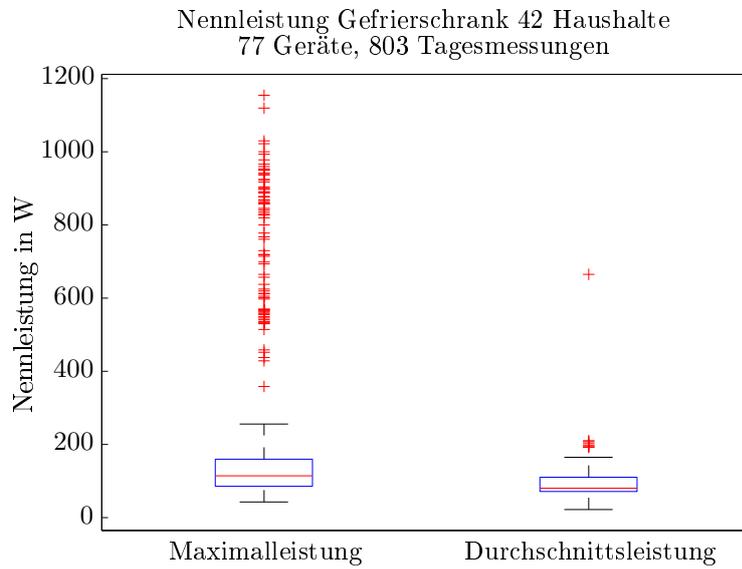


Abbildung 4.16.: Nennleistung für Gefrierschränke - Boxplot

Gerät	Leistung	Zyklen-Zeit	Zyklen/Tag
Kühlschrank	232.5W	10:15 Min.	32.8 $\approx$ 33
Gefrierschrank	193W	17:38 Min.	26,7 $\approx$ 27

Tabelle 4.1.: Zusammenstellung der wichtigsten Daten für zyklische Geräte

## 4. Auswertungen

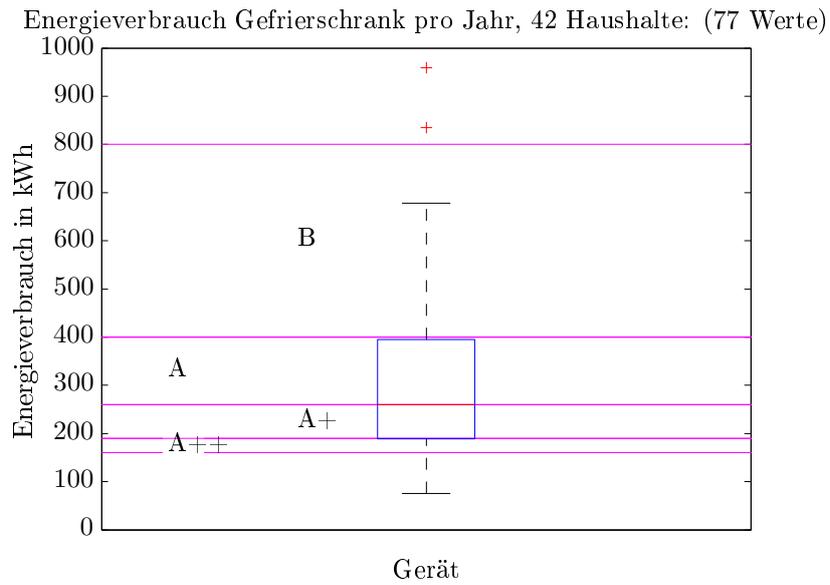


Abbildung 4.17.: Energieverbrauch pro Jahr für Gefrierschränke mit Energieeffizienzklassen nach [12]

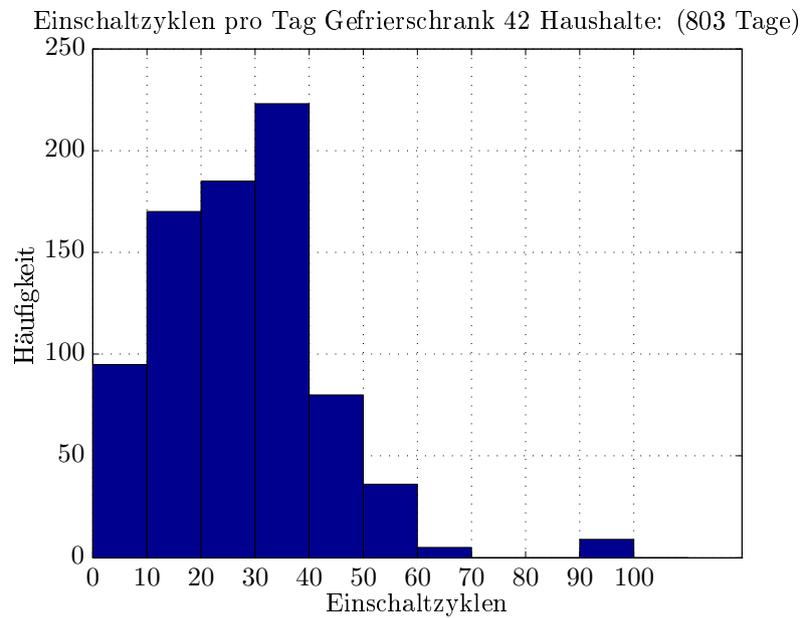


Abbildung 4.18.: Zyklen pro Tag für Gefrierschränke

## 4. Auswertungen

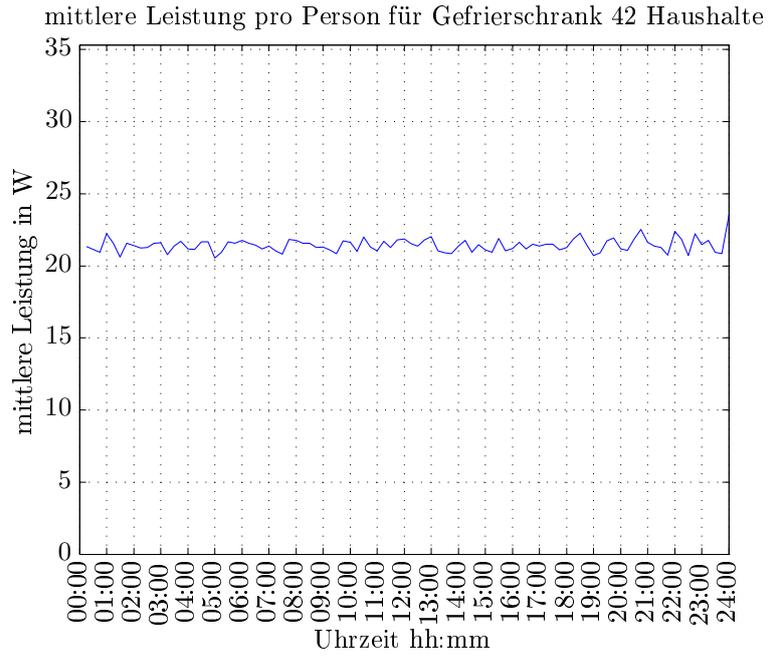


Abbildung 4.19.: mittlere Tagesleistung für Gefrierschränke

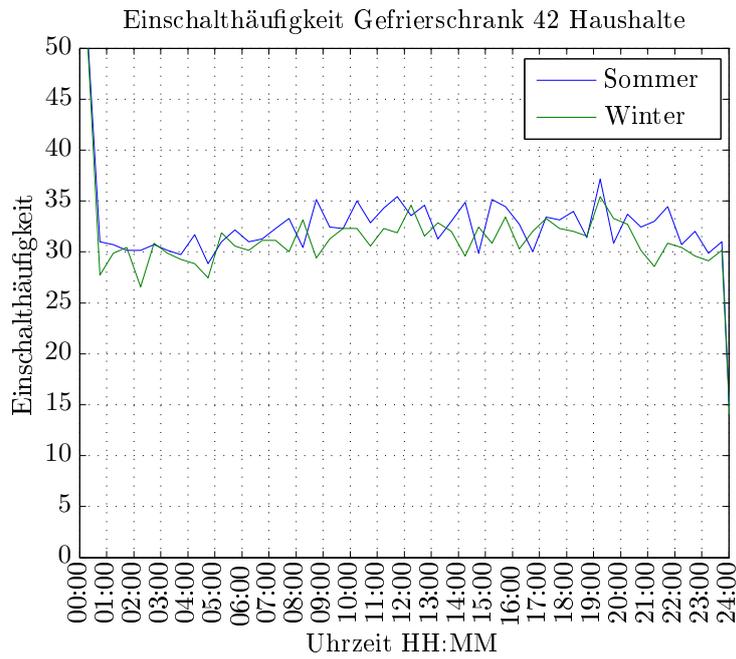


Abbildung 4.20.: Vergleich der Einschalthäufigkeit von Sommer und Winter für Gefrierschränke

#### 4. Auswertungen

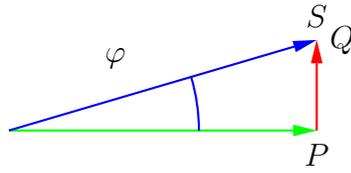


Abbildung 4.21.: Zeigerdiagramm für den betrachteten Zeitpunkt eines Gefrierschranks

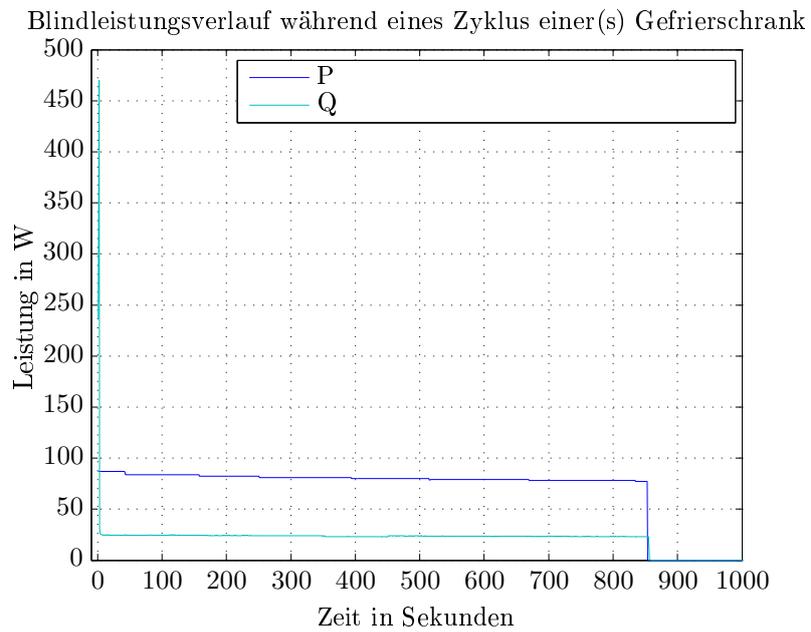


Abbildung 4.22.: Leistungsverlauf für einen typischen Zyklus für Gefrierschränke

## 4.1.2. Waschmaschinen

Bei der Untersuchung von Waschmaschinen wurden wieder die im vorigen Abschnitt beschriebenen Auswertungen durchgeführt. Zusätzlich sind die Leistungsverläufe während eines Zyklus von Interesse, da über diesen Verlauf wenig bis gar keine Informationen verfügbar sind. In Abbildung 4.23 ist die Einsatzhäufigkeit von Waschmaschinen über den Tag verteilt dargestellt. Man erkennt hier schon deutlich, dass Waschmaschinen im Vergleich zu Kühlschränken natürlich weniger oft in Verwendung sind und daher viel weniger Informationen über die Verteilung der Einsatzzeiten vorhanden sind. Die relativ hohe Zahl an Zyklen um Mitternacht ist wiederum auf fehlerhaft erkannte Zyklen zurückzuführen, die durch automatische Algorithmen noch nicht aussortiert worden sind.

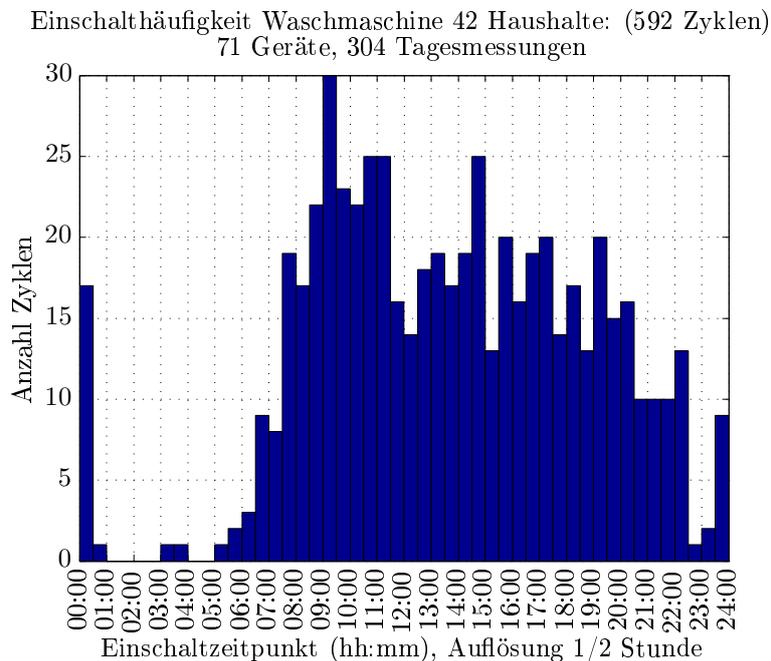


Abbildung 4.23.: Einschaltzeitpunkte von Waschmaschinen

Interessant wäre es auch, den Unterschied zwischen Verwendung von Waschmaschinen an Werktagen und am Wochenende darzustellen. Wie man aus den beiden Abbildungen (4.24 für die Sommer-Messreihe und 4.25 für die Winter-Messreihe) hierzu entnehmen kann, ist hier ein kleiner Unterschied feststellbar. Man erkennt, dass sowohl im Winter als auch im Sommer vermehrt am Wochenende gewaschen wird. Waschmaschinen werden laut dieser Messreihe mit einer etwaigen Gleichverteilung zwischen 07:00 und 22:00 in Betrieb genommen. Auf Grund des geringen Umfangs an Messreihen zu diesem Gerätetyp ist eine genauere Aussage nur mit allzu großer Vorsicht möglich.

Die Einschaltdauer von Waschmaschinen wird in Abbildung 4.26 dargestellt. Man er-

#### 4. Auswertungen

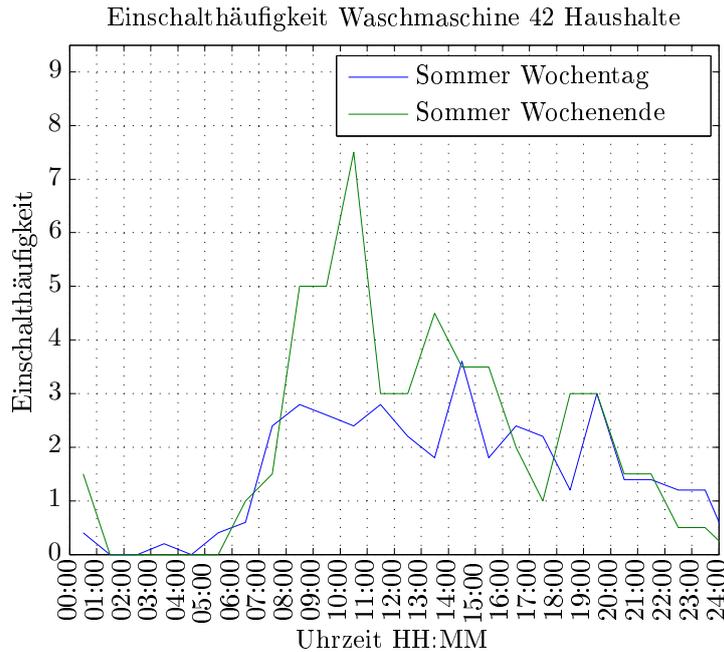


Abbildung 4.24.: Einschaltzeitpunkte von Waschmaschinen im Sommer

kennt eindeutig ein Maximum im Bereich von 1 - 2 Stunden Betriebsdauer, das durchaus realistisch erscheint. Es ergibt sich eine maximale Laufzeit von 24 Stunden, die auf einen Fehler in der Erkennung des Zyklus zurückzuführen ist. Der Mittelwert der Zyklenlänge wäre hier 2:28 Stunden, was auch noch hoch erscheint. Weiters sieht man in der Abbildung vereinzelte Zyklen mit einer Laufzeit von mehr als 6 Stunden. Diese Zyklen werden in einem nächsten Schritt ausgeklammert und danach werden die verbleibenden Werte erneut dargestellt (Abbildung 4.27). Nun ergibt sich eine maximale Laufzeit von 5:19 Stunden, die noch immer sehr hoch erscheint und vielleicht sollte die Begrenzung auf 3 Stunden gesetzt werden. Der Mittelwert der Einschaltdauer liegt mit 1:47 Stunden schon recht gut in den aus der Erfahrung bekannten Werten (siehe hierzu z.B. auch [10] - Seite 11) für einen Waschgang von 1 - 2,5 Stunden.

Die Untersuchung des Leistungsverlaufes einzelner Zyklen im Vergleich wird im folgenden behandelt. Aus der maximalen theoretischen Zyklendauer von 3 Stunden ergibt sich leider auch, dass manchmal, vor allem bei Kurzprogrammen, zwei Zyklen fälschlicherweise zusammengefasst werden und dann die Anfangszeitpunkte der Zyklen nicht mehr stimmen, da dann Leistungssprünge im vorher abgeschnittenen Zyklus als neuer Zyklus erkannt werden und damit auch nachfolgende Zyklen nur mehr eingeschränkt verwertbare Ergebnisse liefern. Ein Beispiel für eine fehlerhafte Erkennung von zwei Zyklen als einen einzigen Zyklus ist in Abbildung 4.28 sehr gut zu sehen. Daraus ist zu schließen, dass der verwendete Suchalgorithmus nicht wirklich geeignet scheint. Die ursprüngliche Annahme bei der Erstellung des relativ einfachen Algorithmus war, dass Waschmaschinen nicht mehrere Male mit geringem Zeitabstand hintereinander eingeschaltet werden,

## 4. Auswertungen

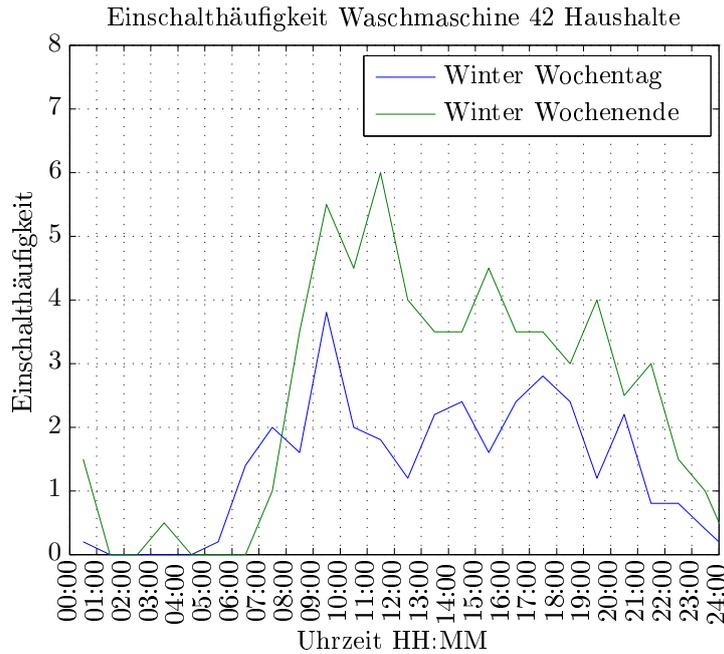


Abbildung 4.25.: Einschaltzeitpunkte von Waschmaschinen im Winter

was sich leider als nicht richtig herausstellte. Vielmehr erkennt man, dass es anscheinend in vielen Haushalten einen „Washtag“ gibt, an dem mehrere Wäscheladungen hintereinander gewaschen werden. Daher wurde ein neuer Suchalgorithmus entwickelt (siehe 2.2), der ausschließlich für Waschmaschinen Verwendung findet und speziell auf die vorher beschriebenen Probleme ausgelegt ist. Mit diesem Algorithmus wurden kaum fehlerhafte Zyklen erkannt. Vereinzelt Zyklen mit konstanter Leistungsaufnahme mit einer Dauer von über 3 Stunden wurden händisch ausgeschlossen.

Bei Waschmaschinen ist die mittlere Leistung je Zyklus in einem relativ kleinen Bereich verteilt, wohingegen die maximale Leistung durch unterschiedliche Heizstäbe eine deutliche Häufung bei rund 2kW, 2.75kW sowie 3.5kW aufweist. Diese Häufung ist besonders gut in Abbildung 4.29 erkennbar, die Verteilung von Durchschnitts-, und Maximalleistung in Abbildung 4.30.

Nun wurde der Energieverbrauch im Vergleich mit den in [12] beschriebenen Werten verglichen. Dabei wurde von einer Waschmenge von 6 kg ausgegangen, da die Referenzzyklen pro kg gewaschener Wäsche ausgehen. Ebenso einen Einfluss auf den Energieverbrauch hat logischerweise die Waschtemperatur sowie das gewählte Programm. Als Referenzprogramm ist ein Norm-Programm „60°C Baumwolle“ angegeben. Daher weisen die Energiewerte durchaus eine hohe Streuung auf und der Vergleich mit den Energieeffizienzklassen ist nur ein einfacher Versuch, einen solchen Vergleich anzustellen und gibt wie schon beim Kühlschrank oder Gefrierschrank auf Grund der zahlreichen nicht näher bekannten Einflussfaktoren auf die Klasseneinteilung nur eine grobe Nä-

## 4. Auswertungen

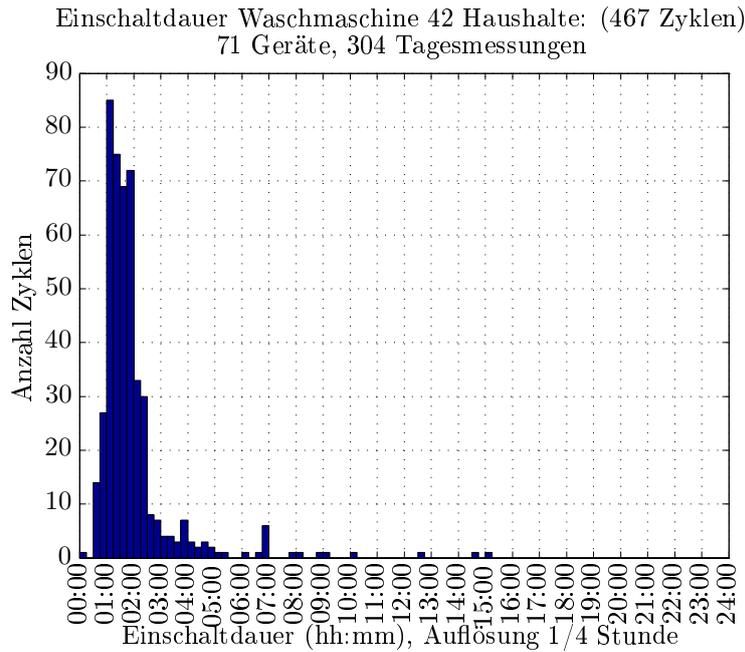


Abbildung 4.26.: Programmlaufzeit von Waschmaschinen

herung wieder, die nicht unbedingt direkt mit der tatsächlichen Energieeffizienzklasse der Geräte zu tun haben muss. In Abbildung 4.31 sieht man nun diesen Vergleich. Man erkennt eine deutliche Streuung des Energieverbrauchs, aber ein durchschnittlicher Energieverbrauch von 0.84 kWh würde bei 6 kg Wäsche und dem Normprogramm auf Klasse A führen, was als Durchschnittswert für alle in Verwendung befindlichen Geräte durchaus realistisch ist.

Nun zu der Untersuchung eines einzelnen Zyklus einer Waschmaschine im Detail. Wie in den Abbildungen 4.32 und 4.33 ersichtlich, ändert sich die Leistungsaufnahme während des Betriebes mehrmals und lässt sich in einige charakteristische Bereiche einteilen.

Man sieht während des Waschganges immer wieder Phasen, wo eine Leistung von rund 2kW verbraucht wird. Diese Bereiche sind auf Heizphasen zurückzuführen. Die meisten gängigen Waschmaschinen haben ein Heizelement mit etwa 2 kW verbaut (siehe u.a. [11]). Dies dauert wie in den beiden Grafiken ersichtlich je nach Temperatur klarerweise unterschiedlich lang. Beim ersten Zyklus dauert die Aufheizphase beispielsweise rund 5 Minuten und daran schließt eine zweite Aufheizphase mit rund 1.5 Minuten an. Beim zweiten Zyklus dauert diese Phase deutlich länger, nämlich etwa eine halbe Stunde. Dies lässt auf eine höhere Wascht Temperatur beim zweiten Zyklus schließen. Außerdem ist der Leistungswert etwas höher, was auf den Betrieb eines zweiten Moduls während der gleichen Zeit zurückzuführen ist. Ein Heizstab würde unabhängig von seiner Einsatzdauer in etwa immer die gleiche Leistung benötigen.

## 4. Auswertungen

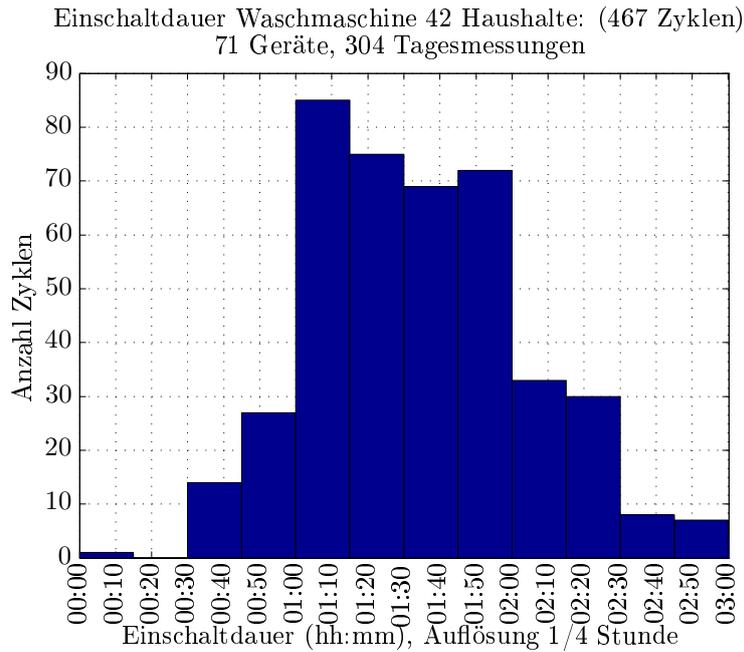


Abbildung 4.27.: Programmlaufzeit von Waschmaschinen (korrigiert auf maximal 6 Stunden)

Weitere markante Bereiche in den Abbildungen sind die Bereiche nahe des Endes des jeweiligen Waschganges. Dies sind die abschließenden Schleuderzyklen, die mit ihrem Verlauf sehr markant sind. Eine hohe Einschaltspitze und daran anschließend ein Abklingen der Leistung ist charakteristisch für Motoren und die Leistung von rund 300 W ist für einen Motor bei Waschmaschinen auch durchaus üblich ist. Meistens wird ein Einphasen-Reihenschlussmotor mit mehreren Statorwicklungen verwendet, damit man auf leichte Weise verschiedene Drehzahlen einstellen kann (siehe hierzu u.a. [19]). Daher kann die Leistungsaufnahme des Motors auch je nach Programm schwanken. Die Schleuderzeit beträgt in diesen beiden Beispielen rund 2 sowie rund 5.5 Minuten.

Die restlichen Module wie Laugenpumpe oder Ablaufpumpe haben typischerweise Leistungen um die 30 Watt und sind daher in der Darstellung auf Grund des Maßstabes nicht gut erkennbar.

Weiters wird in Abbildung 4.33 in grün auch der Blindleistungsverlauf während des gesamten Waschzyklus dargestellt. Man erkennt, dass der ohmsche Heizstab natürlich keine Blindleistung aufnimmt. Während der Schleuderphasen sieht man eine Zunahme der Blindleistungsaufnahme zu Beginn, also beim Einschalten des Motors. Danach nimmt die Blindleistung wieder ab, was darauf schließen lässt, dass der Motor für den Normalbetrieb kompensiert ist.

Für die Erstellung von synthetischen Lastprofilen ist hier wieder der mittlere Lei-

## 4. Auswertungen

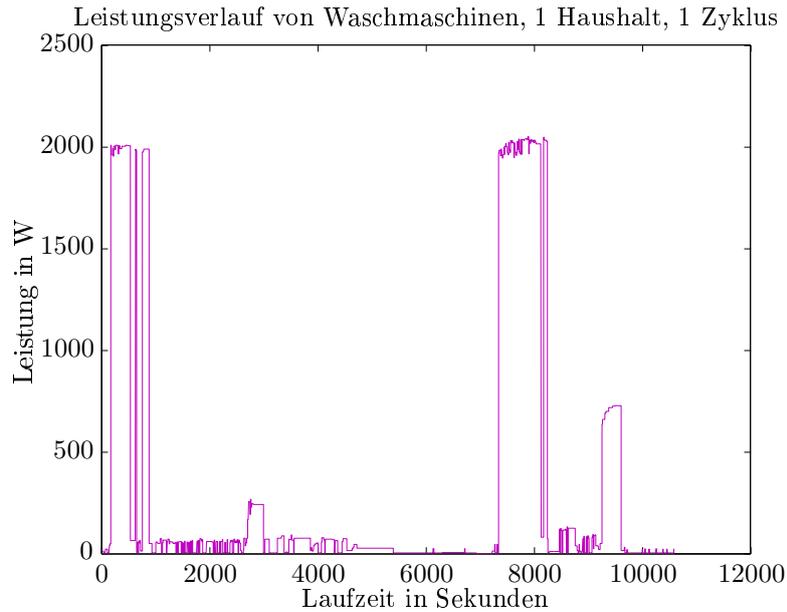


Abbildung 4.28.: Waschmaschine, zwei Zyklen als ein zusammenhängender erkannt

stungsverlauf für Waschmaschinen über einen Tag dargestellt von großem Interesse. Man erkennt eine gewisse Ähnlichkeit mit den in der Literatur verwendeten Verläufen (siehe z.B.: [9]), wie in Abbildung 4.34 dargestellt, Die charakteristische Spitze um 11:00 Uhr ist klar erkennbar, allerdings nicht so stark ausgeprägt wie im Vergleichsverlauf und somit decken sich die bisherigen Daten recht gut mit dieser Auswertung.

Betrachtet man die Einschalthäufigkeit pro Tag, so ergibt sich ein Mittelwert von 1.57 Zyklen / Tag, wie in Abbildung 4.35 klar erkennbar ist. Hierbei ist zu beachten, dass nur Tage gewertet werden, an denen mindestens ein Zyklus aufgezeichnet wurde. Daher ist dieses Ergebnis viel höher, als man tatsächlich erwarten würde. Es besagt nur, dass im Schnitt die Hälfte der Haushalte, in denen gewaschen wurde, einen zweiten Waschgang am gleichen Tag durchführen und bestätigt daher wieder die vorher gefundenen knapp aufeinander folgenden Zyklen, die eigens korrigiert werden mussten.

Betrachtet man jedoch auch Tage in dieser Auswertung, an denen kein Zyklus aufgezeichnet bzw. gefunden wurde, so ergibt sich die Darstellung in Abbildung 4.37. Man erkennt eine deutliche Häufung bei 0, was auch zu erwarten war. Man kann erwarten, dass in deutlich weniger als der Hälfte der Tage nicht gewaschen wird. Tatsächlich wird an 304 von 749 gemessenen Tagesreihen ein Waschgang erkannt. Es ergibt sich klarerweise ein deutlich niedrigerer Mittelwert von 0.64 Zyklen pro Tag, also noch immer, dass im Mittel an etwa 2 von 3 Tagen gewaschen wird.

Im Vergleich der Einschalthäufigkeit pro Tag erkennt man, dass generell öfter am Wochenende gewaschen wird als unter der Woche. Ein Unterschied in der Verwendung

#### 4. Auswertungen

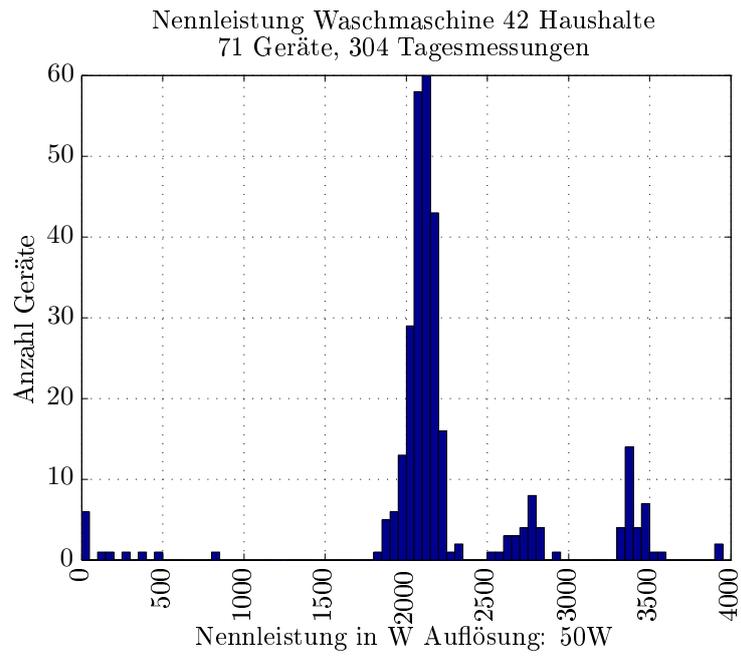


Abbildung 4.29.: Waschmaschine, Nennleistung Maximalwert

von Waschmaschinen zwischen Sommer und Winter wird nicht erwartet und ist auch kaum erkennbar

## 4. Auswertungen

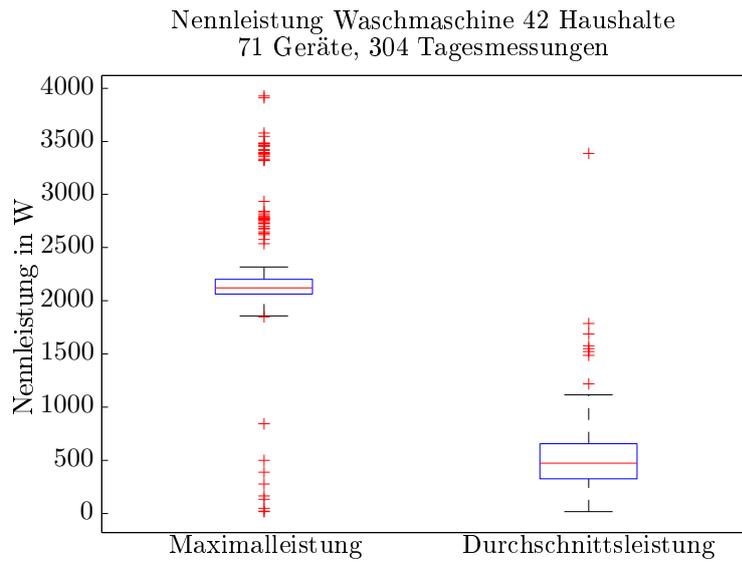


Abbildung 4.30.: Waschmaschine, Boxplot von maximaler sowie durchschnittlicher Nennleistung

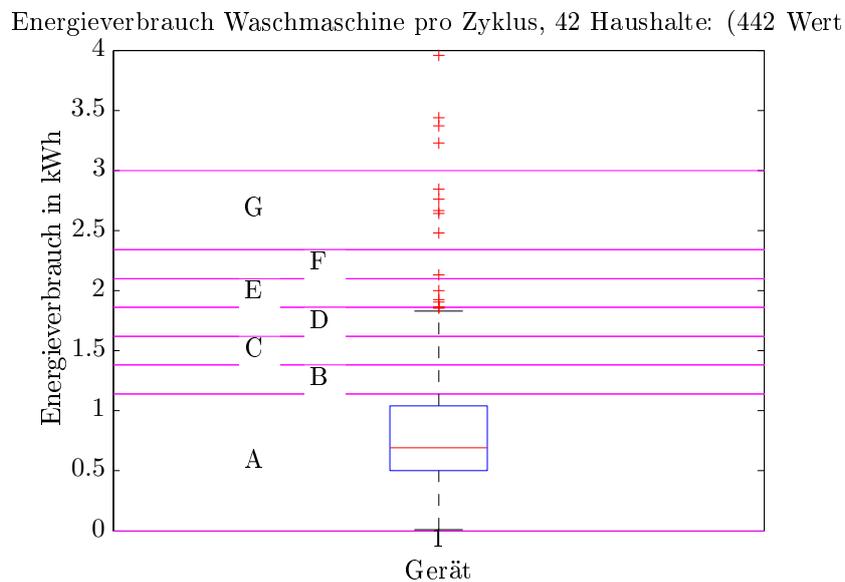


Abbildung 4.31.: Waschmaschine, Energieverbrauch je Zyklus und Energieeffizienzklassen laut [12]

#### 4. Auswertungen

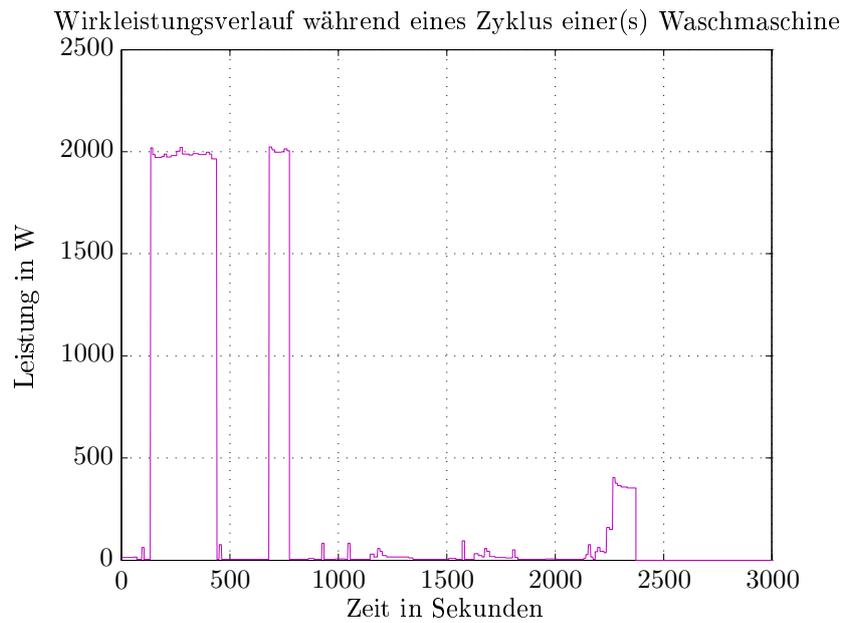


Abbildung 4.32.: Waschmaschine, Leistungsverlauf eines Zyklus

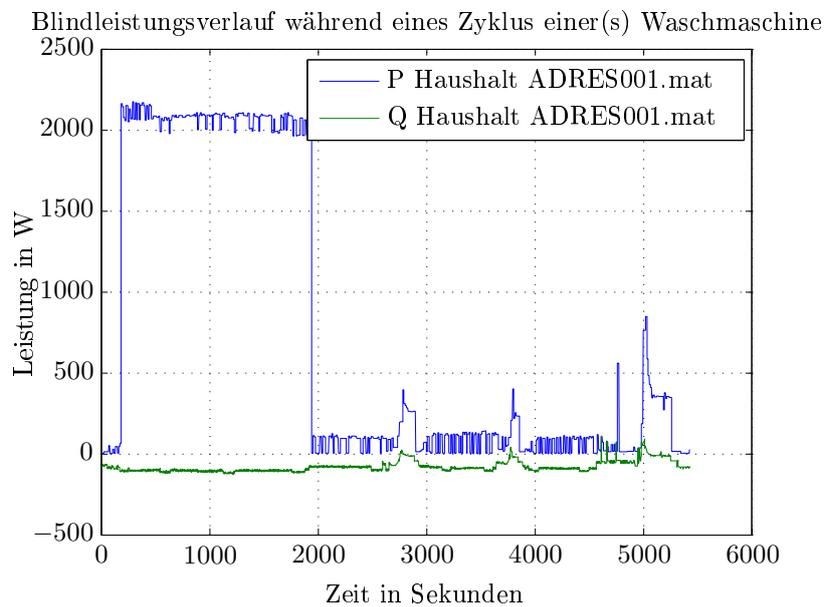


Abbildung 4.33.: Waschmaschine, Wirk-, und Blindleistungsverlauf eines Zyklus

#### 4. Auswertungen

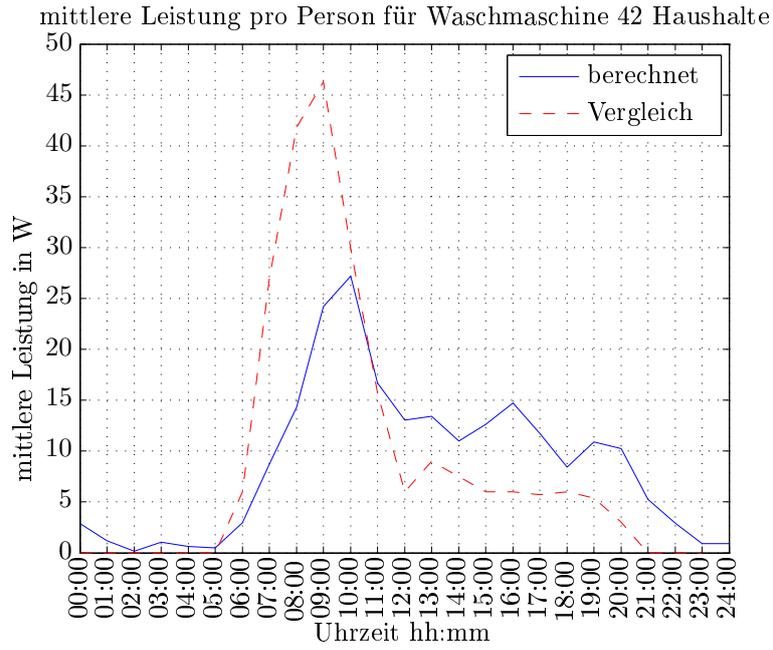


Abbildung 4.34.: Waschmaschine, Verlauf der mittleren Tagesleistung mit Vergleichsverlauf nach [9]

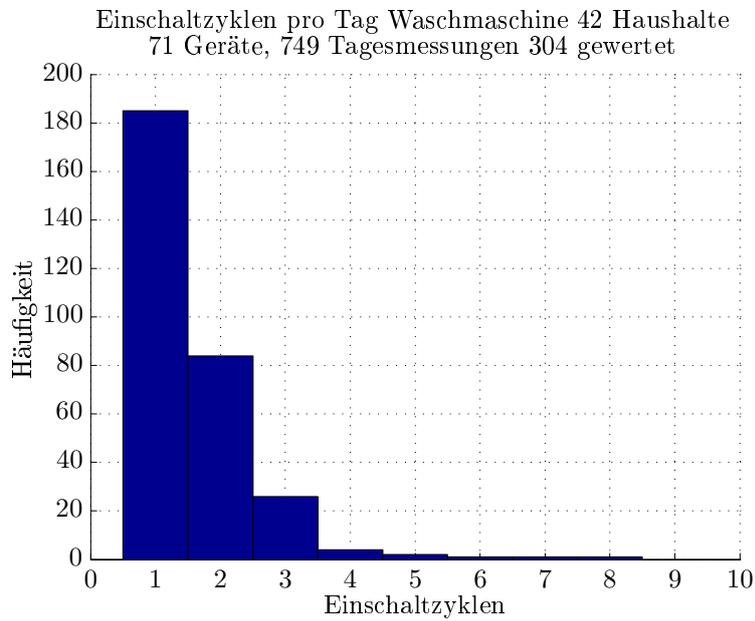


Abbildung 4.35.: Waschmaschine, Einschalthäufigkeit pro Tag

#### 4. Auswertungen

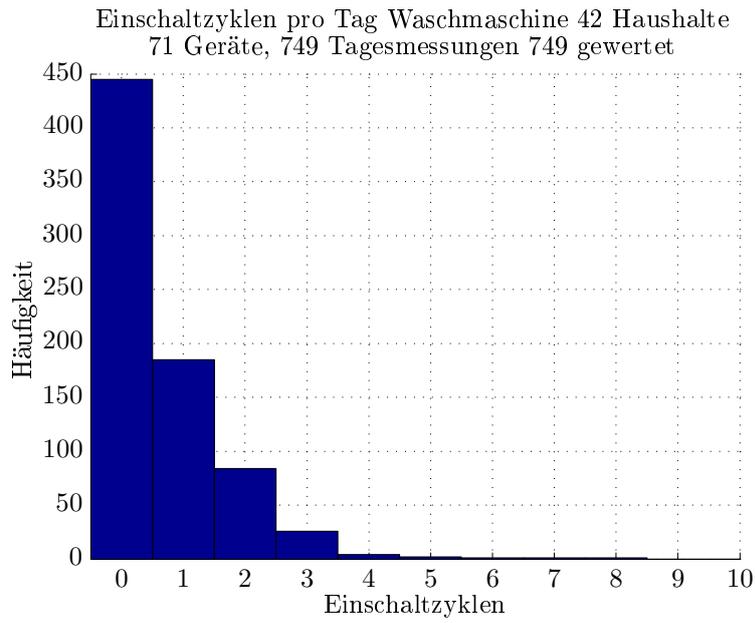


Abbildung 4.36.: Waschmaschine, Einschalthäufigkeit pro Tag mit Tagen ohne Zyklus

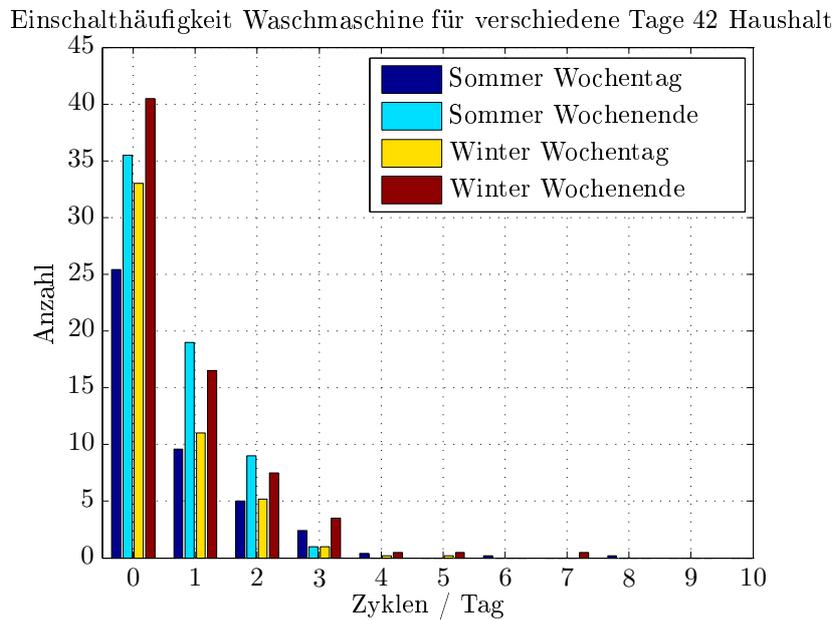


Abbildung 4.37.: Waschmaschine, Vergleich der Einschalthäufigkeit pro Tag für Sommer und Winter

## 4.1.3. Wäschetrockner

Ein weiteres Gerät mit einer alternierenden Leistungsaufnahme während des Betriebes ist ein Wäschetrockner. Die Einschalthäufigkeit über einen Tag verteilt ist in Abbildung 4.38 dargestellt. Wie man erkennt, wurden für Trockner sehr wenige Zyklen vermessen und daher ist eine feinere Unterteilung des Diagramms nicht weiter sinnvoll. Aber auch aus diesen wenigen Zyklen kann man einige interessante Aussagen treffen. Zum Beispiel, dass die Zyklen mit Beginnzeit um 0:00 falsch erkannte Zyklen sein müssten. Weiters sind Trockner vorwiegend in der Zeit von 07:00 - 22:00 mit einem kontinuierlichen Anstieg der Einsatzzeit mit fortschreitender Tageszeit und einer deutlichen Spitze um 11:00 sowie einer leichten Abendspitze um 19:00 in Betrieb genommen worden.

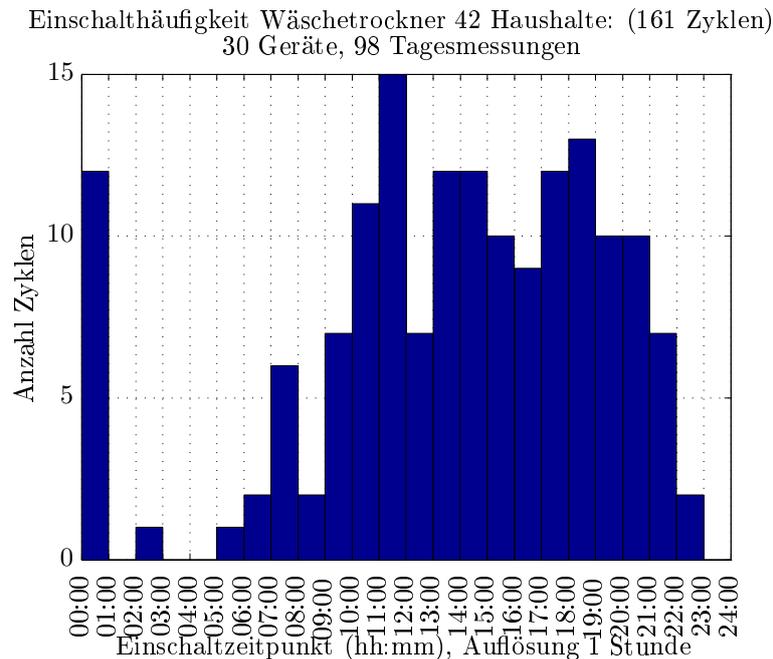


Abbildung 4.38.: Einschaltzeitpunkte von Wäschetrocknern

Bei Trocknern liegt die Einschaltzeit ähnlich wie bei Waschmaschinen typischerweise bei 1.5 - 2 Stunden, wie auch Abbildung 4.39 entnommen werden kann. In der Abbildung sind auch wieder einige fehlerhaft erkannte Zyklen zu erkennen, die eine Dauer von über 5 Stunden aufweisen und daher nicht realistisch sind.

Die typische Nennleistung von Wäschetrocknern liegt bei knapp 2.5 kW und schwankt nur geringfügig um diesen Wert wie aus Abbildung 4.40 hervorgeht. Man muss hier auch wieder zwischen den beiden Möglichkeiten zur Ermittlung der Nennleistung unterscheiden. Wählt man die maximale Leistung, die während eines Zyklus auftritt, so ist diese hier klarerweise deutlich höher als die durchschnittliche Leistung, erkennbar in Abbildung 4.41. Allerdings erkennt man, dass beide Werte eine etwa gleich große Streuung

## 4. Auswertungen

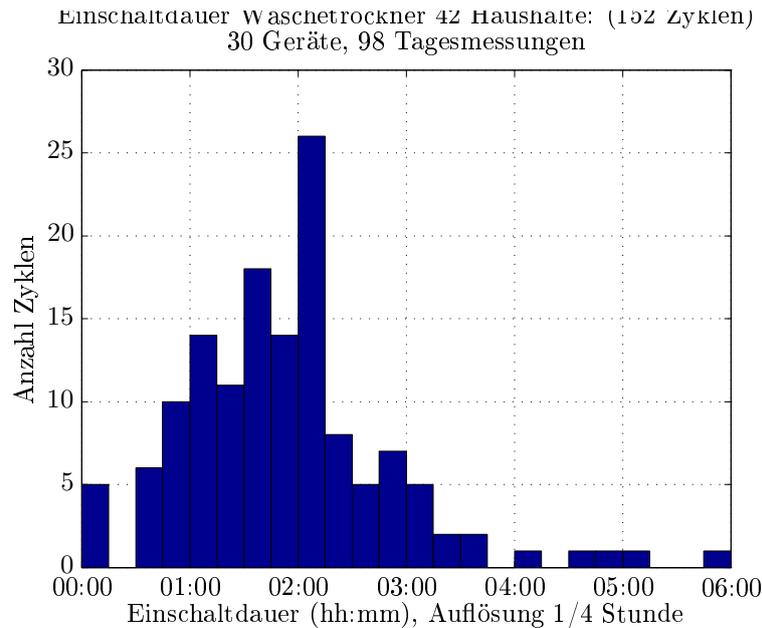


Abbildung 4.39.: Programmlaufzeit von Wäschetrocknern

aufweisen, was auf einen sehr ähnlichen Leistungsverlauf während eines Zyklus und auf eine annähernd gleiche Programmlaufzeit schließen lässt, da sich bei gravierenden Unterschieden in diesen beiden Werten, die Streuung des Durchschnittswertes auch deutlich verändern müsste.

Man erwartet pro Tag etwa die gleiche Einschalthäufigkeit wie bei Waschmaschinen, da diese beiden Geräte im Normalfall hintereinander betrieben werden und ein einzelner Einsatz eines dieser beiden Geräte nicht erwartet wird. In Abbildung 4.42 sieht man diese Verteilung, die ein deutliches Maximum bei 1 Zyklus/Tag besitzt und einen Durchschnittswert von 1.54 aufweist.

Für einen Vergleich des Energieverbrauchs wurden wieder die Energieeffizienzklassen nach [12] und eine Trocknermenge von 6kg angenommen. Da es einige unterschiedliche Typen von Wäschetrocknerkonzepten gibt, ist ein Vergleich ohne genauere Informationen nur bedingt möglich. So sind zum Beispiel die Energieeffizienzklassen für Kondensationstrockner und Abluftrockner unterschiedlich definiert. Weiters wird der Energieverbrauch im Standardprogramm pro kg Wäschemenge angegeben und daher ist der Vergleich natürlich stark von der Menge an Wäsche im Trockner abhängig. Auch der Restwassergehalt der Wäsche hat einen großen Einfluss auf die Dauer und somit den Energieverbrauch eines Trockenvorganges. Also hat auch das verwendete Waschmaschinenprogramm einen indirekten Einfluss auf den Energieverbrauch des Trockners. In Abbildung 4.43 erkennt man, dass der Großteil der Trockner laut diesem Vergleich in Klasse A fallen würde, was meiner Einschätzung nach viel zu niedrig für ältere Geräte ist und ein bedeutender Teil der Trockner eine deutlich schlechtere Klasse haben müsste.

## 4. Auswertungen

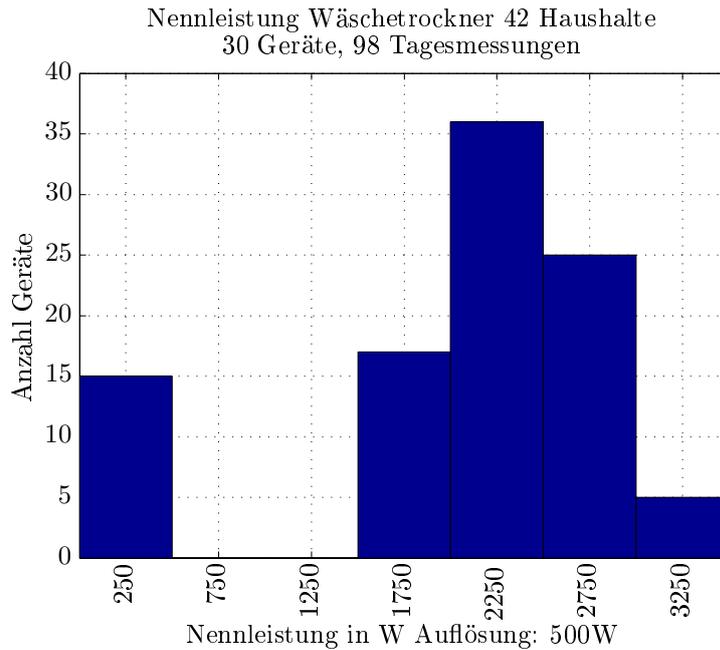


Abbildung 4.40.: Nennleistung von Wäschetrocknern

Dies lässt darauf schließen, dass wahrscheinlich weniger als 6 kg Wäsche im Trockner war und somit die Vergleichsklassen zu hoch angesetzt sind. Ohne weitere Informationen über die Wäschemenge oder den Gerätetypen ist dieser Vergleich nur als erster Anhaltspunkt zu betrachten und kann nicht als Begründung für eine allgemeine Aussage der Art „Ein Großteil der Haushalte hat einen Trockner der Energieeffizienzklasse A in Verwendung!“ herangezogen werden.

In Abbildung 4.44 ist ein typischer Zyklus eines Wäschetrockners dargestellt. Man erkennt, dass ein Trockner im Gegensatz zur Waschmaschine immer wieder eine Leistungsaufnahme von rund 2.5kW mit einer anschließenden kurzen Pause hat. Der hier abgebildete Zyklus dauert etwa 2.5 Stunden und gehört somit zu den eher längeren Zyklen.

In dieser Abbildung ist auch der Blindleistungsverlauf dargestellt. Man erkennt wieder, dass ein Großteil des Verbrauchs auf ohmsche Heizelemente zurückzuführen ist und daher kaum eine Änderung in der Blindleistungsbilanz erkennbar ist.

Interessant wäre es, ob sich ein Unterschied in der Verwendung von Trocknern im Sommer und im Winter erkennen lässt. Erwarten würde man, dass im Sommer bevorzugt Wäsche aufgehängt wird und im Winter häufiger der Trockner zum Einsatz kommt. Dieser Vergleich ist in Abbildung 4.45 dargestellt. Man erkennt keine deutlichen Änderungen im Benutzungsverhalten von Trocknern, was wahrscheinlich auf den höheren Komfort und den geringeren Arbeitsaufwand beim Einsatz von Trocknern zurückzuführen

## 4. Auswertungen

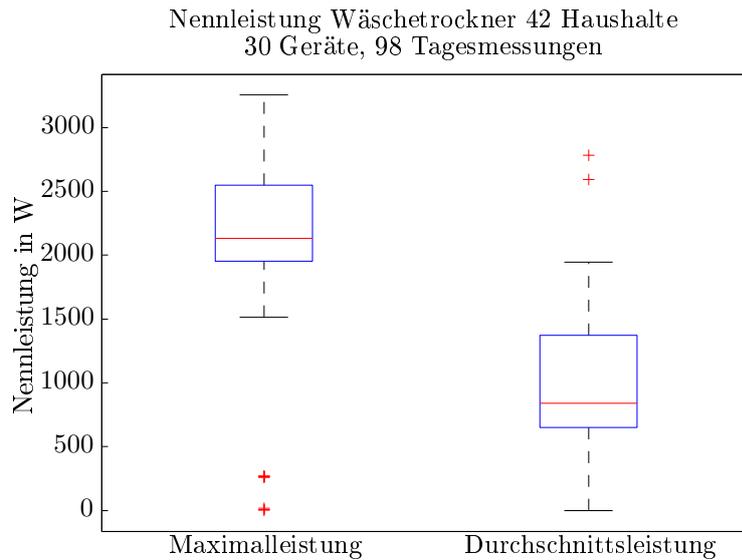


Abbildung 4.41.: Nennleistung von Wäschetrocknern - Boxplot

ren ist. Daraus ergibt sich auch, dass hier durchaus Einsparungspotential vorliegt, da im Sommer die Verwendung von Trocknern nicht unbedingt erforderlich wäre, im Winter aber durchaus als gerechtfertigt betrachtet werden kann.

In Abbildung 4.46 wird wieder der mittlere Leistungsverlauf pro Person über den Tag verteilt dargestellt. Man erkennt deutlich, dass die typische Verwendungszeiten von Trocknern zwischen 08:00 und 21:00 liegen. Der höchste Verbrauch liegt zwischen 12:00 und 15:00, also in einem Zeitraum, in dem eine Photovoltaikanlage bei entsprechenden Wetterverhältnissen den höchsten Ertrag liefern würde. Der direkte Vergleich mit dem Verlauf aus der Literatur ([9]) macht deutlich, dass in diesem Fall die deutlich ausgeprägte Spitze des Vergleichsverlaufes deutlich verschliffen wird und somit der Nutzungszeitpunkt hier eine deutlichere Streuung aufweist als in der zitierten Quelle.

## 4. Auswertungen

Einschaltzyklen pro Tag Wäschetrockner 42 Haushalte: (98 Tage)  
30 Geräte, 98 Tagesmessungen

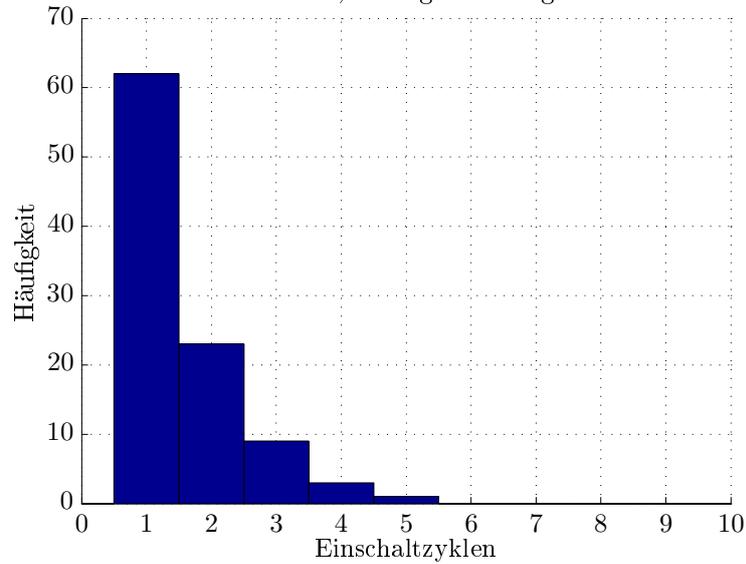


Abbildung 4.42.: Einschaltzyklen / Tag für Wäschetrockner

Energieverbrauch Wäschetrockner pro Zyklus, 42 Haushalte:

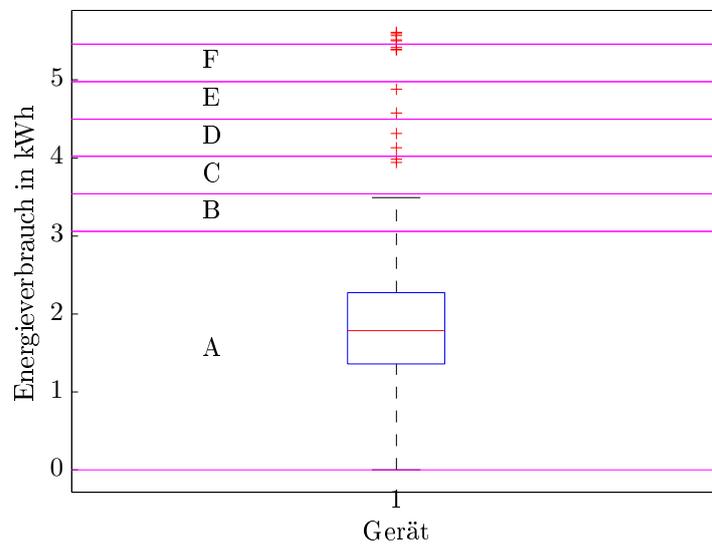


Abbildung 4.43.: Energieverbrauch von Wäschetrocknern mit Referenzklassen nach [12]

#### 4. Auswertungen

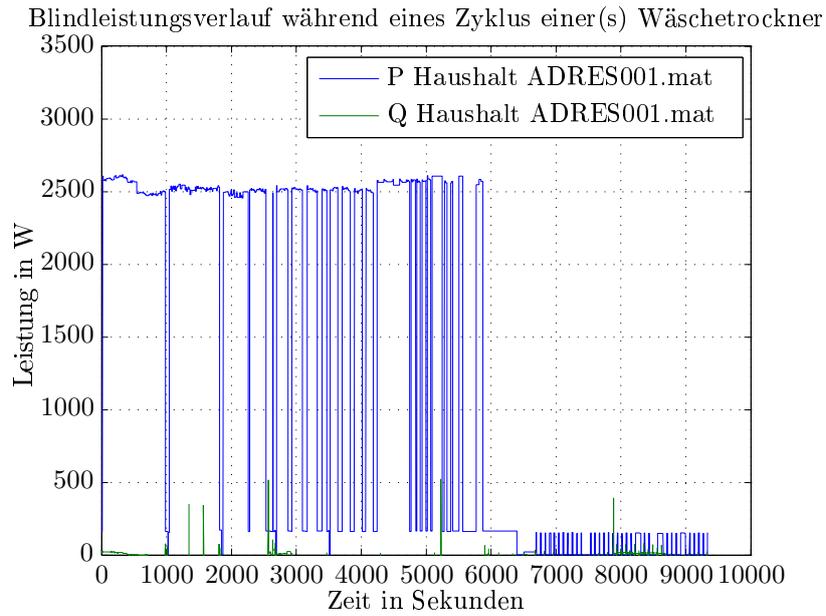


Abbildung 4.44.: Leistungsaufnahme während eines Zyklus von Wäschetrocknern

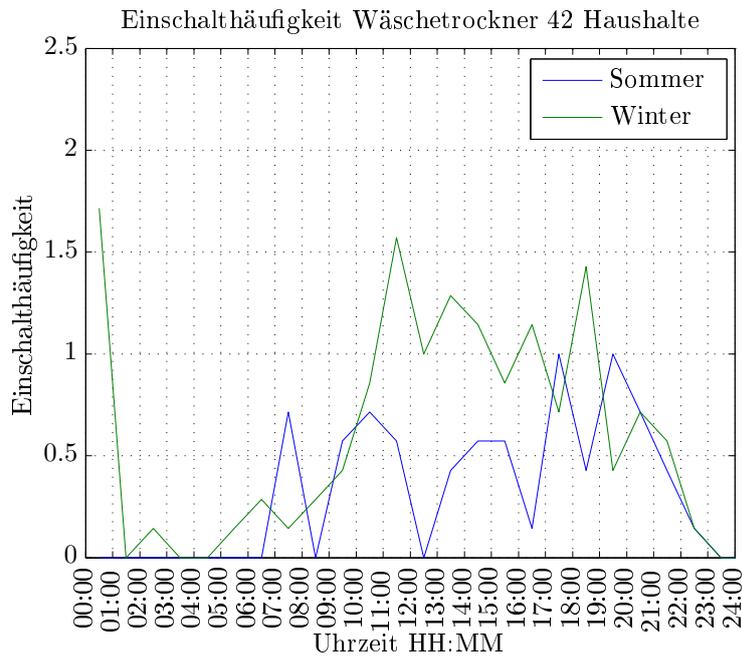


Abbildung 4.45.: Vergleich der Verwendung von Trocknern zwischen Sommer und Winter

## 4. Auswertungen

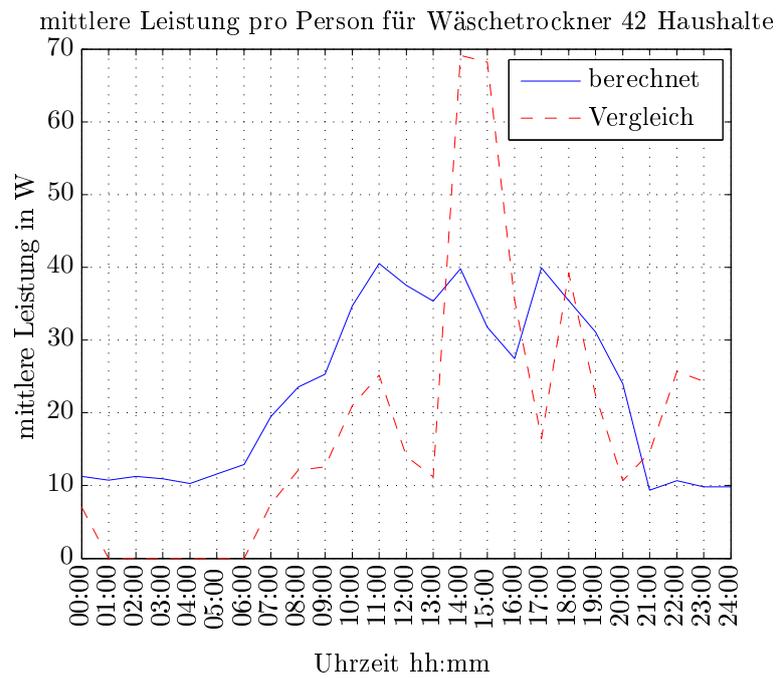


Abbildung 4.46.: mittlere Tagesleistung für Wäschetrockner mit Vergleichsverlauf nach [9]

#### 4.1.4. Geschirrspüler

Das nächste durchaus interessante Haushaltsgerät, das immer häufigeren Einsatz in den Haushalten findet und daher eine genaue Betrachtung durchaus von Bedeutung ist, ist der Geschirrspüler. Er genießt mittlerweile eine Verbreitung von 74 Prozent (Quelle [3]) in den österreichischen Haushalten und erfreut sich somit auf Grund des hohen Komforts sehr großer Beliebtheit.

Wie zu erwarten war, wird der Geschirrspüler vorwiegend nach den drei großen Mahlzeiten, also morgens, mittags oder abends in Betrieb genommen. Auch hier gilt wie vorher schon bei der Waschmaschine oder beim Trockner festgestellt auf Grund der geringen Anzahl an verfügbaren Zyklen, dass eine feinere Zeitauflösung als die hier dargestellte mit 1 Stunde-Schritten das Ergebnis verfälschen würde. In Abbildung 4.47 erkennt man wie beschrieben die drei zu erwartenden Spitzen im Gebrauchsverhalten von Geschirrspülern.

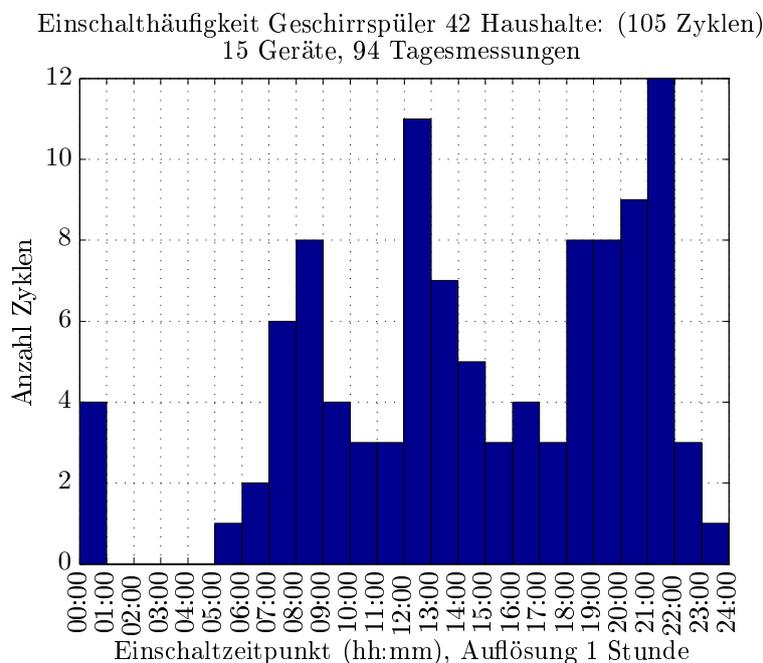


Abbildung 4.47.: Einschaltzeitpunkt von Geschirrspülern

In Abbildung 4.48 ist die typische Laufzeit von Geschirrspülern dargestellt, die im Bereich von 1-3 Stunden liegt. Die mittlere Laufzeit beträgt 1:42 Stunden und entspricht somit der persönlichen Erfahrung beim Einsatz eines Geschirrspülers. Im nächsten Schritt wurde wieder die Nennleistung für Geschirrspüler bestimmt, die erwartungsgemäß bei rund 2.500 W liegen sollte. Wie in Abbildung 4.49 ersichtlich, stimmt diese Erwartung mit den Ergebnissen recht gut überein. Der durchschnittliche Leistungswert für alle Geschirrspüler liegt bei 2.478 W. Es gibt nur einige wenige Geräte mit einer deutlich

## 4. Auswertungen

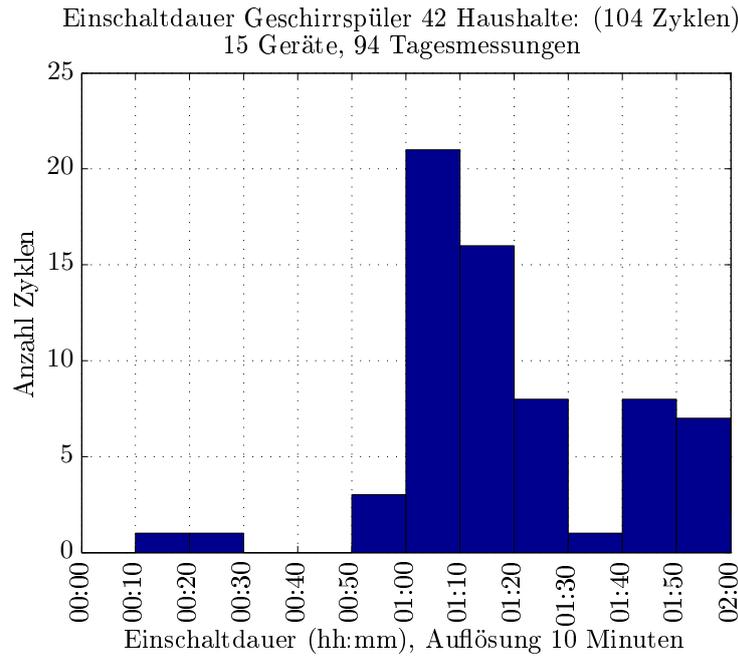


Abbildung 4.48.: Programmlaufzeit von Geschirrspülern

abweichenden Leistung, aber diese Werte sind auf fehlerhafte Zyklen zurückzuführen. In Abbildung 4.50 ist die geringe Streuung der Nennleistung von Geschirrspülern noch besser zu erkennen.

Hier bietet sich auch wieder an, den Energieverbrauch je Zyklus zu vergleichen und die Energieeffizienzklassen in diesem Vergleich darzustellen. Zur Ermittlung des Energiewertes der jeweiligen Klasse wurde wieder [12] herangezogen und mit der dort verwendeten Notation ergeben sich die jeweiligen Klassen mit der Annahme von 10 Maßgedecken ( $S = 10$ ) laut folgender Formel:

$$C_R = 1.35 + 0.025 \cdot S = 1.6 \quad (4.1)$$

$$C = E_I \cdot C_R = E_I \cdot 1.6 \quad (4.2)$$

Natürlich ist auch hier eine tatsächliche Zuordnung zur jeweiligen Klasse unter anderem vom Nutzinhalt des betrachteten Geschirrspülers abhängig. Bei modernen Geräten gibt es weiters eine Vielzahl an verschiedenen Programmen zur Auswahl. Außerdem gibt es mittlerweile Geräte, die die Programmlaufzeit von der tatsächlichen Beladung abhängig gestalten. Auf Grund dieser unbekanntenen Einflussfaktoren auf den tatsächlichen Leistungsverbrauch ist die in Abbildung 4.49 getroffene Zuordnung ein Versuch, solch eine Zuordnung zu treffen und die tatsächliche Aussagekraft dieses Diagramms muss stets kritisch betrachtet werden.

## 4. Auswertungen

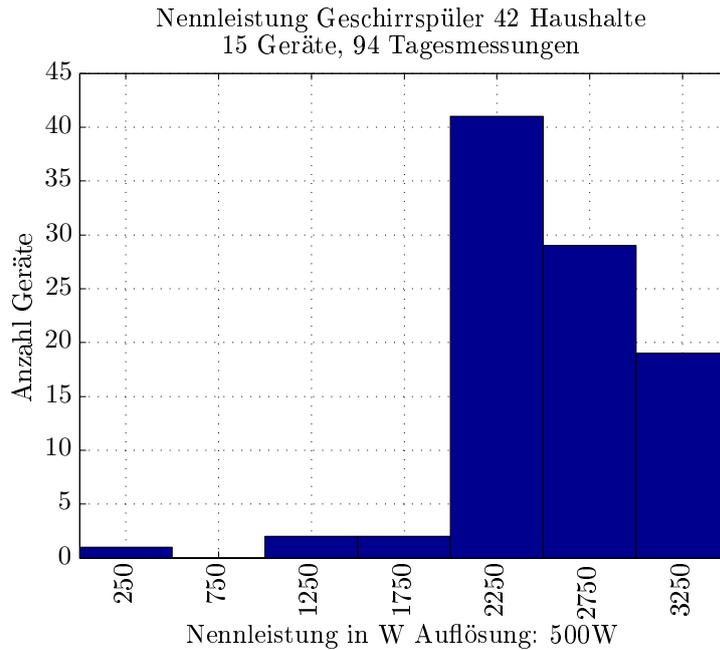


Abbildung 4.49.: Nennleistung von Geschirrspülern

Die Einschalthäufigkeit für Geschirrspüler pro Tag wird in folgender Abbildung (4.52) dargestellt. Die maximale Zahl an Zyklen/Tag liegt bei 3, die durchschnittliche bei 1.1.

In einem weiteren Schritt wird der Leistungsverlauf eines typischen Geschirrspülers dargestellt und die markanten Bereiche näher beschrieben. In den Abbildungen 4.53, 4.54 sowie 4.55 sind drei Zyklen dargestellt. Man erkennt am Beginn jedes Zyklus einige Heizphasen mit einer Leistung von 2-3 kW und Dauern im Bereich von 7 - 15 Minuten. Danach gibt es je nach Gerät einen Rückgang im Leistungsverbrauch, der auf eine Spülung des Geschirrs mit dem vorgeheizten Wasser zurückzuführen ist. Gefolgt wird dies meist von einer erneuten Aufheizphase, also einem Nachspülen des Geschirrs ohne Zugabe von Reinigungsmitteln. Beim Verlauf in Abbildung 4.54 erkennt man, dass der erste Waschvorgang mehrmals wiederholt wird, woraus sich schließen lässt, dass hier ein Programm für stark verschmutztes Geschirr verwendet wurde. In Abbildung 4.55 sind drei kurze Leistungsspitzen am Ende des Waschganges erkenntlich, die auf eine Trocknung des Geschirrs schließen lassen. Eine Vermutung ist, dass hier eine Art Heißluftgebläse in Verwendung ist, da die Leistungsaufnahme doch deutlich von den vorangegangenen Heizphasen abweicht und daher auf einen anderen Verbraucher schließen lässt. Weiters erkennt man eine Einschaltspitze, die auf einen Motor schließen lässt. Mit einer Leistung von rund 2.5 kW ist dieser Wert allerdings für einen reinen Motor zu hoch. Daher die Vermutung, dass zusätzlich ein Heizelement in Betrieb ist und somit von einem Heißluftgebläse ausgegangen werden kann.

## 4. Auswertungen

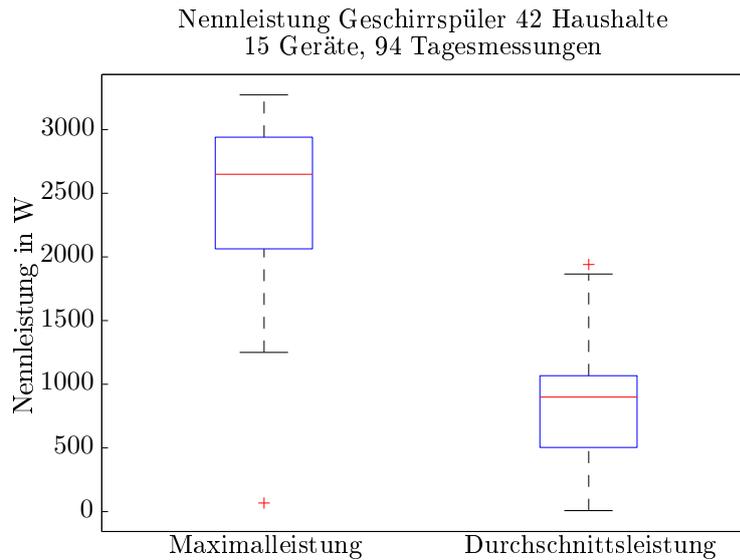


Abbildung 4.50.: Nennleistung von Geschirrspülern - Boxplot

Alle drei dargestellten Zyklen haben eine gesamte Laufzeit von etwas mehr als einer Stunde und liegen somit im Durchschnitt aller Zyklen.

Wie zu erwarten war, ist der Blindleistungsverlauf eines Geschirrspülers nicht weiter markant, da ein Großteil der Leistungsaufnahme durch einen ohmschen Verbraucher, nämlich einen Heizstab, entsteht und daher keine Blindleistung aus dem Netz benötigt wird.

Als nächstes wird die mittlere Tagesleistung für Geschirrspüler dargestellt. In Abbildung 4.56 erkennt man klar die vorher schon beschriebenen Spitzen in der Leistungsaufnahme nach den drei großen Mahlzeiten. Auch hier wird wieder der Vergleich mit dem in [9] verwendeten Verlauf angeführt. Man erkennt hier, dass in dieser Messreihe die Morgenspitze deutlicher ausgeprägt ist, dafür aber die Mittagsspitze abgeschwächt erscheint. Die Position der Spitzen ist ziemlich ähnlich in beiden Verläufen.

## 4. Auswertungen

Energieverbrauch Geschirrspüler pro Zyklus, 42 Haushalte: (116 Werte)

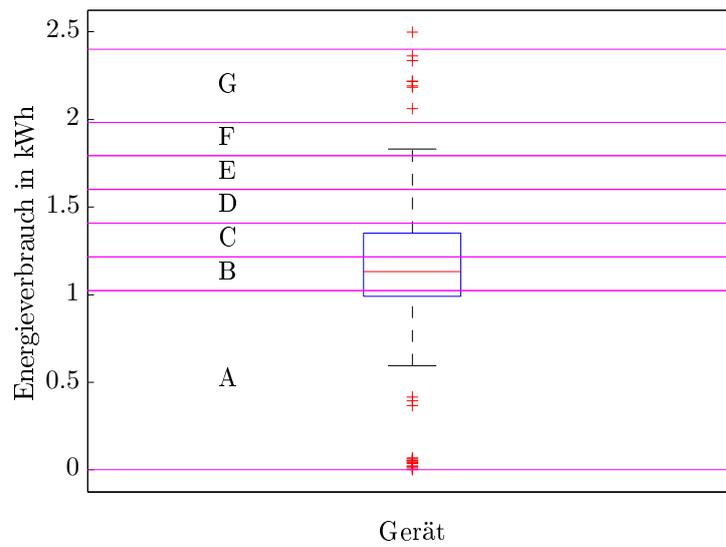


Abbildung 4.51.: Energieverbrauch von Geschirrspülern mit den Energieeffizienzklassen nach [12]

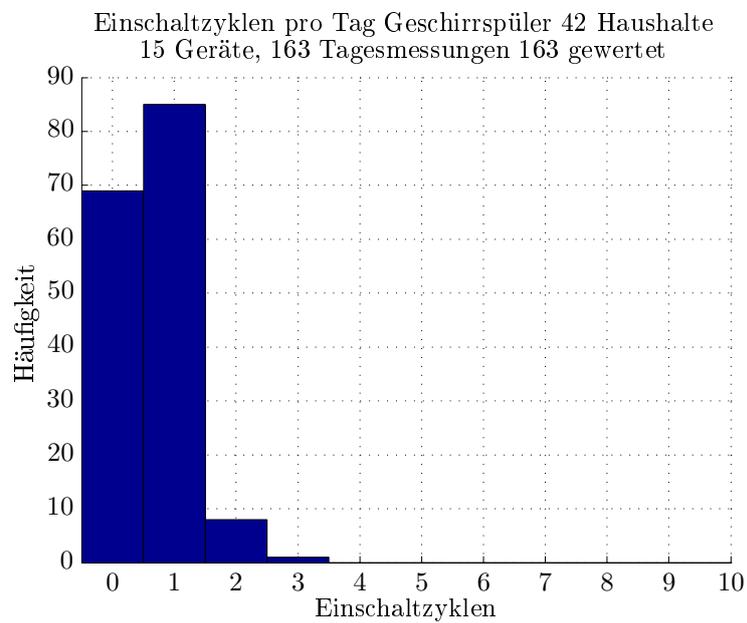


Abbildung 4.52.: Einschaltzyklen pro Tag

#### 4. Auswertungen

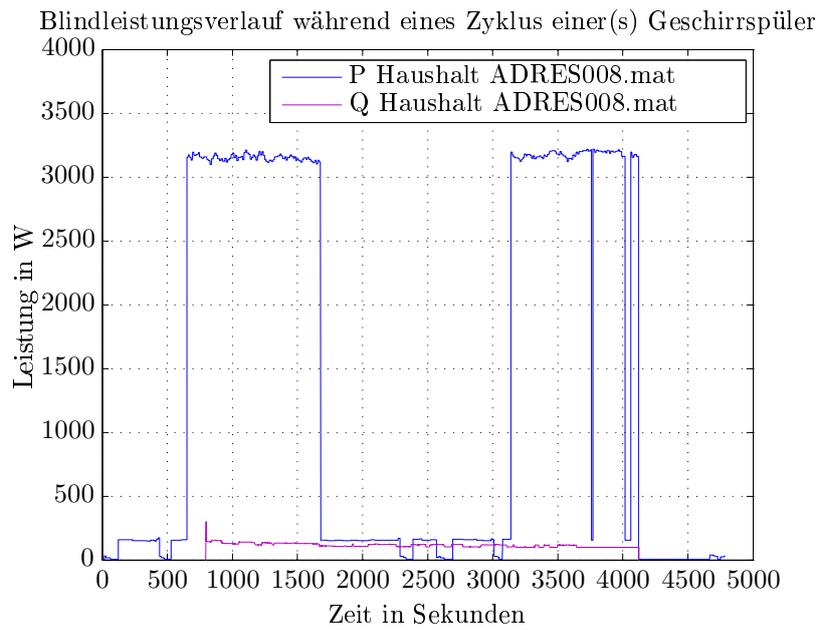


Abbildung 4.53.: Leistungsverlauf eines Geschirrspülers

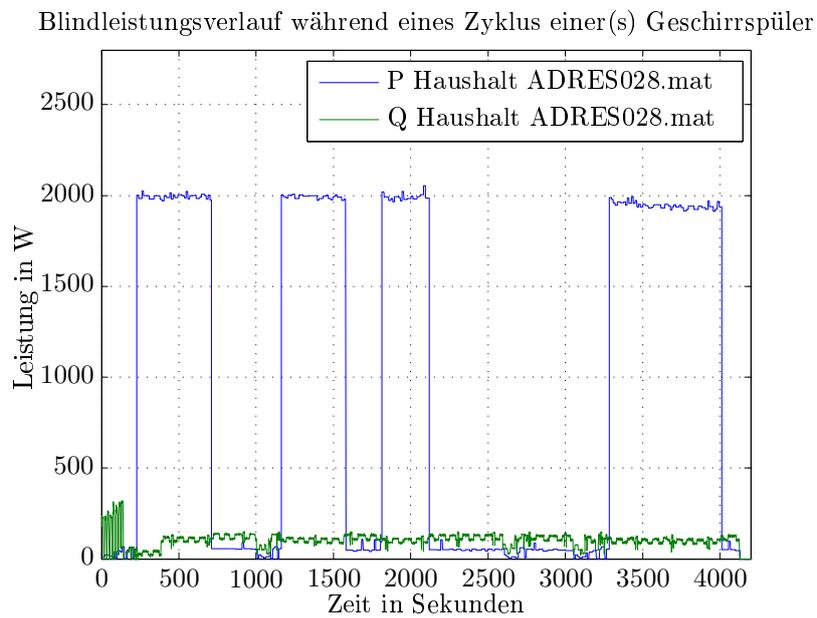


Abbildung 4.54.: Leistungsverlauf eines Geschirrspülers

#### 4. Auswertungen

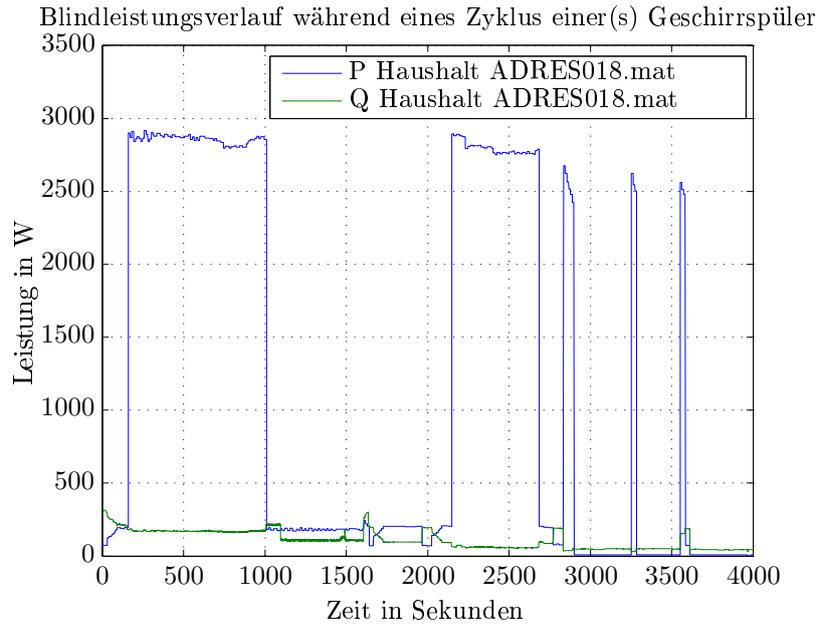


Abbildung 4.55.: Leistungsverlauf eines Geschirrspülers

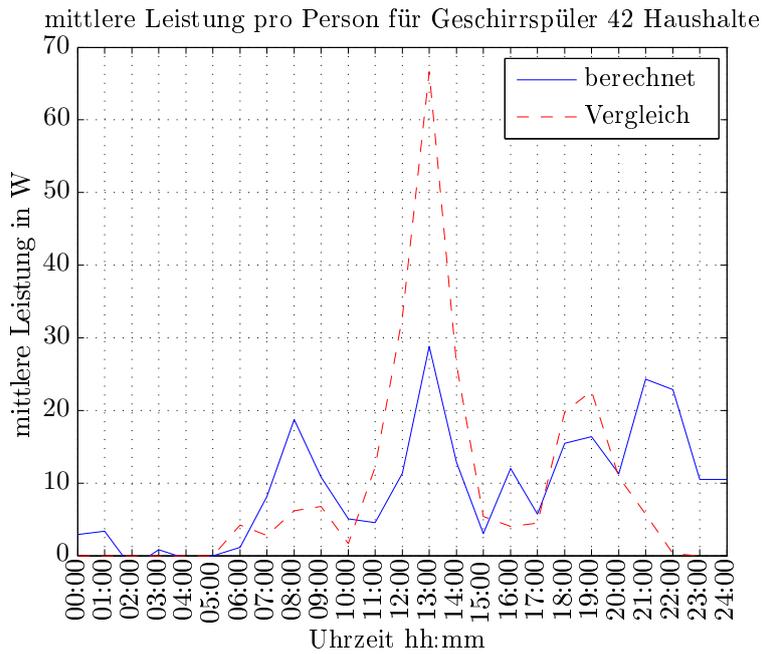


Abbildung 4.56.: mittlerer Tagesleistungsverlauf eines Geschirrspülers mit dem Vergleichsverlauf nach [9]

## 4. Auswertungen

### 4.1.5. Küchengeräte

Als erstes Küchengerät wurde der Wasserkocher untersucht. Hier erwartet man einen verstärkten Einsatz in den Morgenstunden, also bei der Zubereitung des Frühstücks. Dies ist besonders deutlich in Abbildung 4.57 erkennbar. Weiters erkennt man eine kleine Abendspitze um ca 18:00, also beim Abendessen.

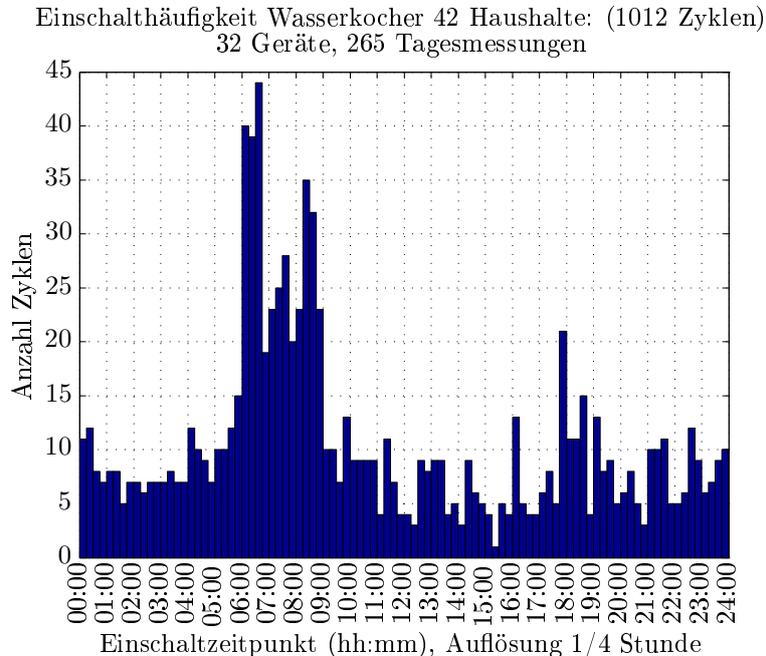


Abbildung 4.57.: Einschaltzeitpunkte für Wasserkocher

Für Wasserkocher erwartet man weiters eine relativ kurze Einschaltdauer von unter 10 Minuten. Rechnerisch ergibt sich eine durchschnittliche Betriebsdauer von 1 Minute und 52 Sekunden, also deutlich unter der erwarteten Dauer. Die Verteilung ist in Abbildung 4.58 besonders markant. Man sieht ein deutliches Maximum in der Klasse 0-5 Minuten und einige wenige Ausreißer deutlich darüber. Weiters ergibt sich für Wasserkocher eine durchschnittliche Leistung von rund 1.8 kW sowie eine maximale Anschlussleistung von 2.8 kW, was sehr gut mit der Erfahrung übereinstimmt. Die gängigsten Geräte haben eine Anschlussleistung von 1.5 - 2.5kW, je nach Fassungsvermögen des Gerätes. Die Verteilung der Anschlussleistung ist in Abbildung 4.59 sowie in 4.60 dargestellt. Hier liegen die maximale sowie durchschnittliche Verteilung sehr knapp beieinander, da diese Geräte eine konstante Leistungsaufnahme haben, die auch relativ unabhängig von der Betriebsdauer ist.

Der durchschnittliche Energieverbrauch pro Zyklus für einen Wasserkocher liegt bei

## 4. Auswertungen

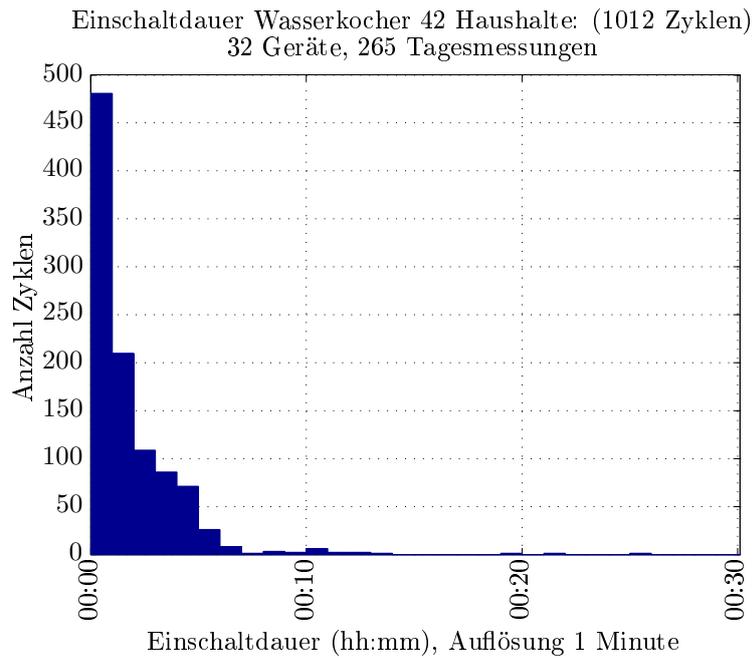


Abbildung 4.58.: Einschaltdauer für Wasserkocher

430 Wh, der maximale Verbrauch bei etwa 1 kWh, was im Verhältnis zur Nennleistung und zur erwarteten Betriebsdauer zu hoch ist und daher auf einen Fehler in der Messung hindeutet.

Auch für Wasserkocher wurde der durchschnittliche Leistungsverbrauch über einen Tag verteilt in Abbildung 4.62 dargestellt. Man erkennt wieder die beiden Spitzen in den Morgen-, und Abendstunden.

#### 4. Auswertungen

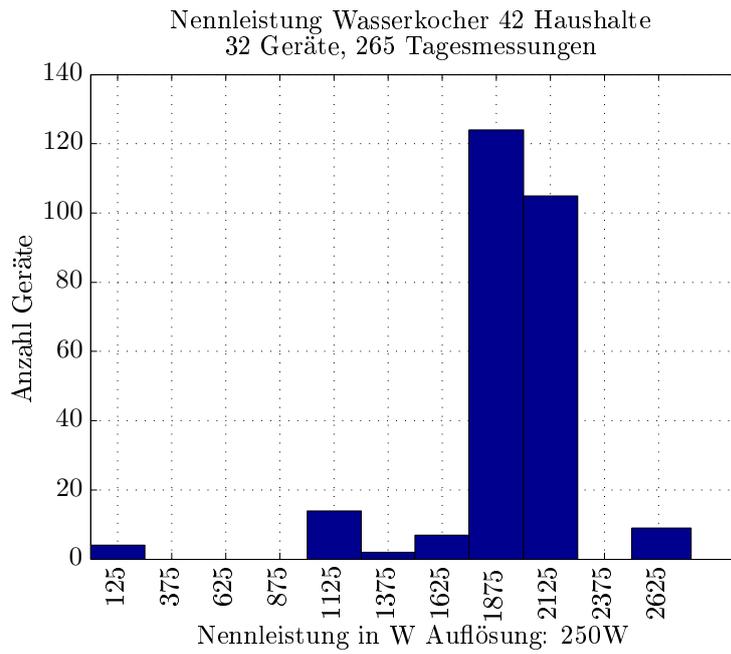


Abbildung 4.59.: Nennleistung für Wasserkocher

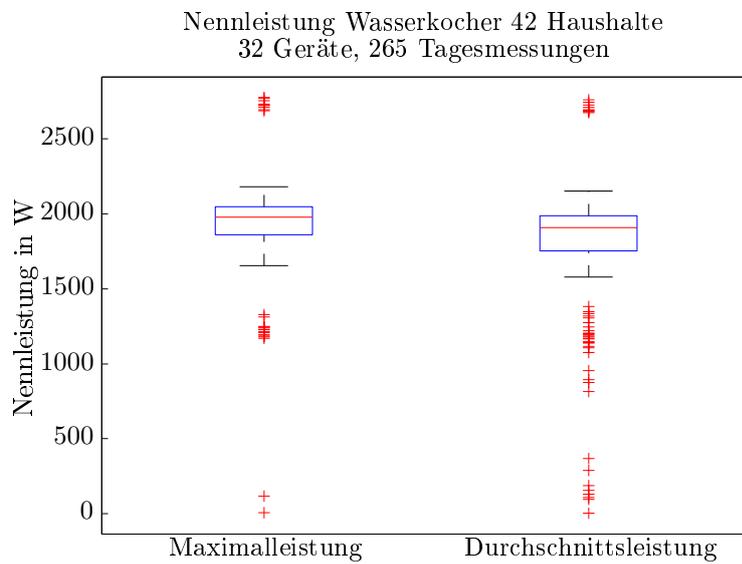


Abbildung 4.60.: Nennleistung für Wasserkocher - Boxplot

## 4. Auswertungen

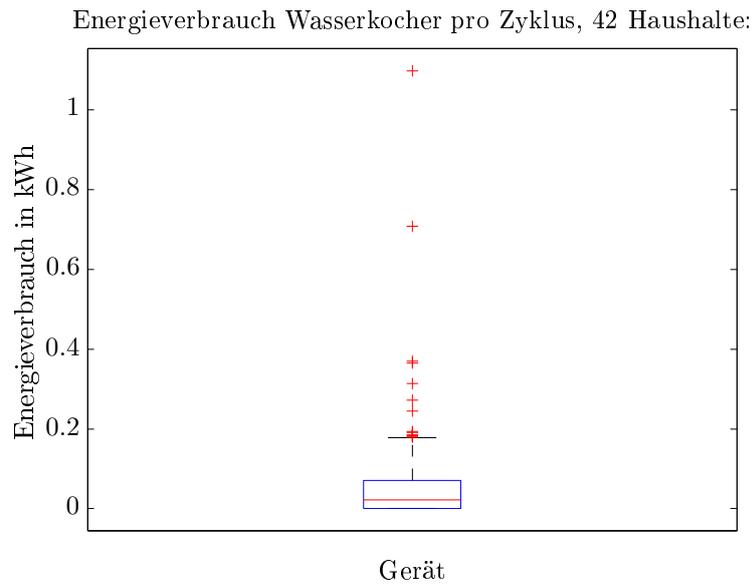


Abbildung 4.61.: Energieverbrauch pro Zyklus für Wasserkocher

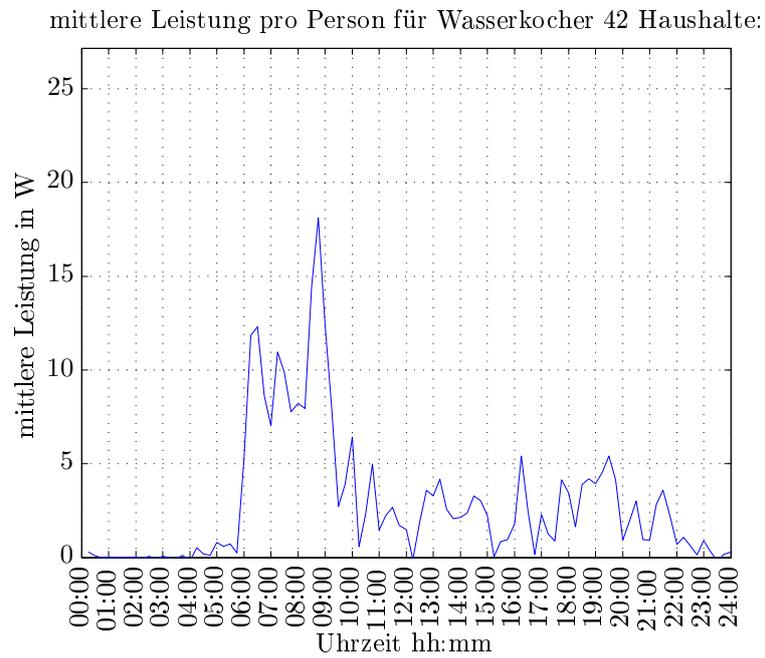


Abbildung 4.62.: mittlerer Tagesleistungsverbrauch für Wasserkocher

#### 4. Auswertungen

Als nächstes Küchengerät wurden Toaster näher beleuchtet. Auch hier erwartet man verstärkte Nutzung in den Morgen-, und Abendstunden, wie auch deutlich in Abbildung 4.63 klar erkennbar. Auch die Einschaltdauer liegt erwartungsgemäß im Bereich von 15 Minuten. Für die Nennleistung erwartet man eine kleine Streuung um den Durchschnittswert von knapp 1kW. Hier zeigt sich eine relativ große Streuung (siehe Abbildung 4.65, was auf die vielen unterschiedlichen Gerätetypen und damit verbundenen Leistungswerte erklärbar ist. Weiters lässt sich vermuten, dass an der Steckdose für Toaster auch teilweise andere Geräte im Einsatz waren und daher auch falsche Leistungswerte aufgezeichnet wurden. Der durchschnittliche Energieverbrauch je Zyklus, dargestellt in Abbildung 4.66, liegt bei etwa 73 Wh, der maximale Verbrauch bei etwa dem fünffachen Wert, also rund 370 Wh, was wiederum auf die große Streuung der Nennleistung zurückzuführen ist. Die Blindleistungsbilanz ist wie bei einigen vorher beschriebenen Geräten auf Grund des ohmschen Verbrauches eines Heizstabes wieder sehr gut, wie in Abbildung 4.67 durch kaum eine Veränderung in der Blindleistung bei doch recht großer Wirkleistung erkennbar. Weiters sieht man das typische Verhalten von thermostatgesteuerten Geräten, erkennbar durch mehrere hintereinander folgende Leistungsaufnahmen mit annähernd konstanter Leistung. Man erkennt in den Abbildungen 4.67 sowie 4.68, dass die Blindleistungsaufnahme des Haushaltes geringfügig sinkt, während die Heizelemente des Toasters in Betrieb sind. Dies ist wahrscheinlich auf eine leicht verminderte Netzspannung auf Grund der hohen Anschlussleistung zurückzuführen.

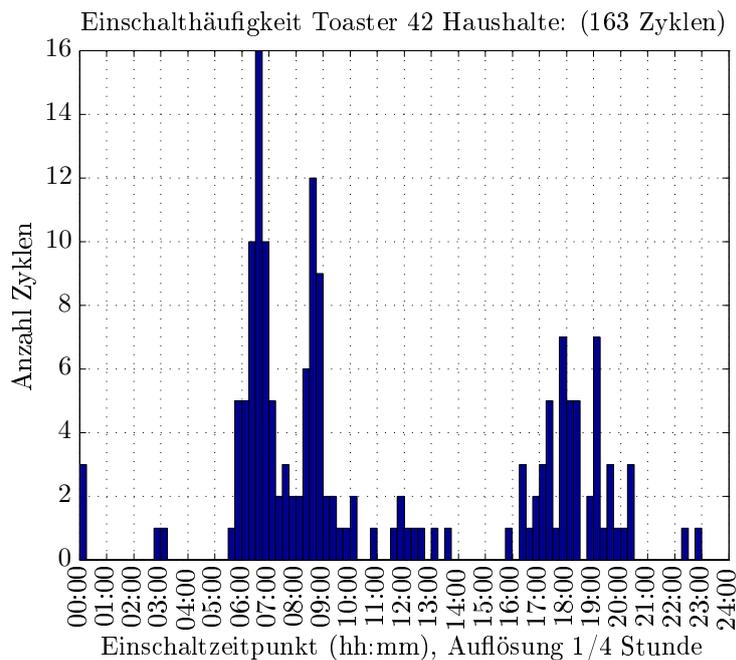


Abbildung 4.63.: Einschaltzeitpunkte für Toaster

#### 4. Auswertungen

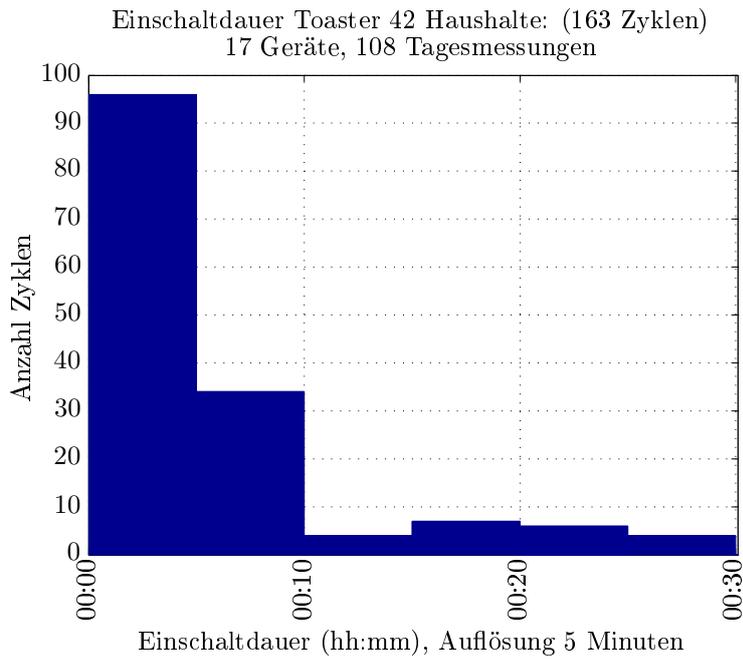


Abbildung 4.64.: Einschaltdauer für Toaster

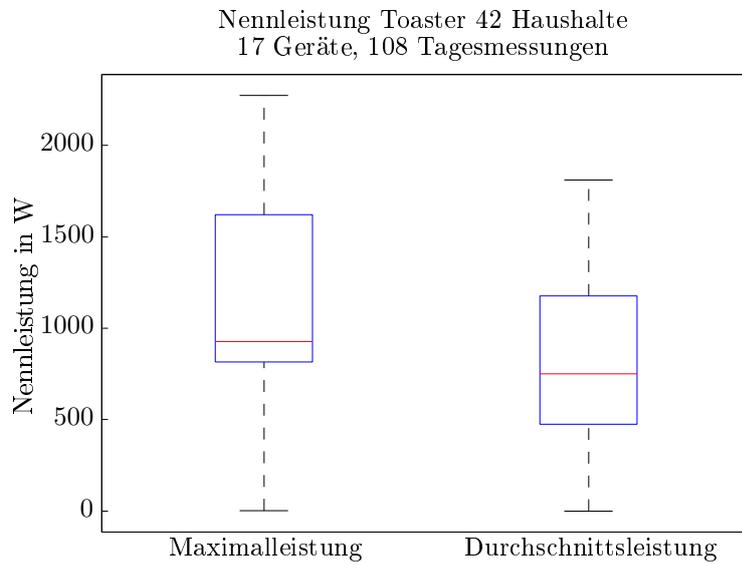


Abbildung 4.65.: Nennleistung für Toaster

## 4. Auswertungen

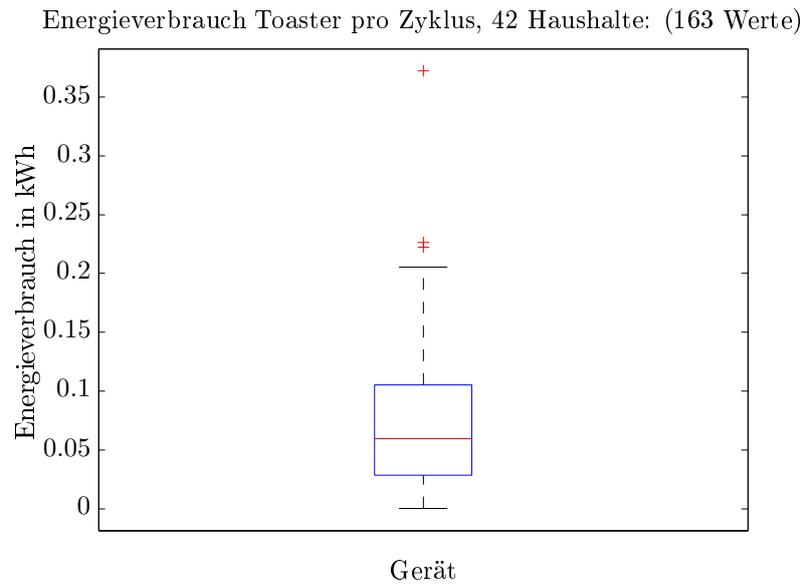


Abbildung 4.66.: Energieverbrauch pro Zyklus für Toaster

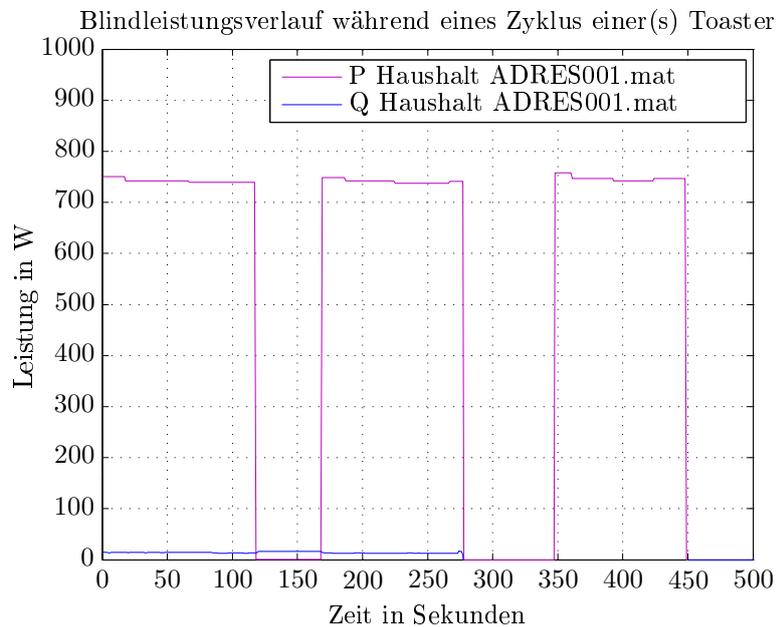


Abbildung 4.67.: Leistungsverlauf während eines Zyklus für Toaster

#### 4. Auswertungen

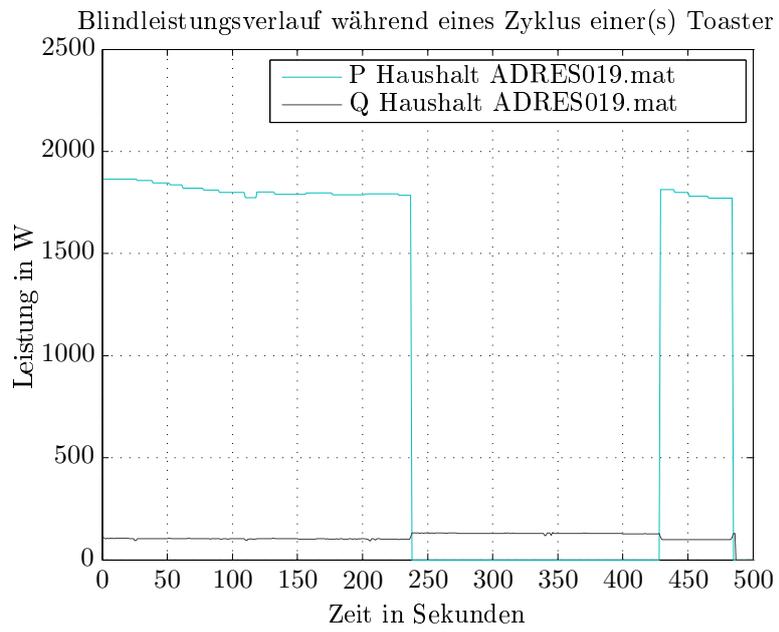


Abbildung 4.68.: Leistungsverlauf während eines Zyklus für Toaster

## 4. Auswertungen

Als letztes im Detail vermessenes Küchengerät wird die Mikrowelle näher beleuchtet. Man erwartet hier auch wieder verstärkten Einsatz um die Morgen-, Mittag-, und Abendstunden und dazwischen so gut wie keine Verwendung, besonders deutlich in Abbildung 4.69 erkennbar. Es zeigen sich deutlich Spitzen zwischen 08:00 - 09:00, 12:00 - 13:00 sowie 19:00 - 20:00.

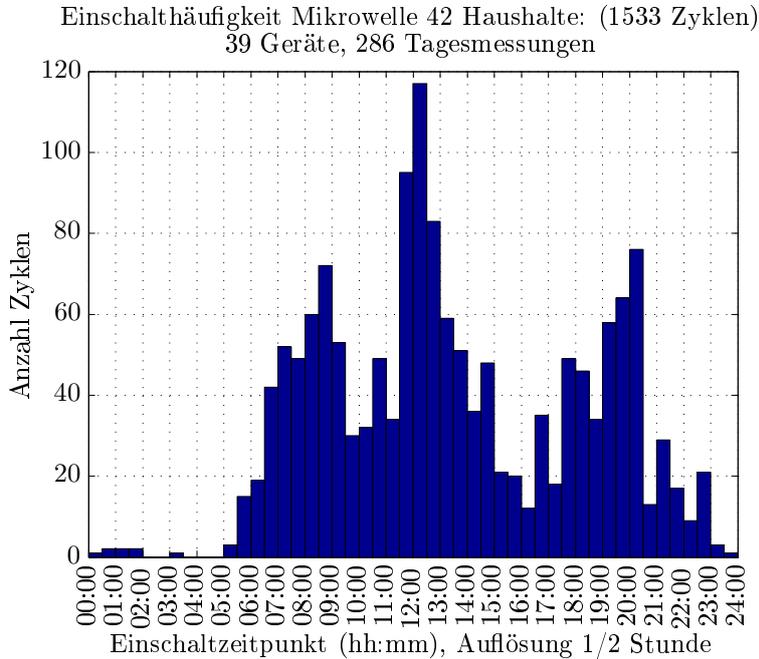


Abbildung 4.69.: Einschaltzeitpunkte für Mikrowellen

Man erwartet sich für Mikrowellen eine Einschaltdauer von maximal 10 Minuten und eine durchschnittliche Dauer, die deutlich darunter im Bereich von 1-2 Minuten liegt. Tatsächlich zeigt Abbildung 4.70, dass die Einschaltdauer im Bereich von 5 Minuten liegt. Es errechnet sich ein Durchschnittswert von 1:35 Minuten. Einige Ausreißer nach oben ergeben eine maximale Betriebsdauer von 4:32 Stunden, was klarerweise viel zu hoch ist und daher auf eine fehlerhafte Erkennung schließen lässt.

Der Leistungsbereich von Mikrowellen ist auf Grund der Einstellbarkeit der gewünschten Geräteleistung einer relativ großen Streuung unterworfen und der große Unterschied zwischen maximaler und durchschnittlicher Leistung während eines Zyklus lässt auf eine stark alternierende Leistungsaufnahme schließen. Man erkennt in Abbildung 4.71, dass der Mittelwert der maximalen Leistungsaufnahme bei rund 1.3 kW, und der Maximalwert bei ca. 1.7 kW liegt. Die durchschnittliche Leistungsaufnahme zeigt eine deutlich größere Streuung der ermittelten Werte und es ergeben sich hier ein Mittelwert von ca. 1 kW bei einem Maximalwert von 1.7 kW. Der typische Leistungsverlauf von Mikrowellen ist in den Abbildungen 4.72, 4.73 sowie 4.74 dargestellt. Es werden jeweils die Wirk-, und Blindleistung geplottet. Man erkennt Bereiche konstanter Wirkleistung und

#### 4. Auswertungen

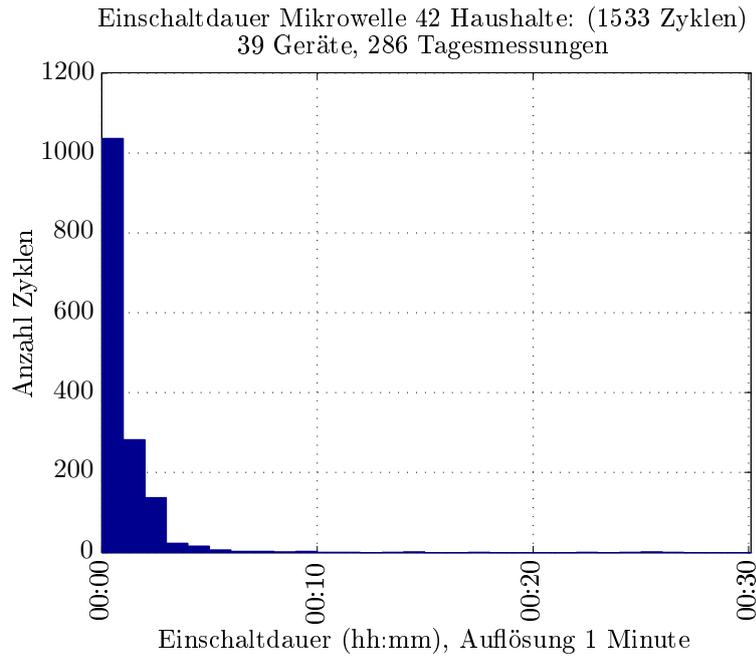


Abbildung 4.70.: Einschaltdauer für Mikrowellen

davor immer eine Spitze in der Blindleistungsaufnahme, was auf das Einschalten des Hochspannungstransformators mit angeschlossenem Magnetron und mit dem daher auftretenden induktiven Magnetisierungsstrom verbunden werden kann. In der Abbildung 4.73 sieht man zwei aufeinanderfolgende Zyklen, in Abbildung 4.74 sieht man mehrere kürzere gleich aussehende Verläufe. Daraus kann man schließen, dass die eingestellte Leistung bei Mikrowellen durch Modulation der Einschaltdauer des Magnetrons erreicht wird und nicht dessen Leistung geregelt wird, was einen höheren Aufwand bedeuten würde.

Interessant wäre es insbesondere bei Mikrowellen, ob man einen Unterschied in der Verwendung an Werktagen und Wochenenden erkennen kann. Man erwartet sich eventuell, dass bei einem Haushalt mit älteren Kindern beide Elternteile berufstätig sind und daher zu Mittag ein vorgekochtes Essen gewärmt wird. Diese Erwartung lässt sich aus der vorliegenden Messreihe nicht bestätigen, vielmehr zeigt sich, dass kaum ein Unterschied zwischen den beiden Tagestypen weder im Sommer (Abbildung 4.75) noch im Winter (Abbildung 4.76) erkennbar ist.

Die mittlere Tagesleistung ist in Abbildung 4.77 dargestellt. Man erkennt auch hier die deutlichen Spitzen am Morgen, zu Mittag und am Abend.

#### 4. Auswertungen

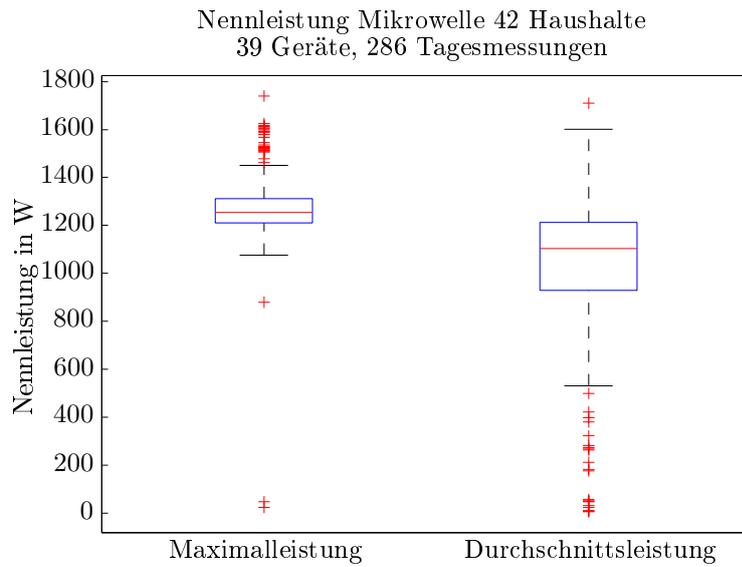


Abbildung 4.71.: Nennleistung für Mikrowellen

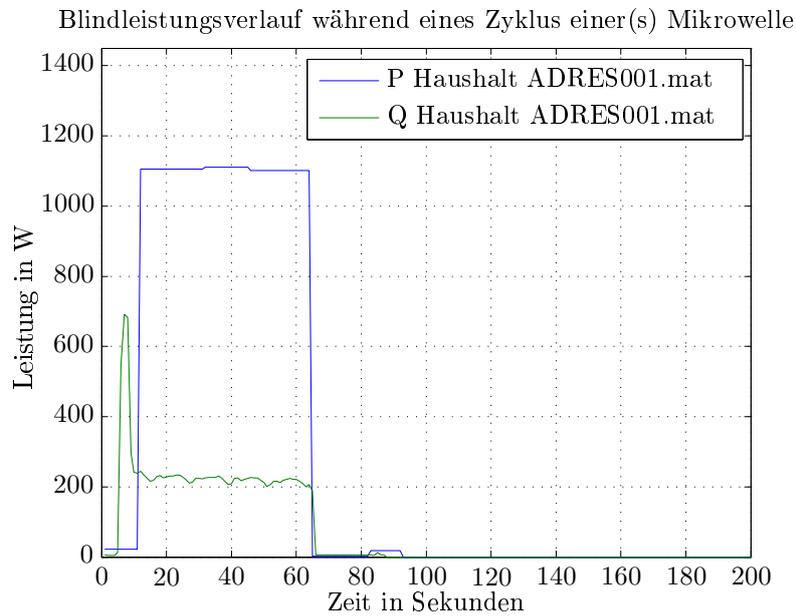


Abbildung 4.72.: Leistungsverlauf für Mikrowellen

#### 4. Auswertungen

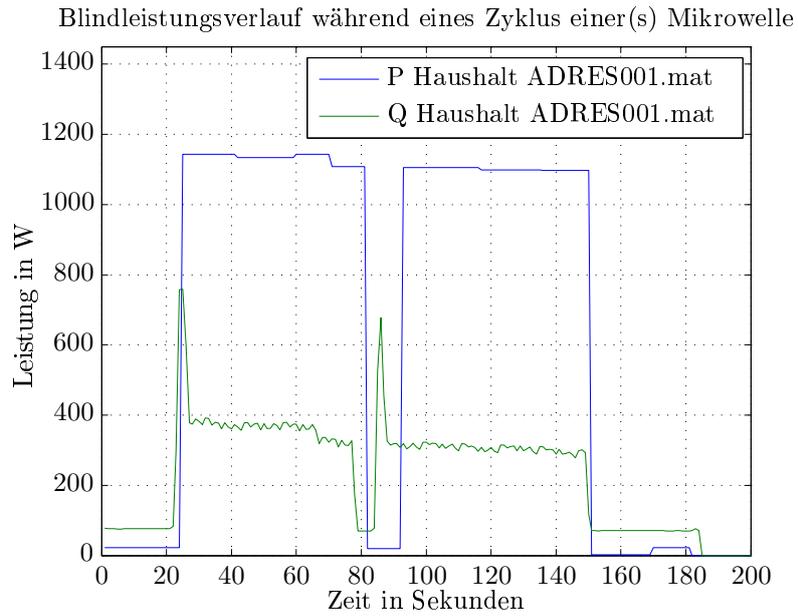


Abbildung 4.73.: Leistungsverlauf für Mikrowellen

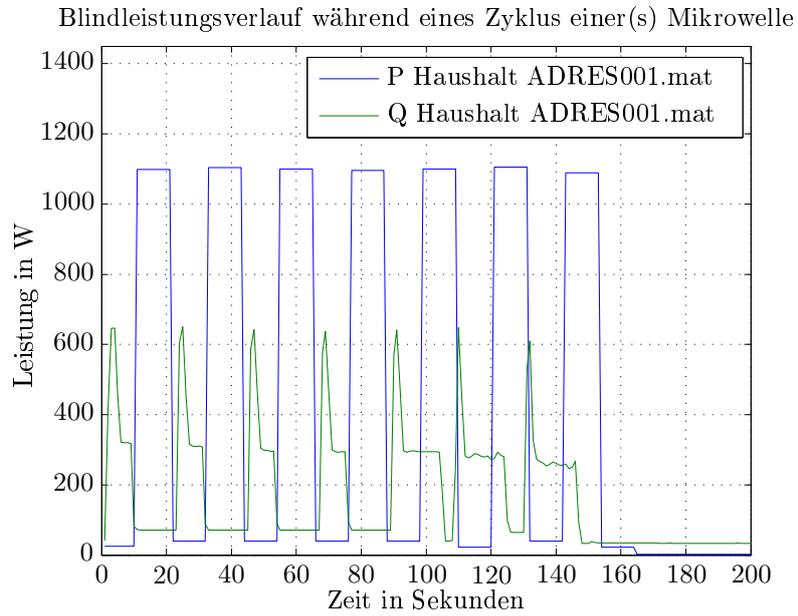


Abbildung 4.74.: Leistungsverlauf für Mikrowellen

#### 4. Auswertungen

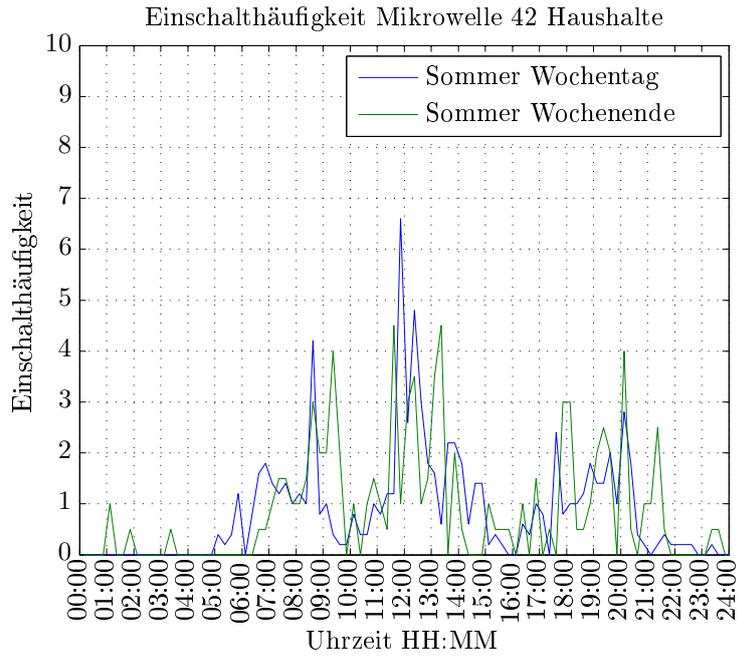


Abbildung 4.75.: Vergleich der Einschalthäufigkeit für Mikrowellen im Sommer

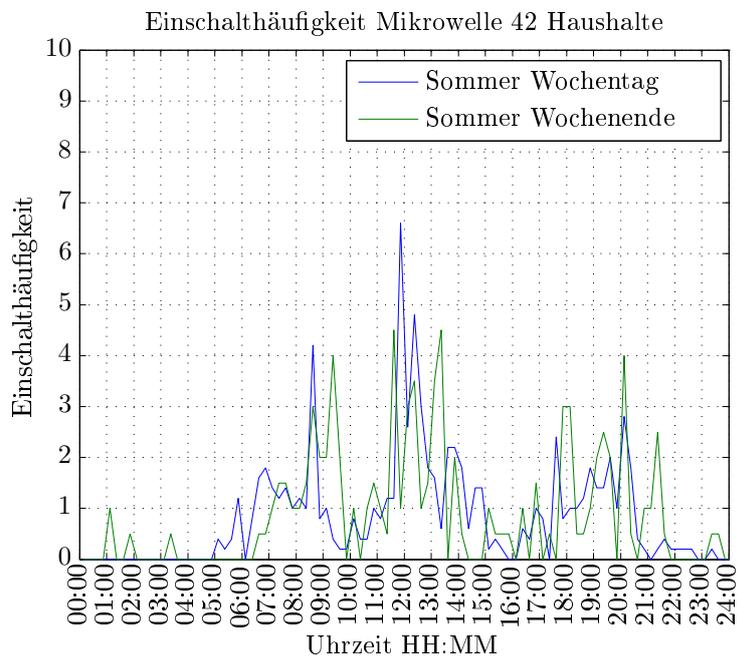


Abbildung 4.76.: Vergleich der Einschalthäufigkeit für Mikrowellen im Winter

## 4. Auswertungen

mittlere Leistung pro Person für Mikrowelle 42 Haushalte, 287 Zyklen

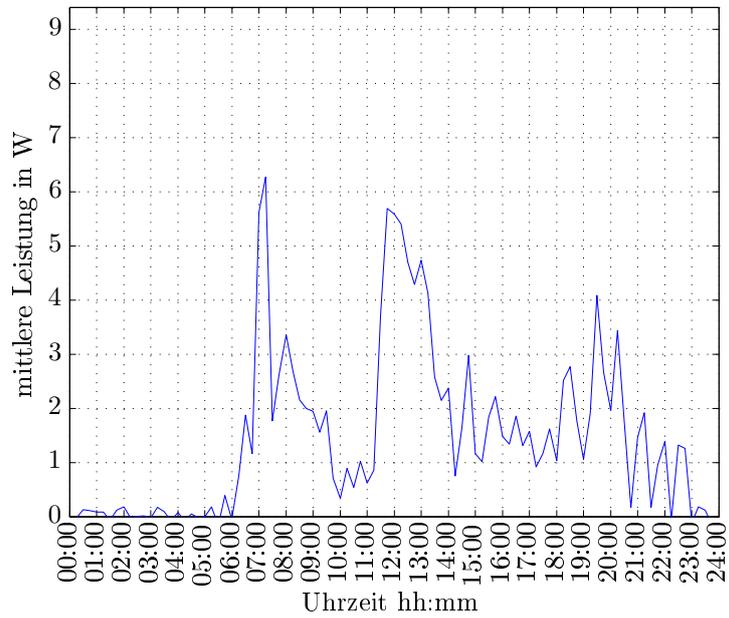


Abbildung 4.77.: mittlere Tagesleistung für Mikrowellen

## 4.1.6. Multimedia

In die Kategorie Multimedia fallen eine Reihe von Geräten, unter anderem auch solche, die eine relativ konstante Leistungsaufnahme während des gesamten Tages haben. Daher ist hier die Auswertung der Einschalthäufigkeit mit relativ vielen Zyklen beginnend um Mitternacht auch nicht wirklich aussagekräftig. Man erkennt in Abbildung 4.78 zwar einen deutliche Anstieg der Einschalthäufigkeit in den Abendstunden aber relativ zu der großen Anzahl an Zyklen beginnend um Mitternacht ist dieser Anstieg kaum signifikant. Daher wird nochmals der Verlauf der Einschaltzeitpunkte mit dem Ausschnitt bis maximal 200 Zyklen im Detail dargestellt (Abbildung 4.79).

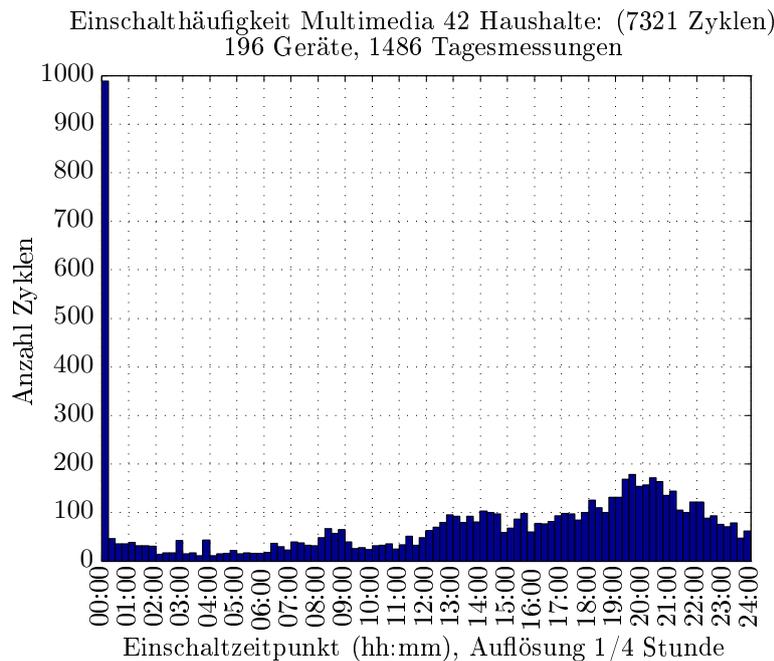


Abbildung 4.78.: Einschaltzeitpunkte für Multimedia-Geräte

Viel besser sieht man die verstärkte Nutzung von Multimediageräten in den Abendstunden in Abbildung 4.80, wenn man die mittlere Tagesleistung berechnet. Hier fallen die vielen Geräte mit konstanter Leistungsaufnahme auf Grund der geringeren Nennleistung nicht mehr so stark ins Gewicht und daher erkennt man die zu erwartende Abendspitze hier viel besser. Der Vergleich mit dem Verlauf aus [9] lässt deutlich die Abendspitze in beiden Verläufen erkennen. Der annähernd konstante Unterschied in den beiden Verläufen lässt sich darauf zurückführen, dass in dieser Arbeit die Stand-By Leistung nicht bei der Ermittlung des Tagesleistungsverlaufes berücksichtigt wurde, in der zitierten Quelle aber sehr wohl in Betracht gezogen worden ist.

Hier wäre auch ein Unterschied in der Nutzung im Sommer und Winter sowie am Wochenende und an Werktagen zu erwarten. In Abbildung 4.81 ist der Vergleich zwi-

## 4. Auswertungen

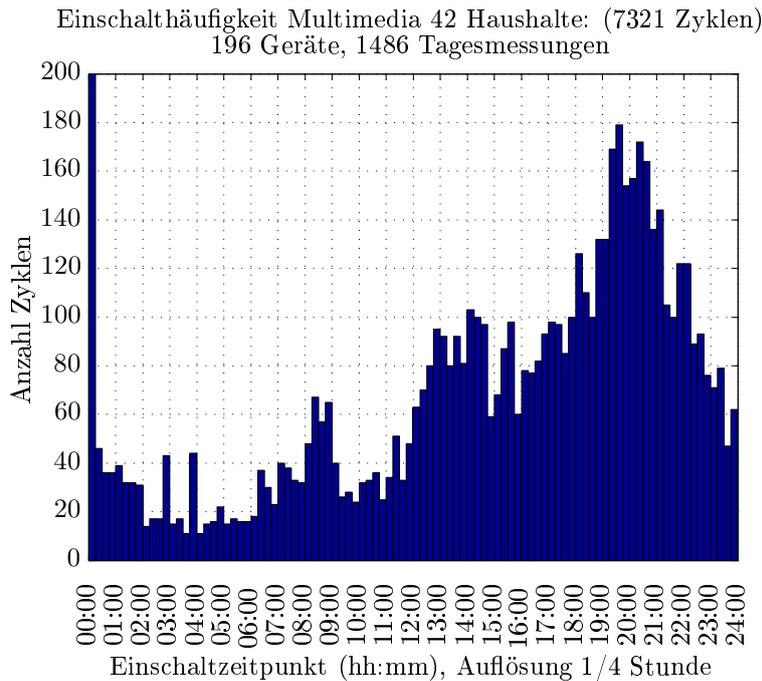


Abbildung 4.79.: Einschaltzeitpunkte für Multimedia-Geräte - Ausschnitt bis 200 Zyklen

schen Sommer-, und Winternutzung dargestellt. Man erkennt eine deutlich gesteigerte Nutzung im Winter im Vergleich zum Sommer, die wahrscheinlich auf die kältere Temperatur und den früheren Einbruch der Dunkelheit und damit die deutlich geringeren Aktivitäten im Freien zurückzuführen ist. Im Vergleich zwischen Werktagen und Wochenenden im Winter (Abbildung 4.82) erkennt man auch wieder eine verstärkte Nutzung am Wochenende, die beim Vergleich im Sommer (Abbildung 4.83) nicht erkennbar ist.

Wenn man die Nennleistung der Geräte in dieser Kategorie vergleicht, erkennt man auch wieder die große Streuung, siehe Abbildung 4.84. Dies ist auch darauf zurückzuführen, dass es sehr viele verschiedene Gerätetypen gibt, die alle in diese Kategorie fallen. Angefangen bei Kleingeräten, wie Küchenradios oder Routern bis hin zu den größeren Verbrauchern mit Fernseher und Computer gehören diese Geräte alle zur Multimediaausstattung von Haushalten.

Genauso stark ist die Streuung bei der Nutzungsdauer von dieser Geräteklasse zu erkennen. Der Mittelwert liegt bei etwa 30 Minuten, wenn man Zyklen über 6 Stunden ausblendet. Anderenfalls liegt dieser Mittelwert bei 2:30 Stunden und ist somit realistischer. In Abbildung 4.85 ist die Einschaltdauer aller Zyklen dargestellt. Betrachtet man den Bereich mit Nutzungsdauern unter 1 Stunde nochmals im Detail (Abbildung 4.86), so erkennt man, dass ein Großteil der Geräte Betriebsdauern von 0-10 Minuten haben.

#### 4. Auswertungen

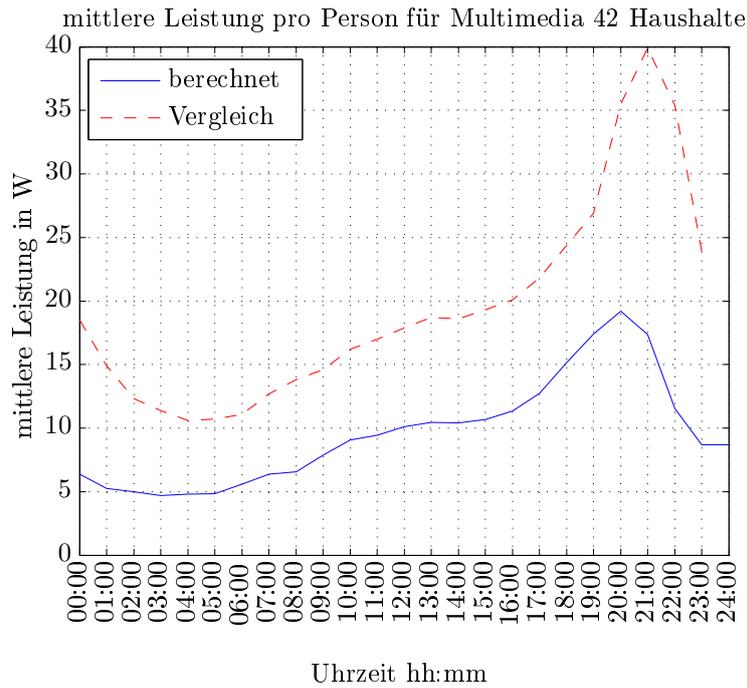


Abbildung 4.80.: mittlere Leistungsaufnahme für Multimedia-Geräte mit Vergleichsverlauf nach [9]

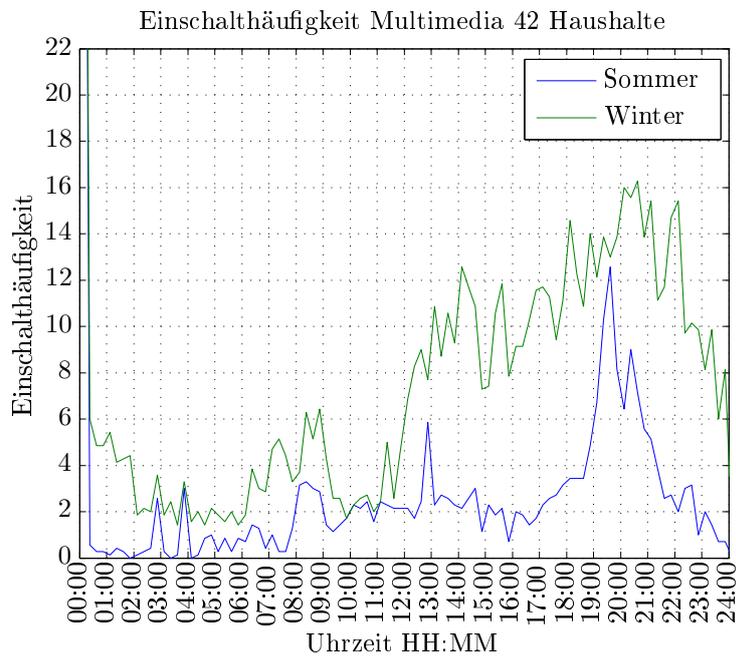


Abbildung 4.81.: Vergleich der Einschalthäufigkeit für Multimedia-Geräte Sommer vs. Winter

#### 4. Auswertungen

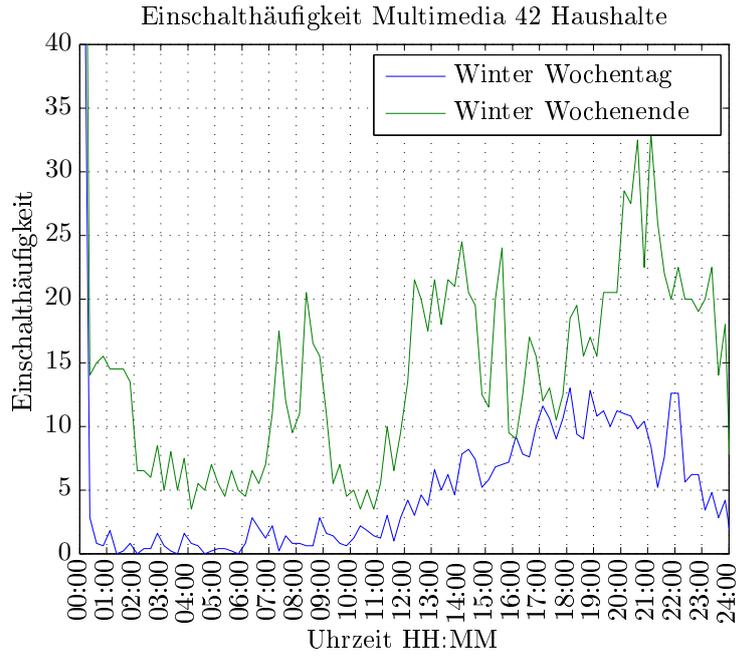


Abbildung 4.82.: Vergleich der Einschalthäufigkeit für Multimedia-Geräte Winter Werktag vs. Wochenende

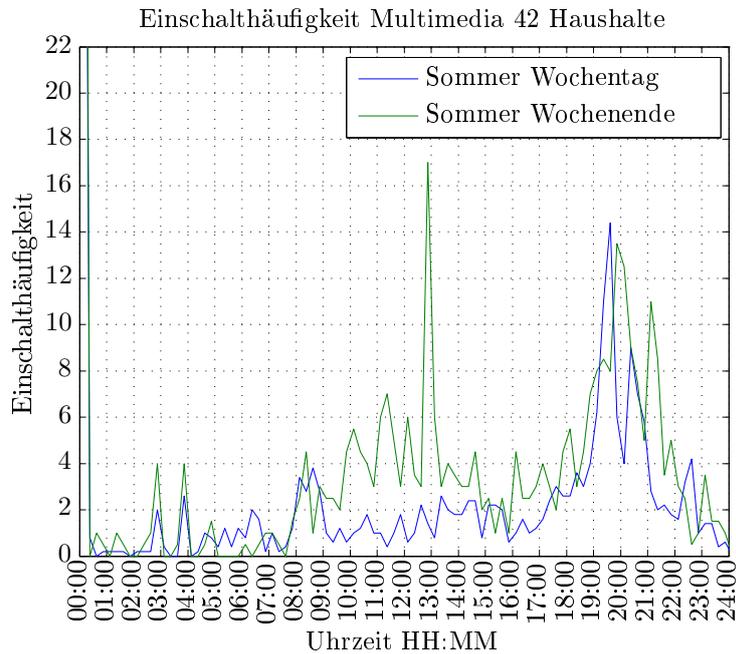


Abbildung 4.83.: Vergleich der Einschalthäufigkeit für Multimedia-Geräte Sommer Werktag vs. Wochenende

#### 4. Auswertungen

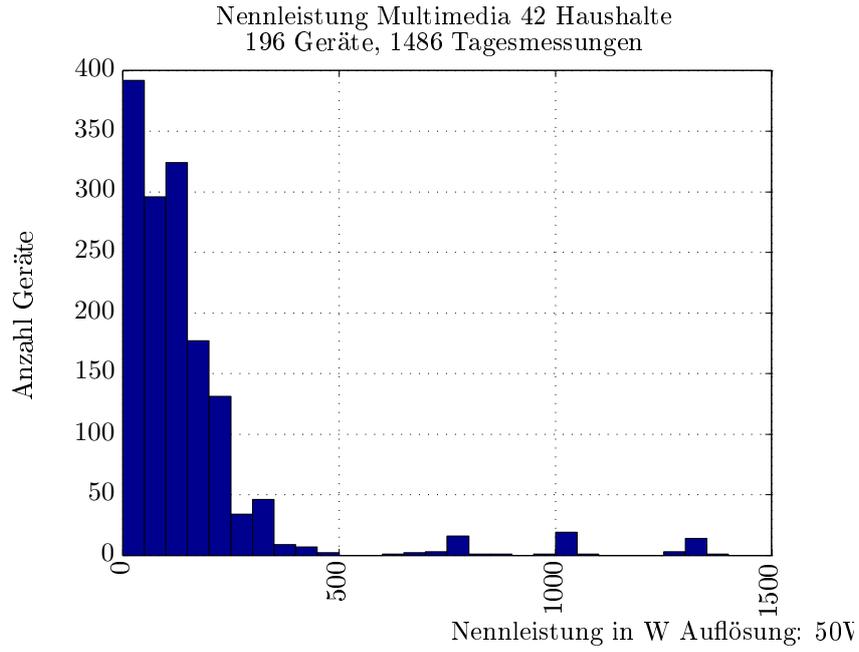


Abbildung 4.84.: Vergleich der Nennleistung für Multimedia-Geräte

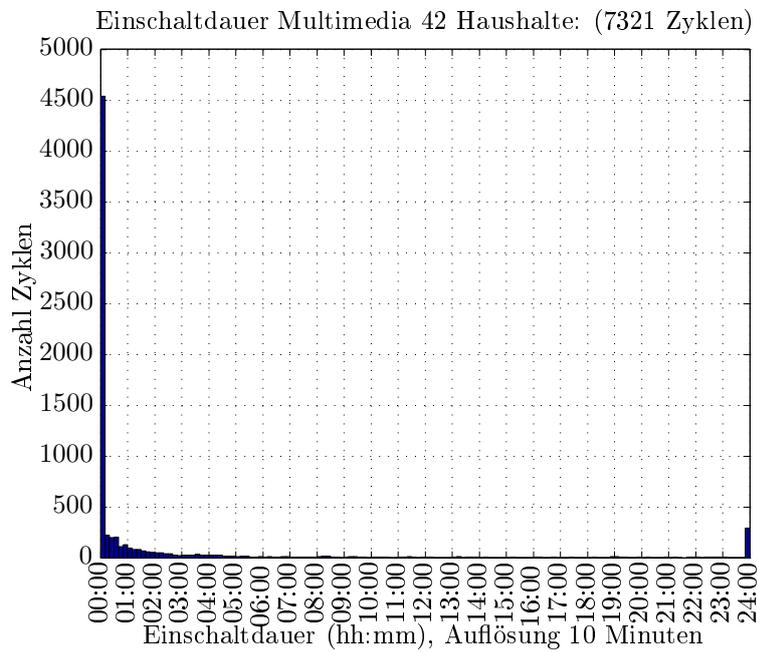


Abbildung 4.85.: Vergleich der Einschaltdauer für Multimedia-Geräte

## 4. Auswertungen

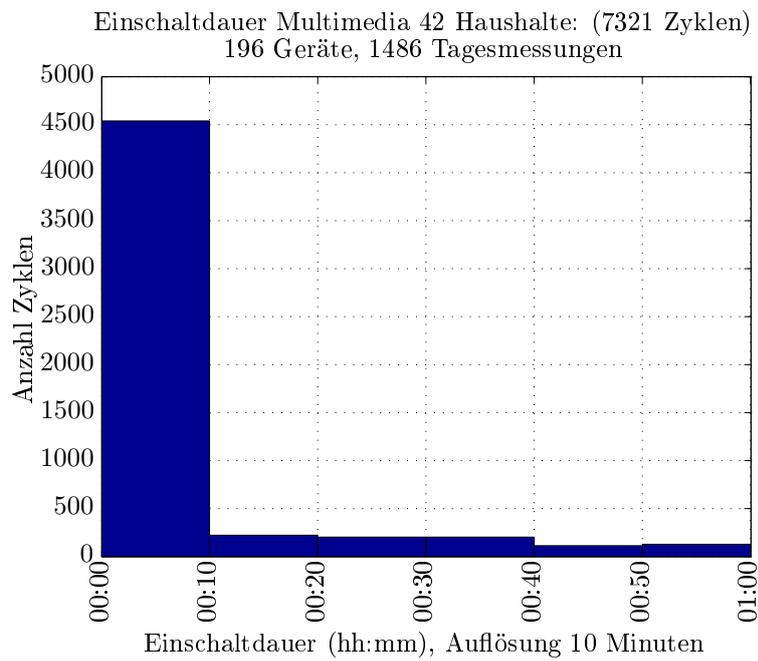


Abbildung 4.86.: Vergleich der Einschaltdauer für Multimedia-Geräte - Ausschnitt bis 1 Stunde

### 4.1.7. weitere vermessene Geräte

Zwei typische Geräte, die in Haushalten öfters zum Einsatz kommen, wurden in dieser Messreihe auch noch in mehreren Haushalten detailliert vermessen. Hier soll kurz auf die Auswertung der Messungen für Föhn und Bügeleisen eingegangen werden. Beim Bügeleisen beispielsweise erwartet man sich verstärkte Nutzung in den Abendstunden und am Wochenende. In Abbildung 4.87 ist dieser vermehrte Einsatz abends deutlich zu erkennen. Man sieht auch wieder sehr viele Zyklen, die um Mitternacht beginnen und auf fehlerhaft erkannte Daten zurückzuführen sind. Deutlich zeigt sich ein weiteres Phänomen in Abbildung 4.89, in der man erkennt, dass die Nennleistung von Bügeleisen einer extrem großen Streuung unterworfen ist. Die Maximalleistung liegt bei 2.7 kW, die minimale Leistung bei weniger als 10 W. Auch der Durchschnittswert ist deutlich niedriger als der Maximalwert und beträgt nur 1.2 kW, was für Bügeleisen zu niedrig erscheint. Daraus lässt sich schließen, dass an dieser Steckdose, die als Bügelsteckdose gekennzeichnet war, auch andere Geräte verwendet wurden, die eine deutlich abweichende Leistungsaufnahme hatten. Die typische Nutzungsdauer von Bügeleisen ergibt sich mit rund 30 Minuten und entspricht somit auch der Erwartung, erkennbar in Abbildung 4.88. Auch hier sieht man nochmals, dass auch Geräte mit deutlich längerer Betriebszeit zum Einsatz kamen. In den folgenden Abbildungen sind einige charakteristische Verläufe für Bügeleisen dargestellt. In der ersten Abbildung 4.90 sieht man den typischen Verlauf für ein temperaturgesteuertes Gerät. Je nach Wärmebedarf erfolgt eine Aufheizphase oder eine Pause. Interessant ist hierbei, dass die Blindleistungsaufnahme während des Heizens kapazitiv ist, obwohl man sich ohmsches Verhalten erwarten würde. In Abbildung 4.91 sieht man ein deutlich anderes Verhalten. Wenn man weiterhin davon ausgeht, dass es sich hierbei um ein Bügeleisen handelt, so würde man auf eine Dampfbügelstation schließen, die zwei unabhängige Heizelemente und somit zwei unabhängige und auch verschieden hohe Leistungsaufnahmen aufweist. Es wird einerseits das Wasser auf Dampftemperatur gehalten und andererseits die Bügelfläche je nach Bedarf aufgeheizt. Vergleicht man das Nutzungsverhalten von Werktagen und Wochenende im Sommer (4.92) und Winter (4.93) miteinander, so erkennt man, dass deutlich öfter am Wochenende gebügelt wird als an Werktagen. Der Verlauf der mittleren Tagesleistung ist in Abbildung 4.94 dargestellt. Man erkennt hier wieder den erwarteten Verlauf mit deutlichen Maxima in den Abendstunden und am frühen Nachmittag.

#### 4. Auswertungen

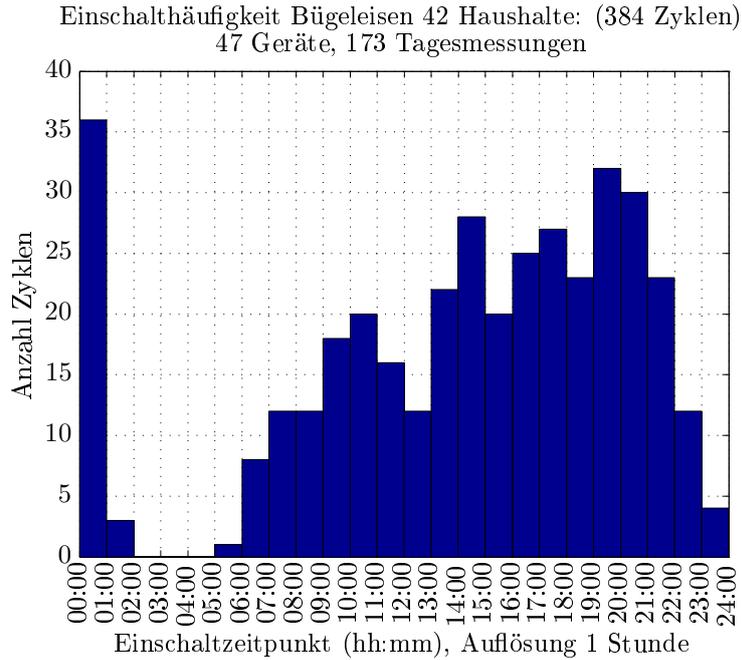


Abbildung 4.87.: Einschaltzeitpunkte für die untersuchten Bügeleisen

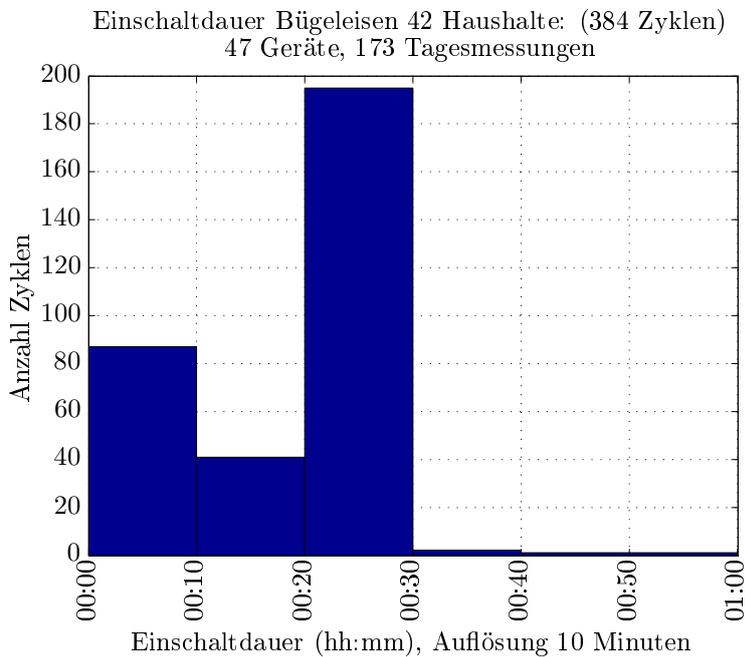


Abbildung 4.88.: Einsatzdauer für die untersuchten Bügeleisen

## 4. Auswertungen

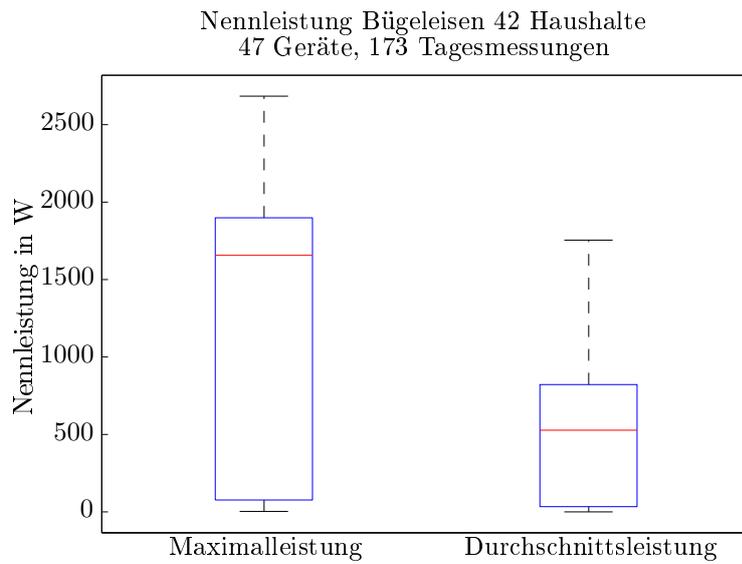


Abbildung 4.89.: Nennleistungen für die untersuchten Bügeleisen

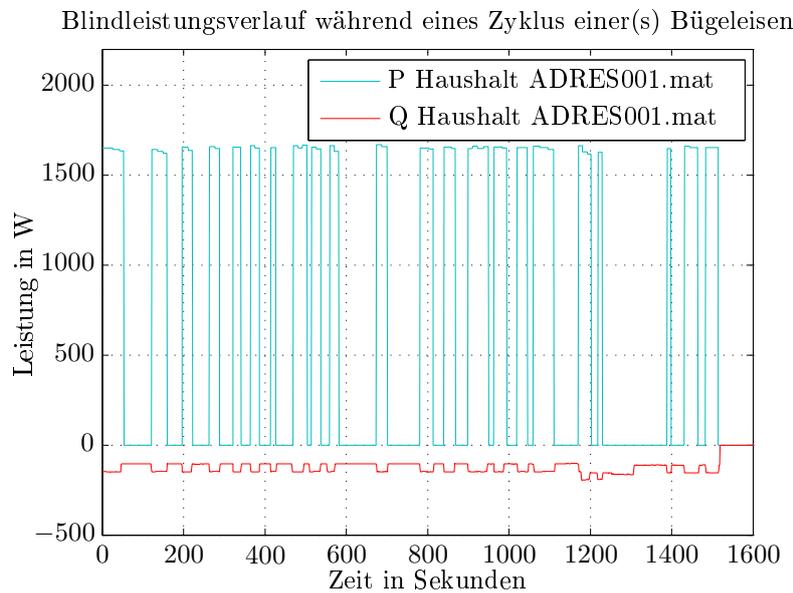


Abbildung 4.90.: Leistungsverlauf 1 für die untersuchten Bügeleisen

#### 4. Auswertungen

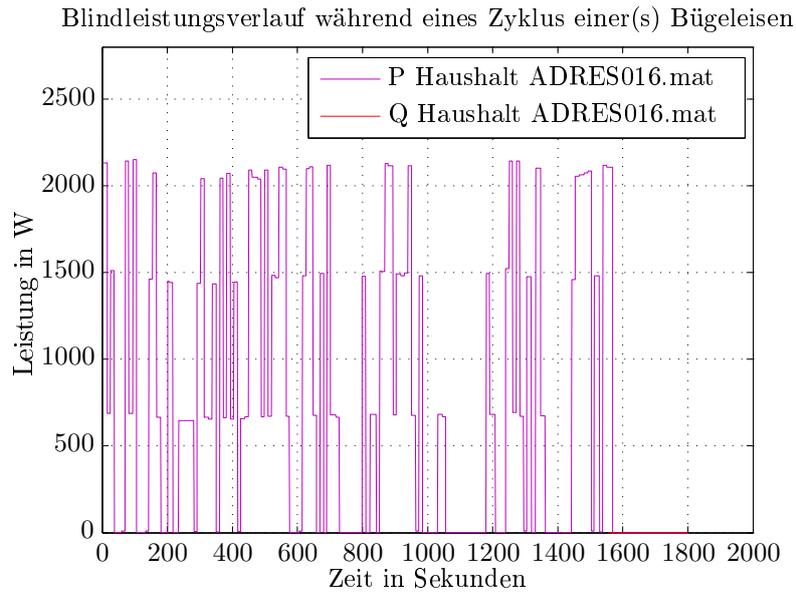


Abbildung 4.91.: Leistungsverlauf 2 für die untersuchten Bügeleisen

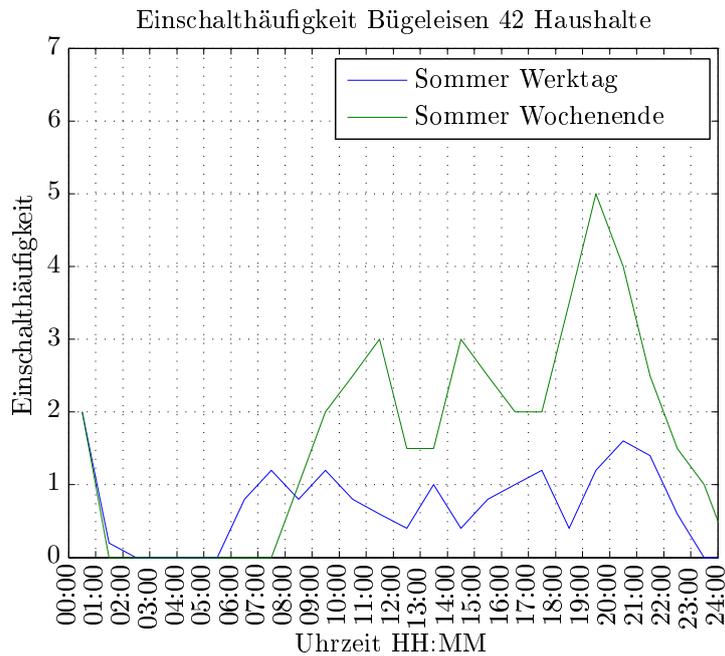


Abbildung 4.92.: Vergleich des Nutzungsverhaltens im Sommer der untersuchten Bügeleisen

#### 4. Auswertungen

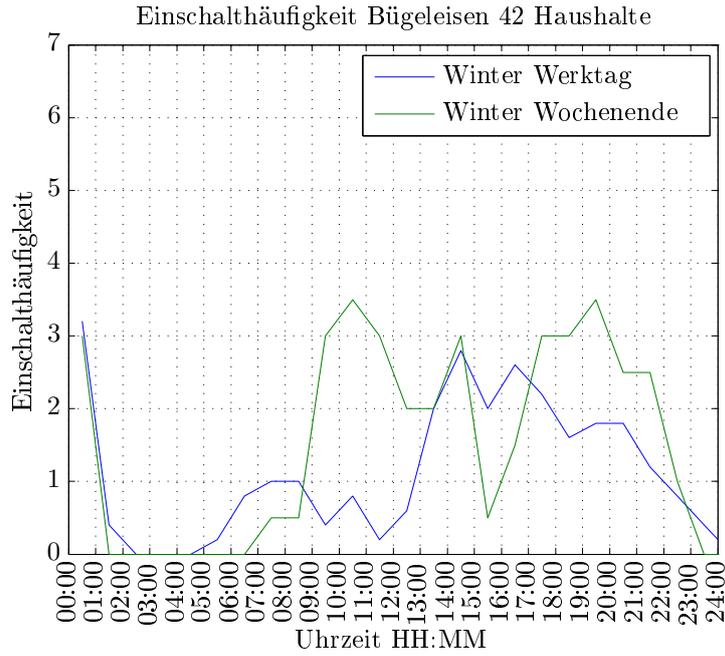


Abbildung 4.93.: Vergleich des Nutzungsverhaltens im Winter für die untersuchten Bügeleisen

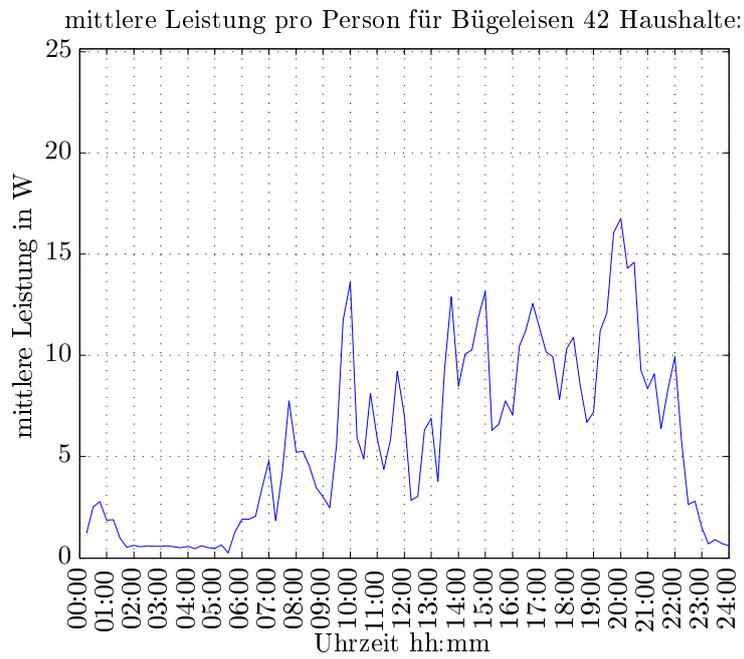


Abbildung 4.94.: mittlere Tagesleistung für die untersuchten Bügeleisen

#### 4. Auswertungen

Für einen Föhn ergeben sich ebenfalls einige interessante Erkenntnisse. Man erwartet sich hier verstärkten Einsatz in den Morgen-, und Abendstunden, wie auch ganz klar in Abbildung 4.95 erkennbar. Auch für die Einsatzdauer erwartet man sich einen Wert von rund 5 Minuten. Rechnerisch ergibt sich hier eine mittlere Verwendungsdauer von 4:30 Minuten. In Abbildung 4.96 ist die Verteilung sehr deutlich erkennbar. Es gibt nur sehr wenige Zyklen, die länger als 5 Minuten dauern. Die Nennleistung erwartet man sich im Bereich von 1-2 kW. In Abbildung 4.97 erkennt man eine doch recht große Streuung der Leistung, was wahrscheinlich darauf zurückzuführen ist, dass an dieser Steckdose auch andere Geräte wie Rasierer mit deutlich geringerer Leistung zum Einsatz gekommen sind. Bei den beiden Leistungsverläufen in den Abbildungen 4.98 und 4.99 sieht man sehr gut, dass während des Betriebes die Leistungsstufe des Föhns umgeschaltet wurde und durch die Seriendiode nur etwa die halbe Leistung aufgenommen wurde. In Abbildung 4.100 ist die mittlere Tagesleistung pro Person während eines Tages dargestellt. Auch hier sieht man wieder die verstärkte Nutzung in den Morgen-, und Abendstunden.

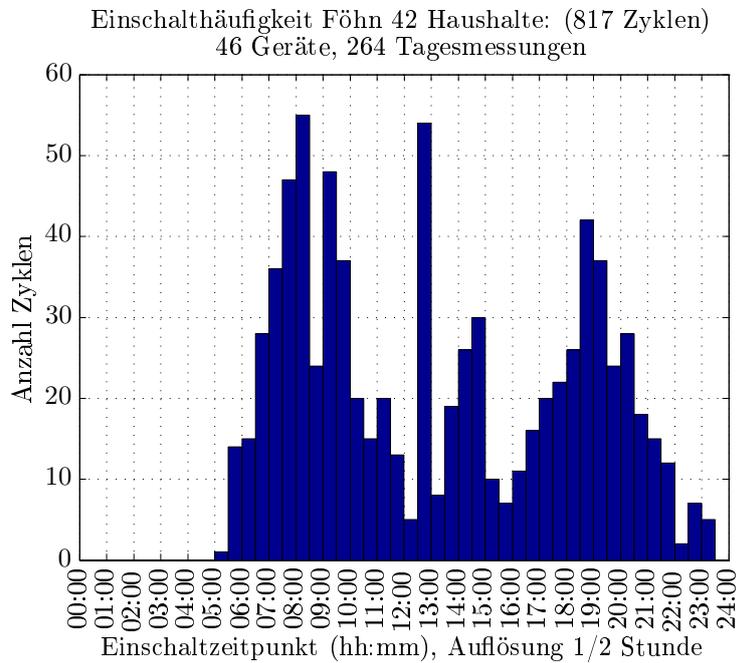


Abbildung 4.95.: Einsatzzeitpunkte für Föhns

#### 4. Auswertungen

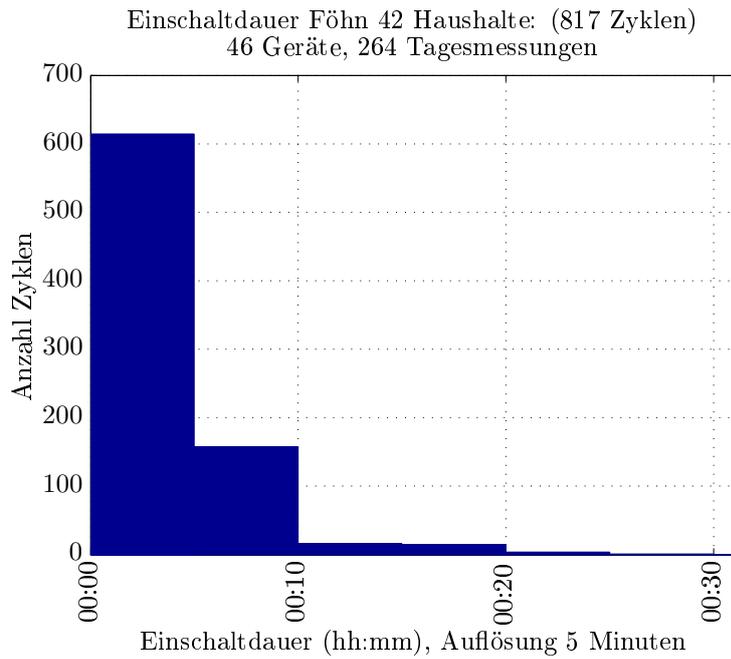


Abbildung 4.96.: Einschaltdauer für Föhns

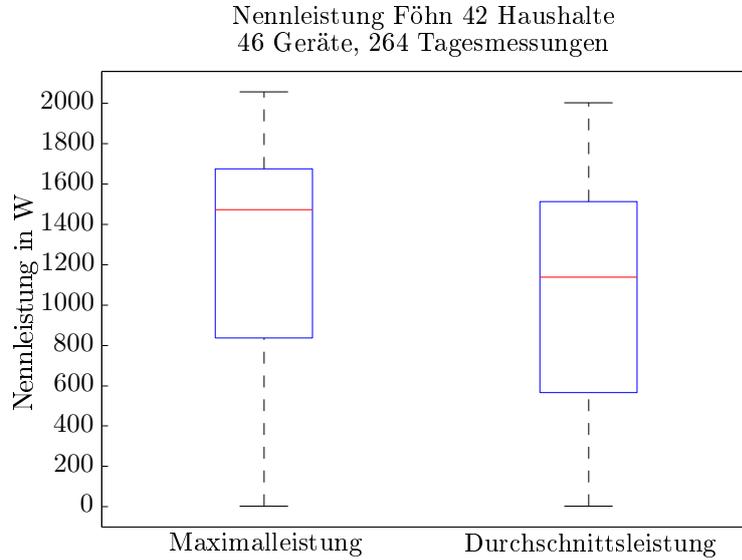


Abbildung 4.97.: Nennleistung für Föhns

#### 4. Auswertungen

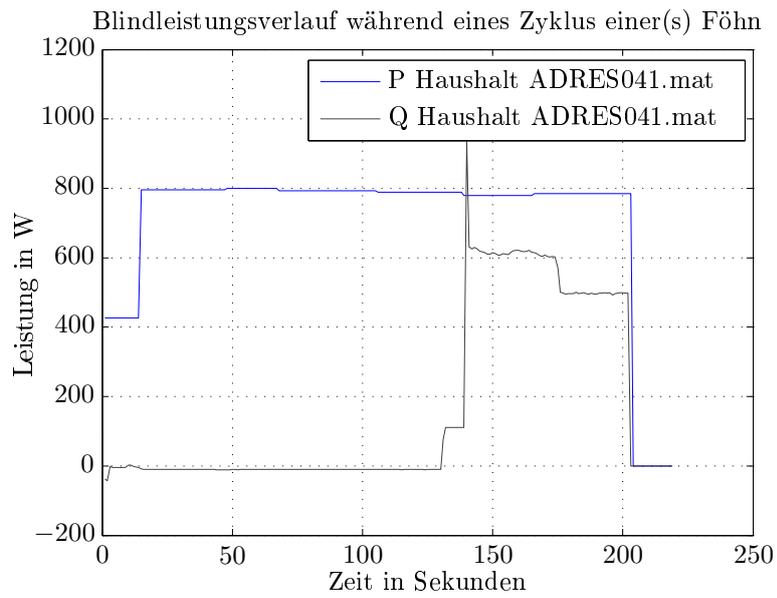


Abbildung 4.98.: Leistungsverlauf für Föhns

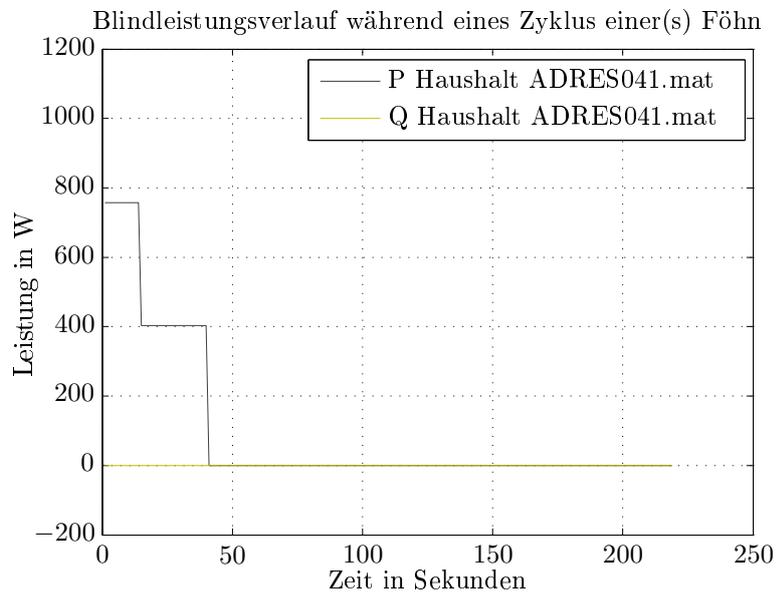


Abbildung 4.99.: Leistungsverlauf 2 für Föhns

## 4. Auswertungen

mittlere Leistung pro Person für Föhn 42 Haushalte: (265 Zyklen)

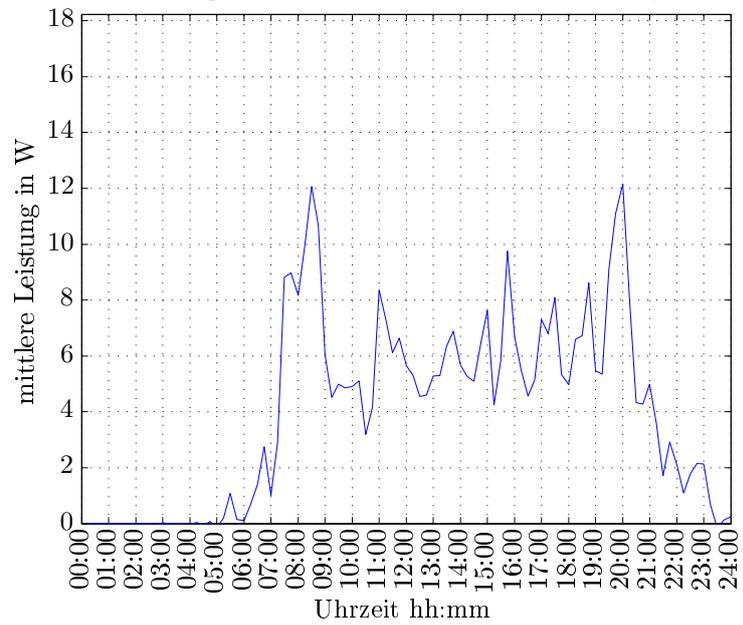


Abbildung 4.100.: mittlere Tagesleistung für Föhns

## 4. Auswertungen

### 4.1.8. Sonstige Geräte

Es wurden weiters einige Geräte vermessen, die nicht näher in die hier untersuchten Klassen passen oder in wenigen Haushalten vermessen wurden und daher keine eigene Klasse eingeführt wurde. Trotzdem soll diese Klasse kurz untersucht werden. Als erstes wurde der Einschaltzeitpunkt dieser Geräteklasse näher beleuchtet. Da keine Informationen über die tatsächlichen Geräte zur Verfügung stehen, lässt sich der in Abbildung 4.101 dargestellte Verlauf sehr schwierig begründen. Es zeigt sich, dass ein Großteil dieser Geräte tagsüber zum Einsatz kommen und daher wahrscheinlich Haushaltsgeräte wie Staubsauger oder Küchengeräte sind. Der Vergleich der Nennleistung zeigt, dass hier alle möglichen Geräte zum Einsatz kommen. Angefangen bei solchen mit Leistungen von wenigen Watt bis hin zu größeren Geräten mit fast 4kW Anschlussleistung. In den beiden Abbildungen 4.102 sowie 4.103 ist dieser Vergleich dargestellt.

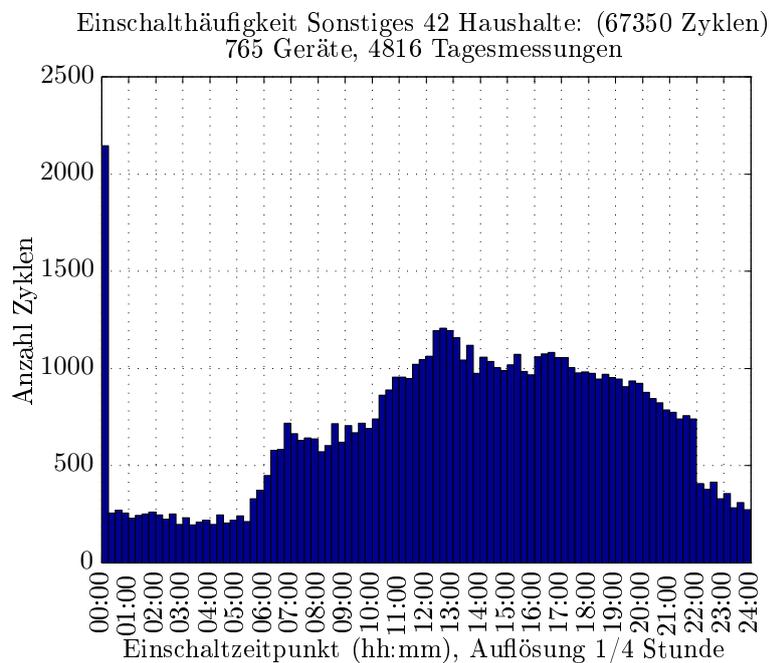


Abbildung 4.101.: Einschaltzeitpunkte für nicht näher unterteilte Geräte

#### 4. Auswertungen

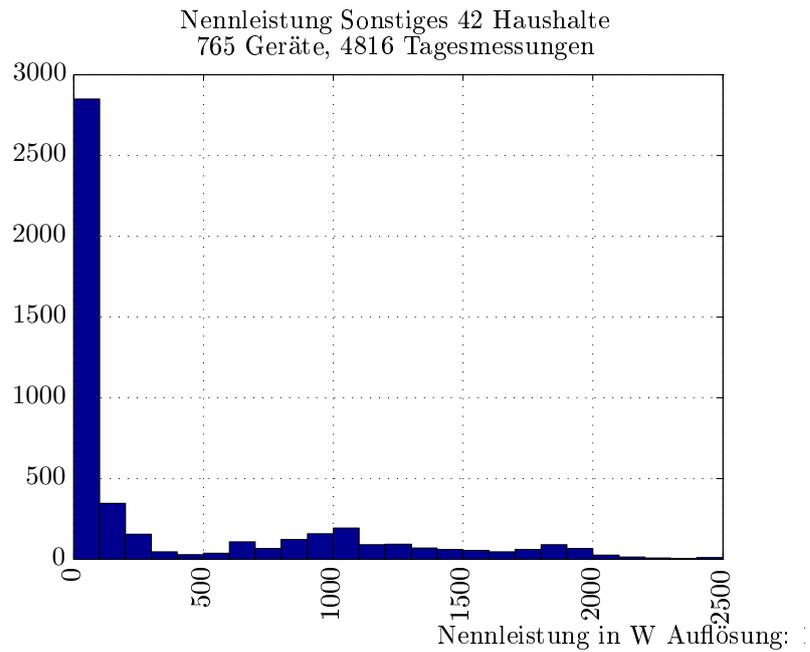


Abbildung 4.102.: Nennleistung für nicht näher unterteilte Geräte

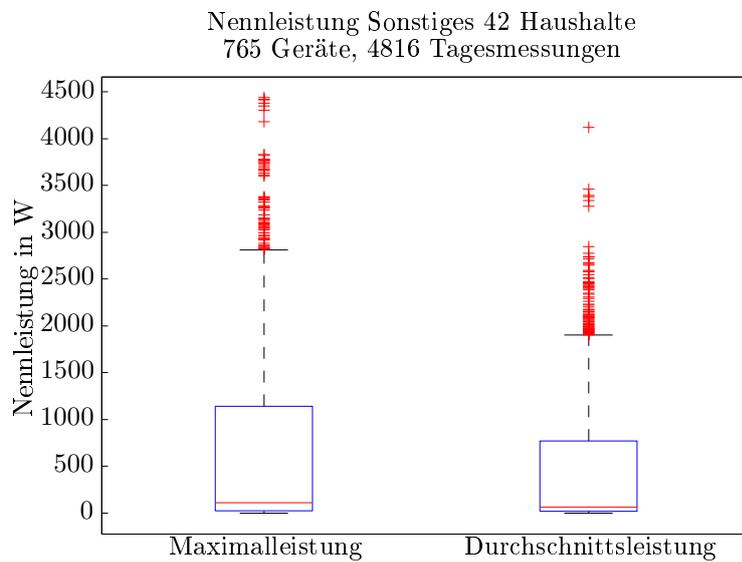


Abbildung 4.103.: Nennleistung für nicht näher unterteilte Geräte - Boxplot

## 4. Auswertungen

### 4.1.9. Zusammenfassung der wichtigsten Geräteparameter

In diesem Abschnitt soll eine kurze tabellarische Zusammenfassung der wichtigsten Geräteparameter der hier näher untersuchten Geräte aufgestellt werden. Dazu zählen neben der Anzahl an tatsächlich vermessenen Geräte (kurz Anzahl), der mittleren Anschlussleistung (kurz Leistung), der durchschnittlichen Nutzungsdauer (kurz Dauer) auch die durchschnittlichen Betriebszyklen pro Tag (kurz Zyklen/Tag).

Gerät	Anzahl	Leistung	Dauer	Zyklen/Tag
Kühlschrank	59	88.8	00:10:19	32.8
Gefrierschrank	77	97.6	00:16:17	26.8
Waschmaschine	71	2222.7	01:43:48	0.63
Wäschetrockner	30	1963.4	01:53:21	0.44
Geschirrspüler	15	2478.4	01:42:28	0.64
Wasserkocher	32	1917.4	00:01:52	3.1
Toaster	17	1178.7	00:14:14	0.86
Mikrowelle	39	1265.2	00:01:43	3.74
Multimedia	196	160.8	00:30:00	2.9
Bügeleisen	47	1244.5	00:31:15	0.77
Föhn	46	1256.3	00:04:30	1.67
Sonstiges	765	561.9	00:40:00	5.7

Tabelle 4.2.: Zusammenfassung der wichtigsten Geräteparameter

### 4.2. Haushaltslastprofile

In diesem Abschnitt beschäftigen wir uns mit der gesamten Leistungsaufnahme der Haushalte. Diese Messung wurde ebenfalls mit einer Auflösung von 1 Sekunde durchgeführt und daher stehen hier erstmals sehr genaue Zeitverläufe bereit, die sich doch vom normalerweise verwendeten dynamischen Haushaltslastprofil H0 unterscheiden.

Als Vergleichsprofil wird hier das der BDEW veröffentlichte Lastprofil Type H0 (beziehbar z.B. unter [8]) verwendet. Auf Grund des unterschiedlichen Verlaufs des Profils wird beim Vergleich auch der jeweilige Tag aus dem Vergleichsprofil herangezogen. Falls mehrere Haushalte oder mehrere Tage gewählt werden, wird sowohl ein mittleres Lastprofil aus den Messwerten als auch aus den Werten des Vergleichsprofils berechnet.

In Abbildung 4.104 sieht man zum Beispiel den Leistungsverlauf eines einzelnen Haushaltes an einem Sommertag und im Vergleich dazu das Haushaltslastprofil H0. Eine Ähnlichkeit zwischen den beiden Verläufen ist kaum erkennbar. So gibt es deutliche Spitzen im Lastgang, die im Vergleichsprofil nicht auftreten. Klarerweise kann ein statistisches Modell, wie das zu Grunde liegende Modell beim H0-Profil für einen einzelnen Haushalt nur eine Näherung angeben und keinen echten Verlauf nachbilden. Man erkennt hier aber, dass für leistungsautarke Siedlungen ohne Netzanbindung genauere Messungen als Bedarfsermittlung notwendig sind und eine Deckung des H0-Profiles nicht ausreichend sein wird. Die nächsten Bilder zeigen jeweils den Leistungsverlauf für 5, 10, 15, 20, 30 sowie allen 42 Haushalten an Werktagen.

Hier wird ersichtlich, dass sich die Mittlung über mehrere Haushalte besser an das H0-Profil anpasst. Je mehr Haushalte in dieser Mittelung herangezogen werden, desto besser passt sich der Verlauf an das H0-Profil an und bekräftigt somit die weitreichende Verwendung dieses Profils für die Bedarfsermittlung von größeren Gruppen von Haushalten. Wenn man von einem klassischen Ortsnetztransformator beispielsweise ausgeht und als Belastung eine kleine Siedlung mit etwa 40 Haushalten annimmt, kann man in erster Näherung eine Dimensionierung mit dem H0-Profil durchführen. Dass die Mittelung bereits ab einigen Haushalten bzw bei wenigen Haushalten auch ab einigen Tagen bereits recht brauchbare Ergebnisse liefert, lässt sich belegen, wenn man wie in Abbildung 4.111 dargestellt, die relative Abweichung vom Vergleichsprofil für verschiedene Konstellationen von Haushalten und Tagen ermittelt.

Weiters zeigt sich, dass eine deutlich ausgeprägte Morgenspitze auftritt, die im Standard-Lastprofil nicht wiederzufinden ist. Diese Spitze ist auf den vermehrten Einsatz von Kaffeemaschinen und Wasserkochern bei der Zubereitung des Frühstücks zurückzuführen.

Die Mittagsspitze tritt im errechneten mittleren Profil aller Haushalte um 12:00 Uhr auf, im Vergleichsprofil allerdings erst um 13:00. Ebenso ist die Abendspitze zeitlich versetzt.

Dieser zeitliche Versatz von etwa 1 Stunde des H0-Profiles gegenüber den hier ermit-

#### 4. Auswertungen

telten Leistungsverläufen lässt sich entweder auf fehlerhaft eingestellte Sommer-, bzw Winterzeit beim Messsystem oder auf Nichtbeachten der Zeitumstellung im hier verwendeten Vergleichsprofil einerseits oder aber auf eine tatsächliche Verschiebung des Leistungsverbrauch gegenüber der angenommenen Vergleichswerte zurückführen.

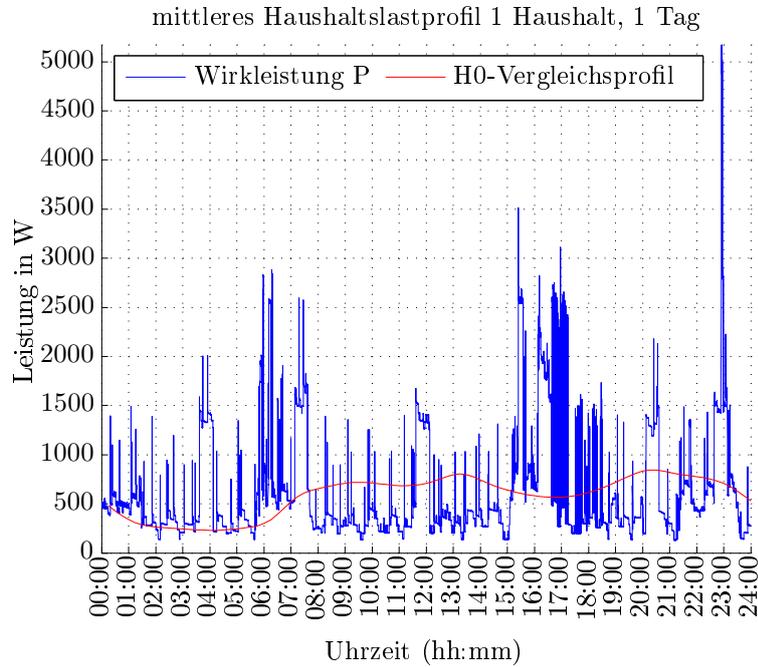


Abbildung 4.104.: Lastprofil von einem Haushalt und H0-Vergleichsprofil

#### 4. Auswertungen

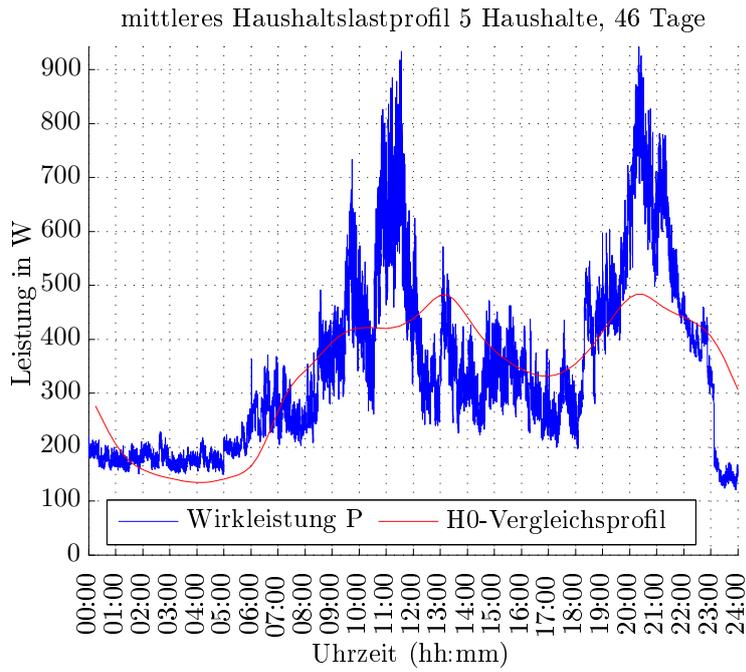


Abbildung 4.105.: Lastprofil von 5 zufällig gewählten Haushalten und H0-Vergleichsprofil

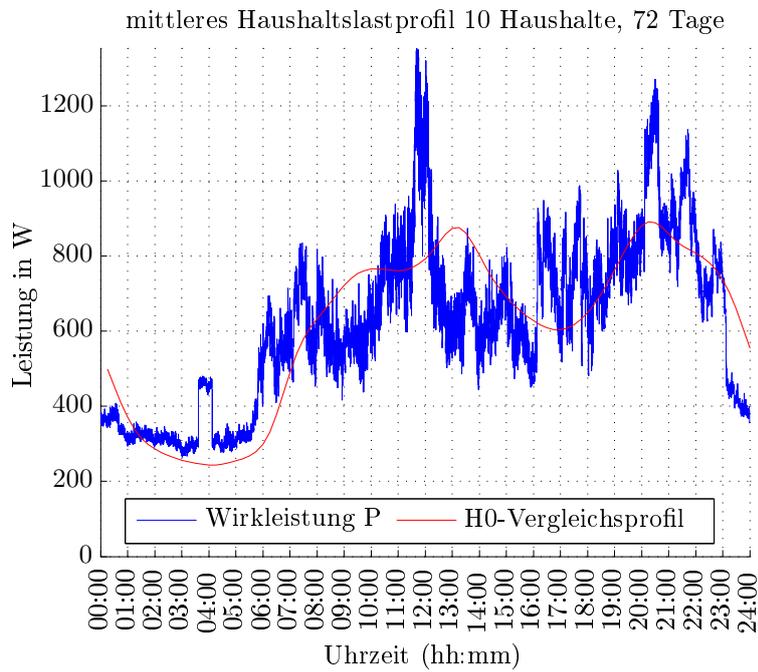


Abbildung 4.106.: Lastprofil von 10 zufällig gewählten Haushalten und H0-Vergleichsprofil

#### 4. Auswertungen

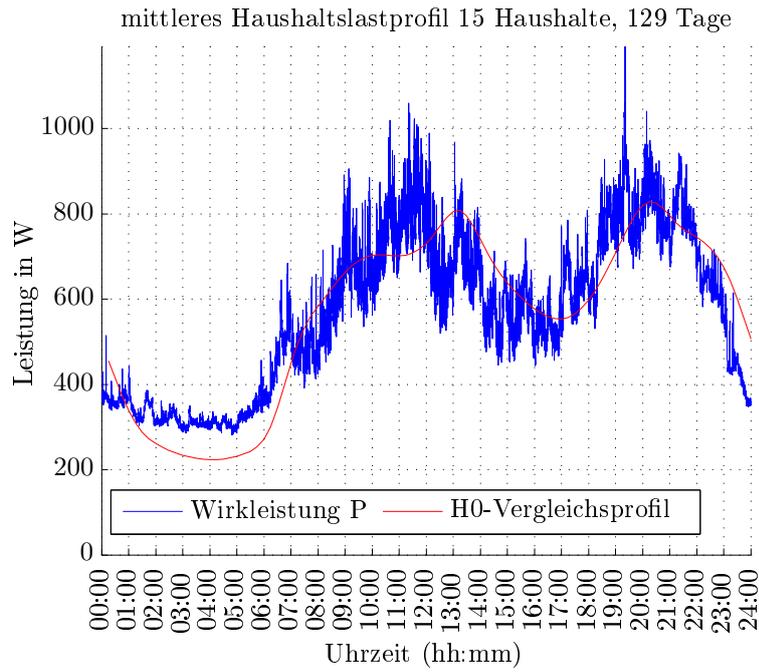


Abbildung 4.107.: Lastprofil von 15 zufällig gewählten Haushalten und H0-Vergleichsprofil

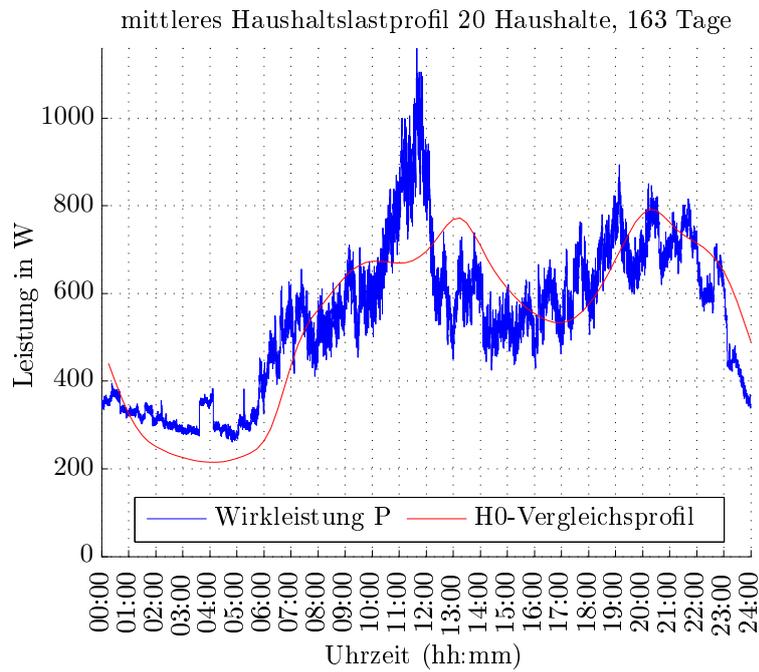


Abbildung 4.108.: Lastprofil von 20 zufällig gewählten Haushalten und H0-Vergleichsprofil

#### 4. Auswertungen

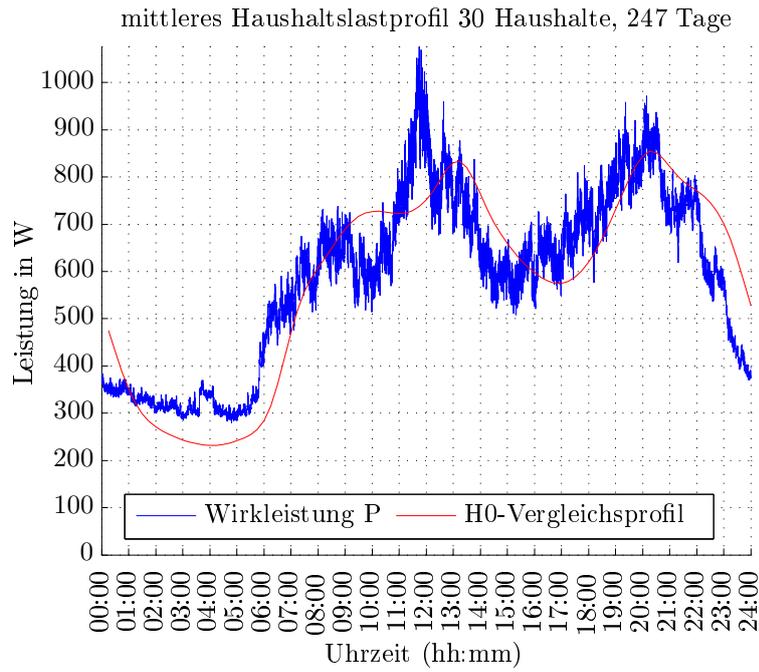


Abbildung 4.109.: Lastprofil von 30 zufällig gewählten Haushalten und H0-Vergleichsprofil

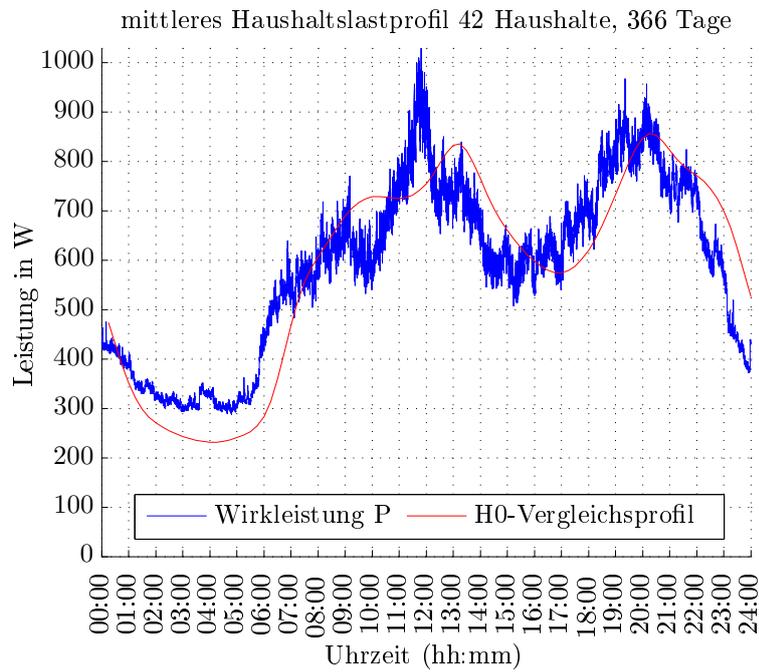


Abbildung 4.110.: Lastprofil von allen 42 Haushalten und H0-Vergleichsprofil

#### 4. Auswertungen

Abweichung des gemessenen Haushaltslastganges vom Vergleichsprofil

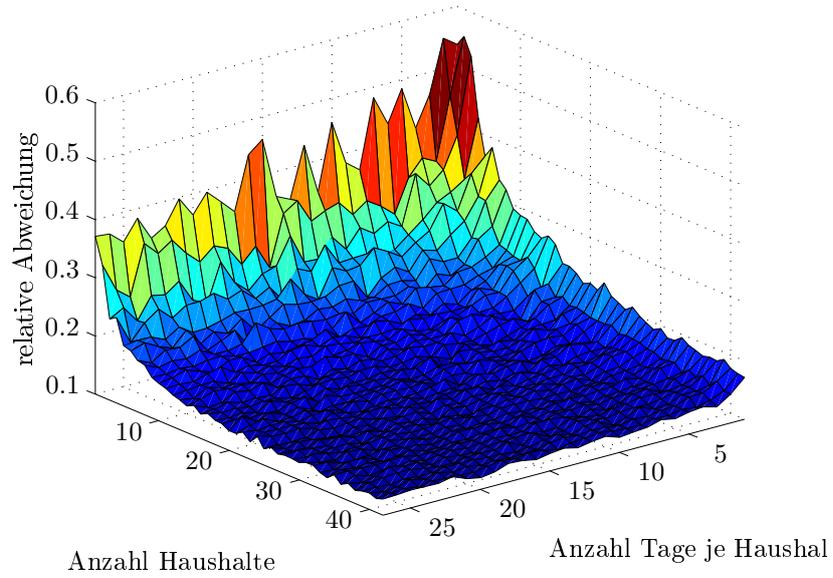


Abbildung 4.111.: Differenz zwischen H0-Vergleichsprofil und Haushaltslastgang für verschiedene Anzahl an Haushalten und Tagen

#### 4. Auswertungen

Von Interesse ist natürlich auch der Blindleistungsverlauf der Haushalte und ob hier eine Korrelation mit dem Wirkleistungsverlauf zu erkennen ist beziehungsweise wie der Verlauf des Leistungsfaktors für Haushalte aussieht. Zu diesem Punkt gibt es in der Literatur kaum Analysen und daher wird hierauf ein besonderes Augenmerk gelegt.

In Abbildung 4.112 ist der Blindleistungsverlauf eines Haushaltes gemeinsam mit der Wirkleistung und dem daraus berechneten Leistungsfaktor dargestellt. Man erkennt, dass in diesem Fall die Blindleistung eigentlich nur dann merkbar vorhanden ist, wenn in der Wirkleistung markante Schaltflanken auftreten. Diese Flanken gehen einher mit einer Verschlechterung des Leistungsfaktors, der einen Großteil der Zeit in einem sehr kleinen Bereich um den idealen Wert 1 schwankt.

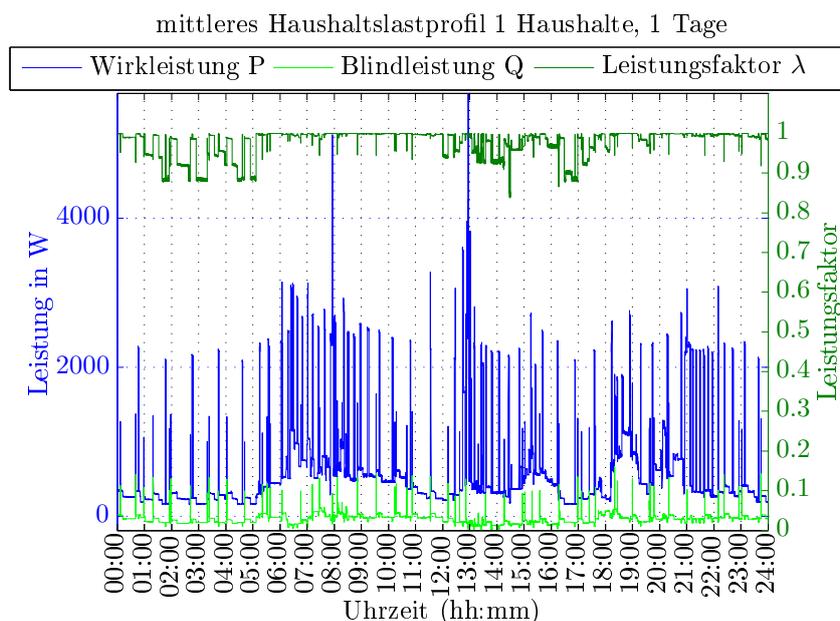


Abbildung 4.112.: Lastprofil von einem zufällig gewählten Haushalt

Ein sehr interessantes Phänomen ist in Abbildung 4.113 zu sehen. Hier gibt es einen Bereich von knapp 2 Stunden zwischen 13:00 und 15:00, in dem sowohl die Blind-, als auch die Wirkleistung extrem viele sehr hohe Flanken aufweisen. In diesem Zeitraum springt somit auch der Leistungsfaktor immer wieder zwischen den Werten 0.98 und 0.32 hin und her. Ein Ausschnitt aus diesem sehr interessanten Zeitverlauf ist in der Abbildung 4.114 zu sehen.

#### 4. Auswertungen

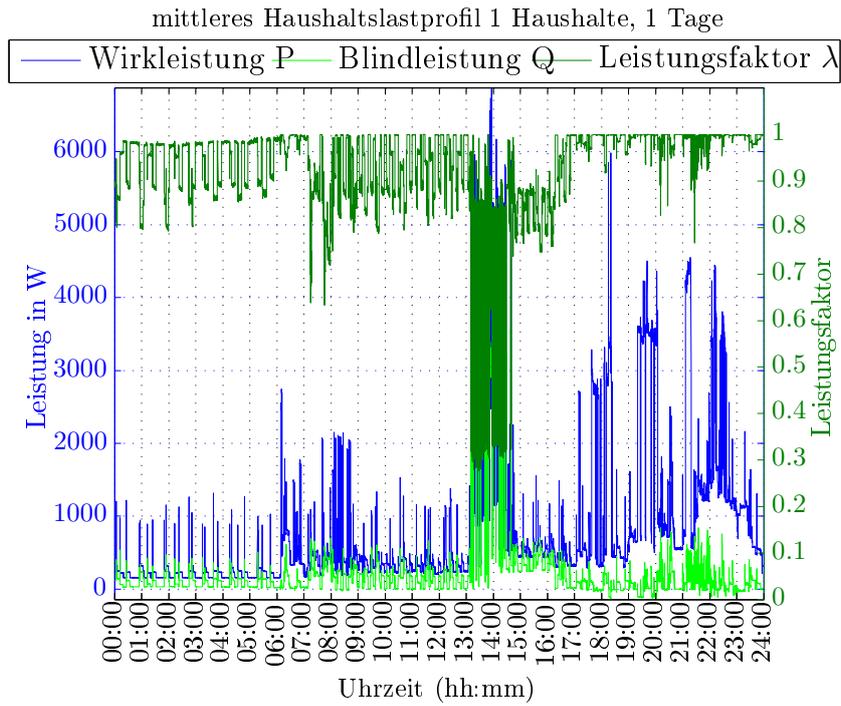


Abbildung 4.113.: Lastprofil von einem zufällig gewählten Haushalt

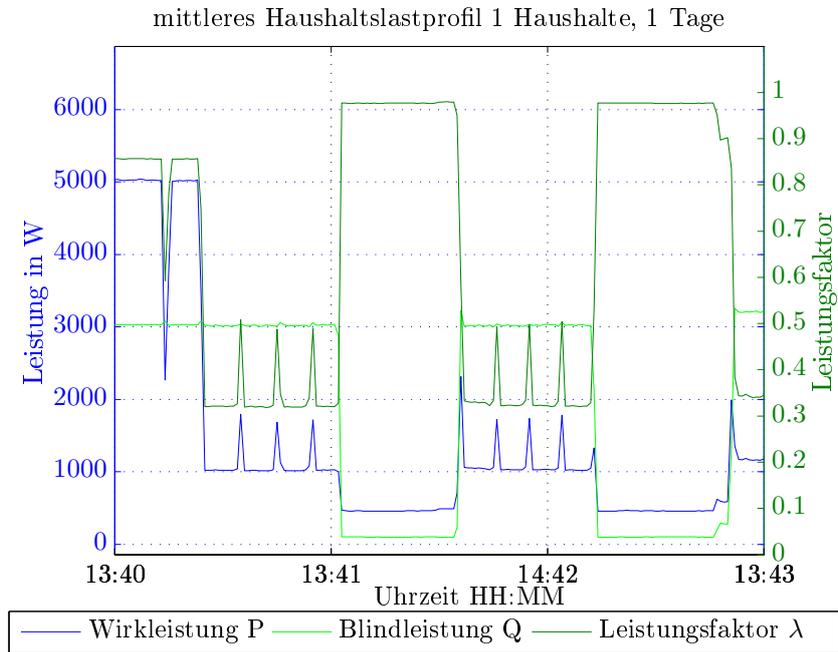


Abbildung 4.114.: Ausschnitt aus dem Lastprofil von einem zufällig gewählten Haushalt

#### 4. Auswertungen

Eine wieder andere Erscheinung kann man bei einem dritten gewählten Haushalt feststellen. In Abbildung 4.115 erkennt man ganz deutlich einen periodischen Verlauf von sowohl Wirk-, als auch Blindleistung und damit verbunden auch dem Leistungsfaktor. Die Periodendauer ist hier etwa 40 Minuten und der Verlauf der Leistungsaufnahme lässt auf ein Kühlgerät schließen, wie in Abbildung 4.116 erkennbar.

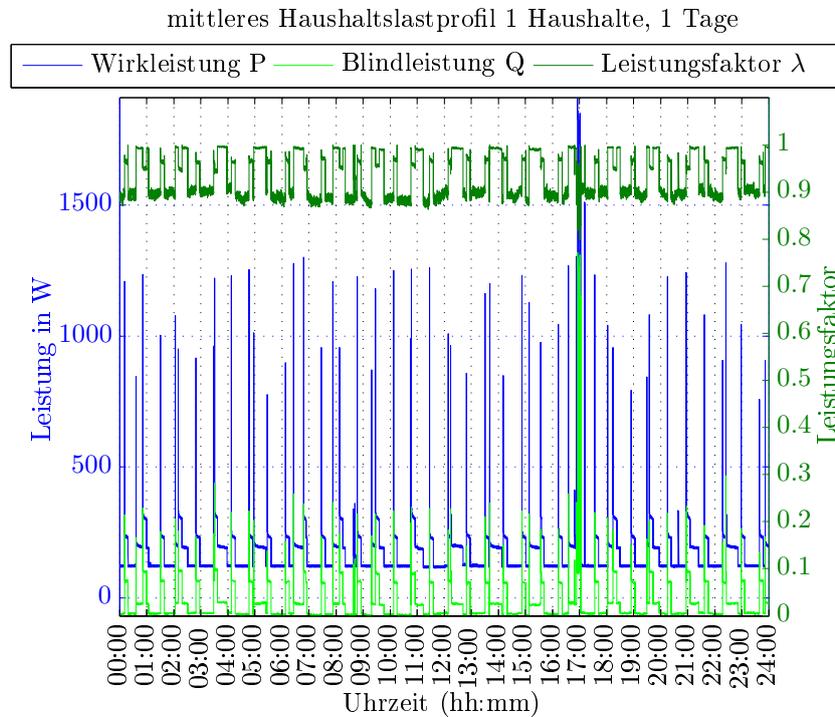


Abbildung 4.115.: Lastprofil von einem zufällig gewählten Haushalt

#### 4. Auswertungen

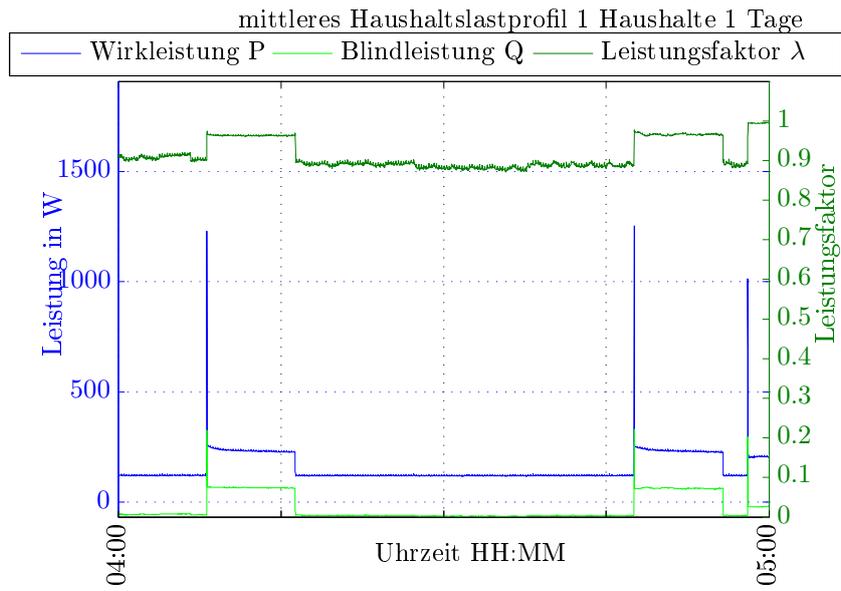


Abbildung 4.116.: Ausschnitt aus dem Lastprofil von einem zufällig gewählten Haushalt

#### 4. Auswertungen

In Abbildung 4.117 wird der mittlere Leistungsfaktor aller Haushalte zusammen mit dem Leistungsverlauf dargestellt. Man erkennt hierbei, dass der Verlauf des Leistungsfaktors über den Tag betrachtet durchaus als konstant angenommen werden kann.

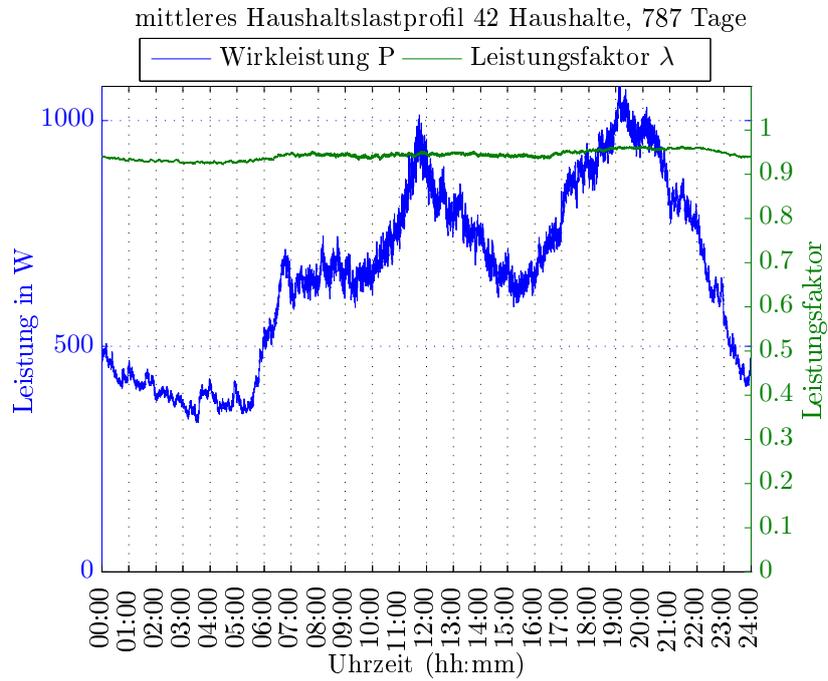


Abbildung 4.117.: Leistungsfaktor  $\lambda$  und Wirkleistung P von allen Haushalten

### 4.3. Restprofil

In diesem Abschnitt wird das hier als Restprofil bezeichnete Tagesprofil untersucht. Das Restprofil bezeichnet hierbei das Haushaltslastprofil abzüglich aller im Detail vermessenen Geräte. Dieses Restprofil ist besonders interessant, da die größeren Lasten wie E-Herd oder Elektroheizungen in den Haushalten nicht im Detail vermessen werden konnten, aber sich hier vielleicht interessante Erkenntnisse extrahieren lassen.

Bei der Generierung des Restprofils traten einige unerwartete Fehler auf, die im folgenden mit einer kurzen Beschreibung des Lösungsvorganges angeführt sind. Im zweiten Abschnitt wird dann auf einige interessante Details in den Restprofilen von 5 zufällig gewählten Haushalten eingegangen. Der dritte Abschnitt widmet sich den Anteilen der einzelnen Geräte sowie des Restprofils an der Summenmessung und soll somit einen Überblick geben, wie viel Prozent der Leistung in den einzelnen Haushalten im Detail vermessen worden sind.

#### 4.3.1. Ermittlung des Restprofils

Die Ermittlung des Restprofils stellte einige unerwartete Herausforderungen an das zu erstellende Programm.

Als erstes kleines Problem ergab sich die Tatsache, dass die Messreihen unterschiedlich lang sind, da das Messsystem für die gesamten Daten jeweils 5 Minuten vor Mitternacht mit der Messung begonnen hat und erst am nächsten Tag fünf Minuten nach Mitternacht die Tagesmessreihe wieder beendet hat. Dies führt zu 300 zusätzlichen Messpunkten am Anfang und Ende der Messreihe, die bei der Auswertung des Restprofils einfach abgeschnitten werden. Weiters sind einige der Messreihen nicht über einen ganzen Tag aufgenommen und bei dieser Betrachtung werden die fehlenden Messwerte mit 0 ergänzt.

Ein zweites größeres Problem wurde bei der Untersuchung des Gesamtverlaufs und eines einzelnen Geräts (in diesem Fall ein Gefrierschrank) entdeckt und ist in der Abbildung 4.118 sowie im Detail in Abbildung 4.119 dargestellt. Wie man erkennt, sind die Start-, sowie Endpunkte der Zyklen zeitlich versetzt. Dies würde nicht weiter schwierig zu lösen sein, wenn die gesamte Detailmessung um eine konstante Zeitdauer versetzt wäre. Dies ist hier leider nicht der Fall. Die Zeitverschiebung der Anfangspunkte schwankt in den dargestellten Lastgängen zwischen 0 und 2 Zeitpunkten, also nicht in einem großen Ausmaß. Trotzdem muss jeder Zyklus in seinem Startpunkt einzeln an das Summenprofil angepasst werden und dann subtrahiert werden. Ein größeres Problem stellt jedoch die Zeitverschiebung der Endzeitpunkte dar. Diese schwankt zwischen 0 und 5 Zeitpunkten und lässt daher nicht nur auf eine Zeitverschiebung der Messreihen zueinander sondern auch auf eine Zeitdehnung einer Messreihe gegenüber der anderen schließen, da die Zyklen in den beiden Darstellungen unterschiedlich lang sind. Zur

#### 4. Auswertungen

Problemlösung wird davon ausgegangen, dass die Summenmessung exakter ist als die Detailmessung und daher die Zeitreihe der Detailmessung zu kurz aufgezeichnet wurde. Die fehlenden Datenpunkte der Detailmessung werden jetzt so ergänzt, dass der gesamte Zyklus von der Summenmessung abgezogen werden kann.

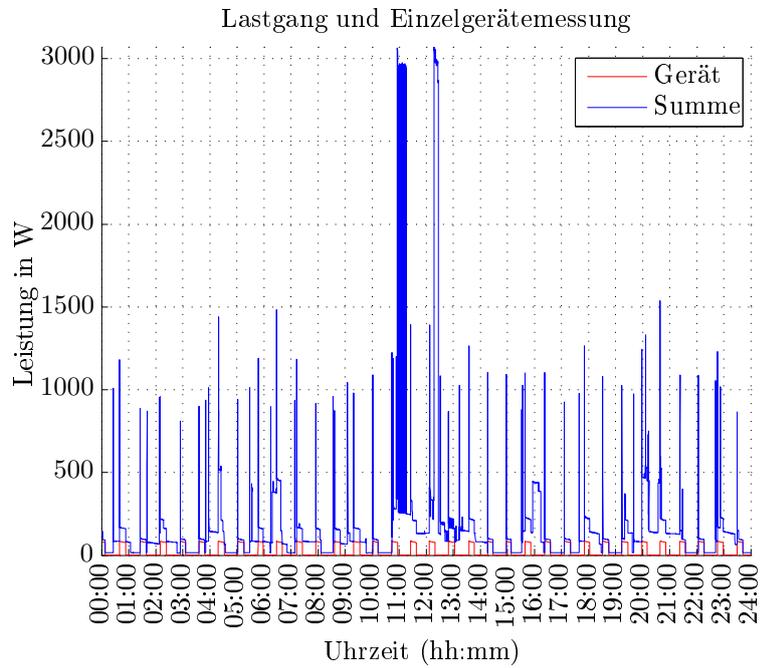


Abbildung 4.118.: Summenlastprofil und Einzelgerätemessung Gefrierschrank vor der Korrektur

## 4. Auswertungen

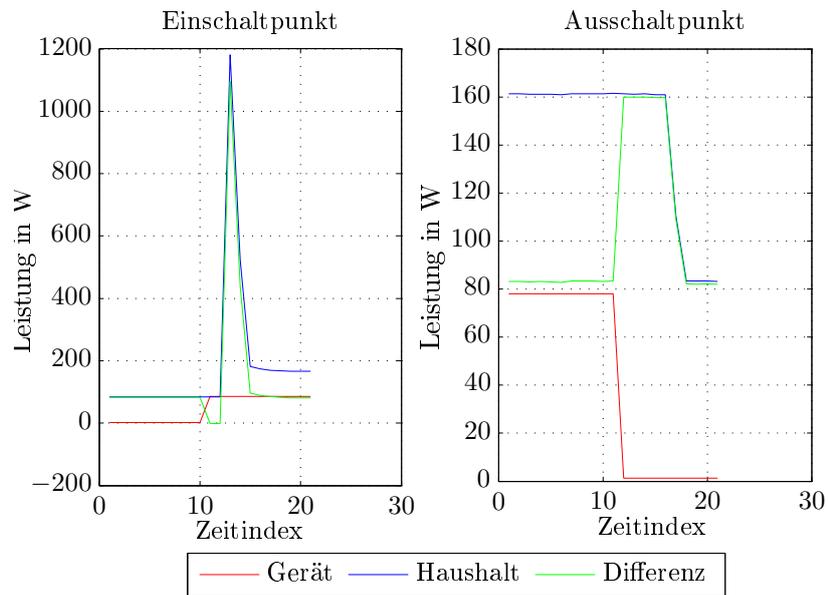


Abbildung 4.119.: Summenlastprofil und Einzelgerätemessung Gefrierschrank - Detailansicht eines Zyklus vor der Korrektur

#### 4. Auswertungen

Ein weiteres Phänomen ist die in diesem Fall zum Einschaltzeitpunkt auftretende Schaltflanke, die in der Detailmessung nicht ersichtlich ist. Ein Ignorieren dieser Flanke würde zu fehlerhafter Darstellung des Restprofils führen. Daher muss diese Flanke als zum eigentlichen Zyklus des Gefrierschranks gehörend gewertet werden und für das Restprofil von der Summenmessung mit einer geeigneten Funktion subtrahiert werden. Dies ist eigentlich nicht weiter verwunderlich, da Gefriergeräte einen Kompressor besitzen, der zum Einschaltzeitpunkt einen sehr hohen Anlaufstrom benötigt.

Das Ergebnis der Subtraktion des hier als Beispiel betrachteten Gefrierschranks für einen Beispielhaushalt ist in Abbildung 4.120 sowie wieder ein einzelner Zyklus im Detail in Abbildung 4.121 dargestellt. Man erkennt nun, dass in der korrigierten Differenzmessung weder die Einschaltspitze am Beginn des Zyklus noch die fehlenden Zeitpunkte des Zyklus in der korrigierten Differenz auftreten.

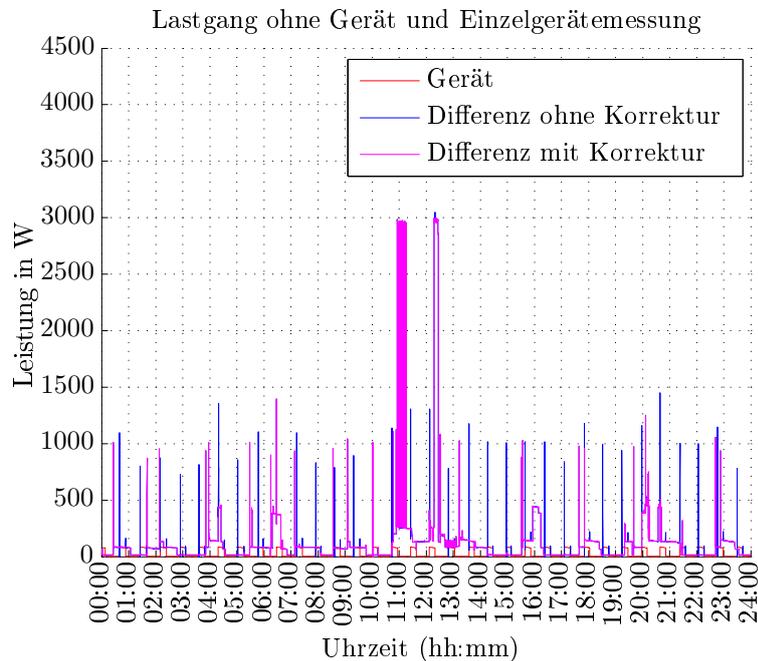


Abbildung 4.120.: Summenlastprofil und Einzelgerätemessung Gefrierschrank nach der Korrektur

#### 4. Auswertungen

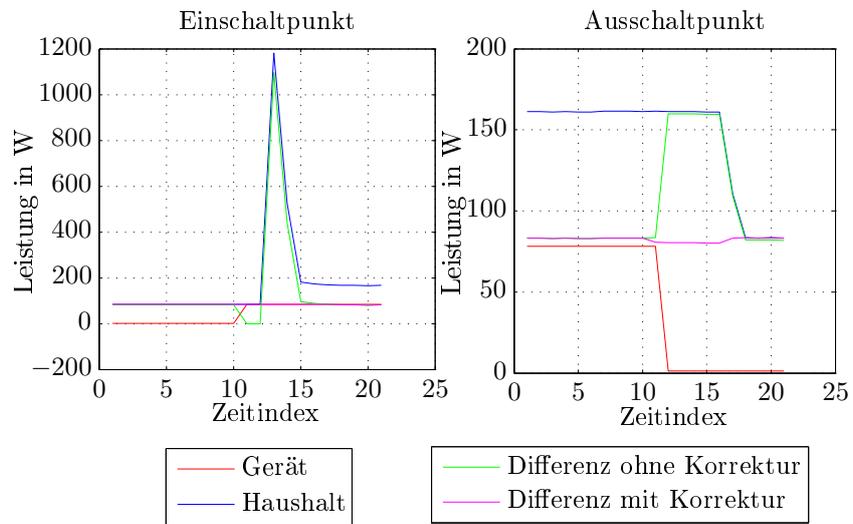


Abbildung 4.121.: Summenlastprofil und Einzelgerätemessung Gefrierschrank nach der Korrektur - Detailansicht eines Zyklus

#### 4. Auswertungen

Beim Durchlaufen des hier beschriebenen Algorithmus für andere Geräte kam es dann zu weiteren Problemen, die wieder in eigenen Routinen behoben werden mussten. So kam es beispielsweise vor, dass die Phasenzuordnung der Geräte falsch war, wie in Abbildung 4.122 sowie im Detail in Abbildung 4.123 leicht erkennbar ist. In diesem Fall muss eine händische Zuordnung zur richtigen Phase getroffen werden, da es für den Algorithmus sehr schwierig ist, diese Zuordnung selbst zu treffen. Für den als Beispiel betrachteten Fall kann eine eindeutige Zuordnung zu einer Phase nicht getroffen werden, wie aus den beiden Abbildungen 4.122 sowie 4.123 leicht ersichtlich ist. Die Ursache für dieses Phänomen ist leider ein großes Rätsel. Entweder ist die Detailmessung so stark zeitversetzt, dass eine optische Zuordnung zu einer Phase nicht möglich ist, oder die Detailmessung ist komplett falsch und kann daher nicht zugeordnet werden. Auf jeden Fall wird bei dem hier betrachteten Haushalt dieser Kühlschrank nicht mehr berücksichtigt.

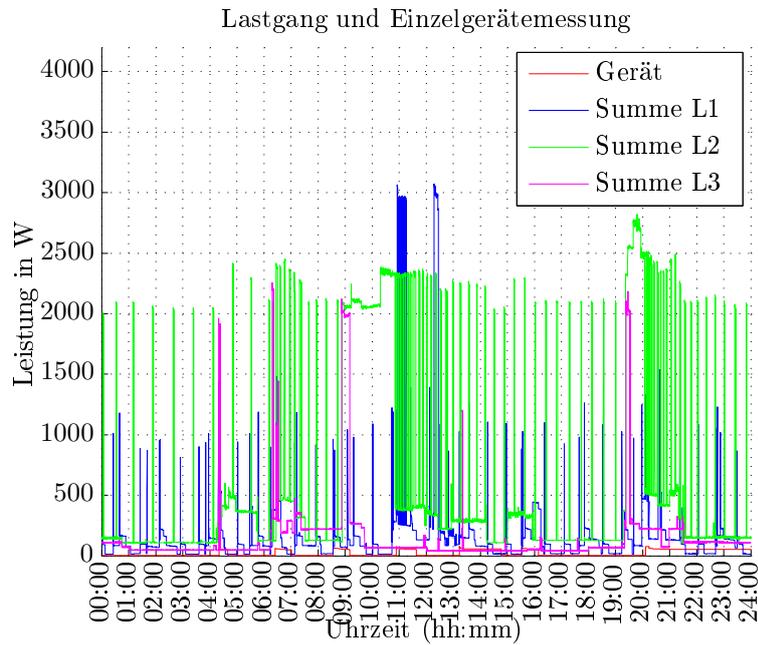


Abbildung 4.122.: Summenlastprofil 3 Phasen und Einzelgerätemessung Kühlschrank

## 4. Auswertungen

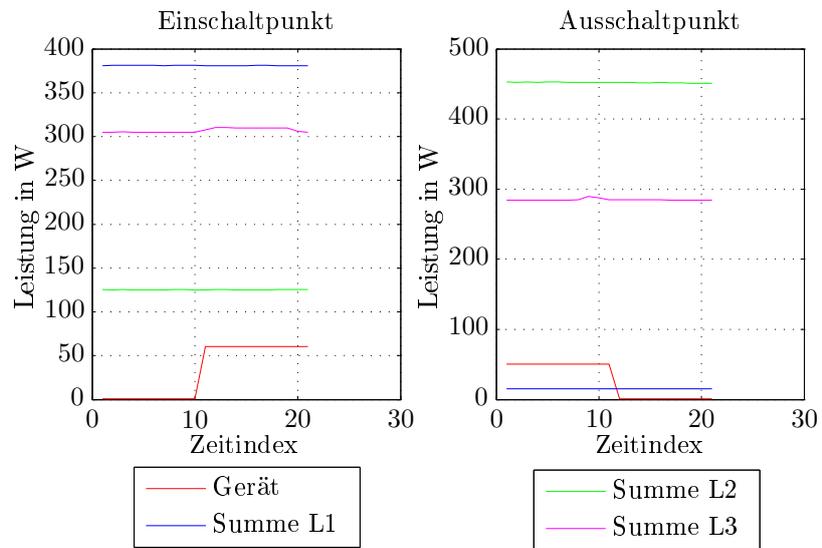


Abbildung 4.123.: Summenlastprofil 3 Phasen und Einzelgerätemessung Kühlschrank  
Detailansicht eines Zyklus

#### 4. Auswertungen

Mit diesen bis jetzt beschriebenen Algorithmen lassen sich bereits die beiden Geräte Kühlschrank und Gefrierschrank von der Summenmessung abziehen. Diese Differenz wird für einen Tag beim bisher verwendeten Beispielhaushalt in Abbildung 4.124 dargestellt. Man erkennt hier jetzt wieder einige typische Verläufe, die zu bereits beschriebenen Geräten passen würden. Wie in Abbildung 4.125 klar ersichtlich würde dieser Leistungsverlauf zu einem Geschirrspüler passen.

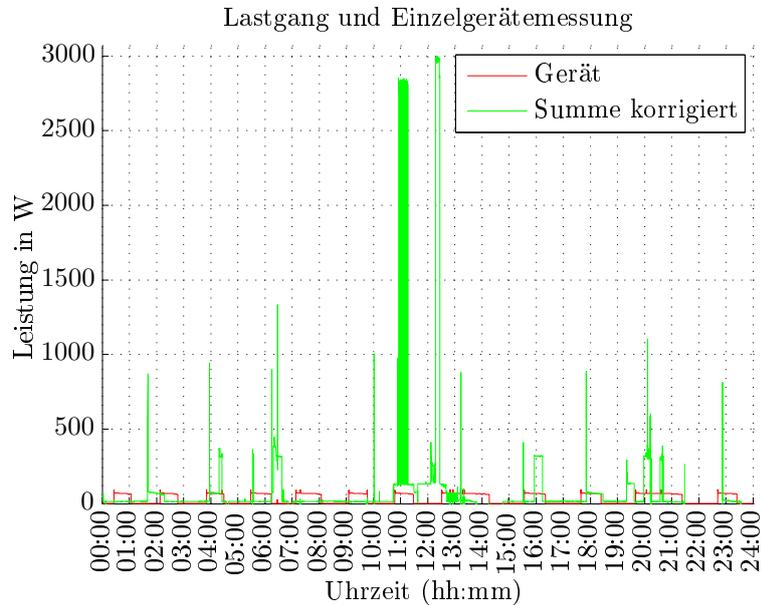


Abbildung 4.124.: Restprofil abzüglich Kühlschränke und Gefrierschrank

In Abbildung 4.126 sowie im Detail in Abbildung 4.127 ist diese Differenz auch ohne den ebenfalls detailliert vermessenen Geschirrspüler dargestellt. Man erkennt jetzt, dass hier die Zeitverschiebungen nicht während der gesamten Programmlaufzeit konstant sind sondern im Bereich von 1-3 Sekunden schwanken. Daher reicht bei diesen Geräten die vorher beschriebene Vorgehensweise nicht aus. Außerdem können während eines Zyklus wieder höhere Schaltflanken auftreten als in der Detailmessung erfasst. Auf Grund der Komplexität dieses Punktes wurde für die Bestimmung des Restprofils nicht weiter auf diese Fehler geachtet, da maximal 1-2 Zeitwerte pro Schaltflanke falsche Ergebnisse liefern und daher in den Abbildungen klar erkennbar sind und bei der Bestimmung der prozentualen im Detail vermessenen Leistung sehr einfach ausgeklammert werden können.

#### 4. Auswertungen

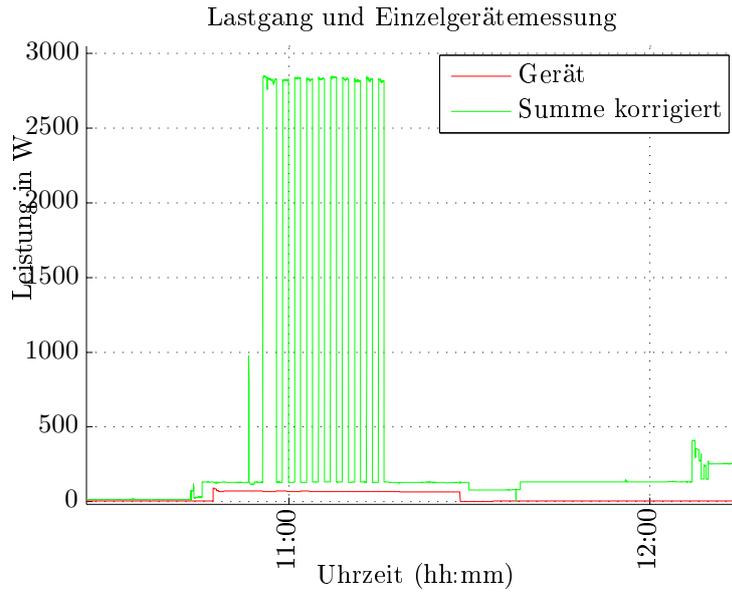


Abbildung 4.125.: Restprofil abzüglich Kühlschränke und Gefrierschrank - Zoom auf interessanten Bereich

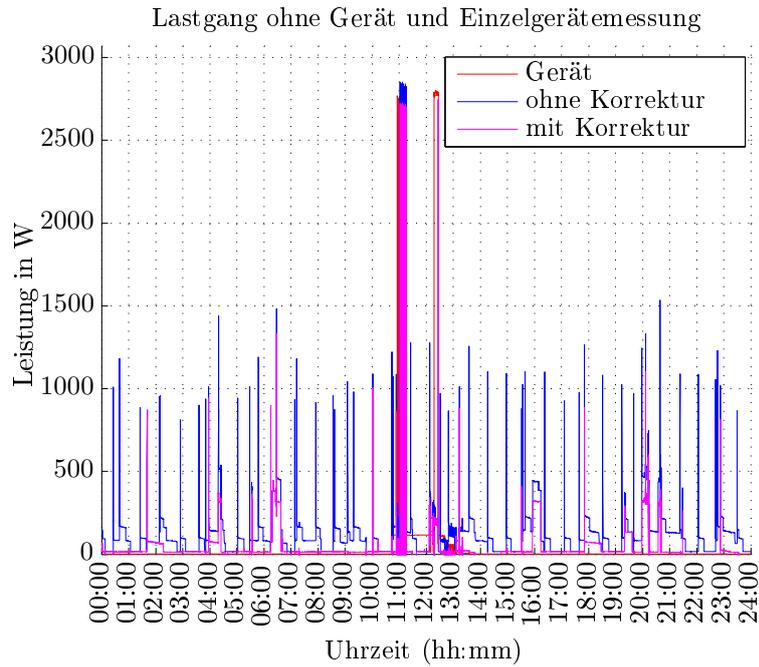


Abbildung 4.126.: Restprofil abzüglich Kühlschränke und Gefrierschrank sowie Geschirrspüler

#### 4. Auswertungen

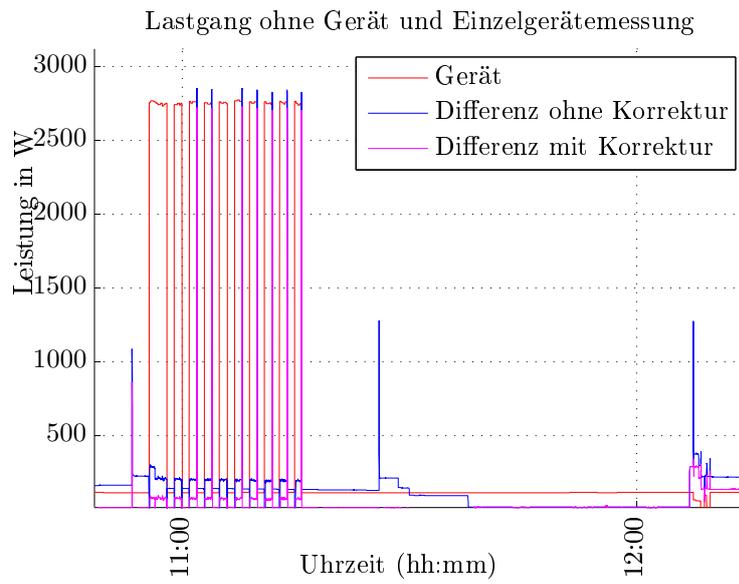


Abbildung 4.127.: Restprofil abzüglich Kühlschränke und Gefrierschrank sowie Geschirrspüler - Zoom auf interessanten Bereich

### 4.3.2. Betrachtungen für 4 Beispielhaushalte

In diesem Abschnitt wird nun das Restprofil von 4 Haushalten als exemplarisches Beispiel für Erkenntnisse aus dem Restprofil dargestellt und der Versuch unternommen, interessante Bereiche zu einzelnen Geräten zuzuordnen. In den folgenden Darstellungen sind die strichlierten Leistungsverläufe die Verläufe der Summenmessung und die durchgezogenen Linien stellen das Restprofil dar.

#### Haushalt 1

In Abbildung 4.128 ist nun das Restprofil des Beispielhaushaltes für einen Tag und die drei Phasen getrennt dargestellt. Man erkennt einen periodischen Verbraucher an der Phase L2, der auf Grund der Leistungsaufnahme von rund 2kW, der mittleren Laufzeit von wenigen Minuten und der Periodenlänge von einer knappen Stunde zu einem Warmwasserboiler passen würde. In Abbildung 4.129 ist einer dieser Zyklen dargestellt.

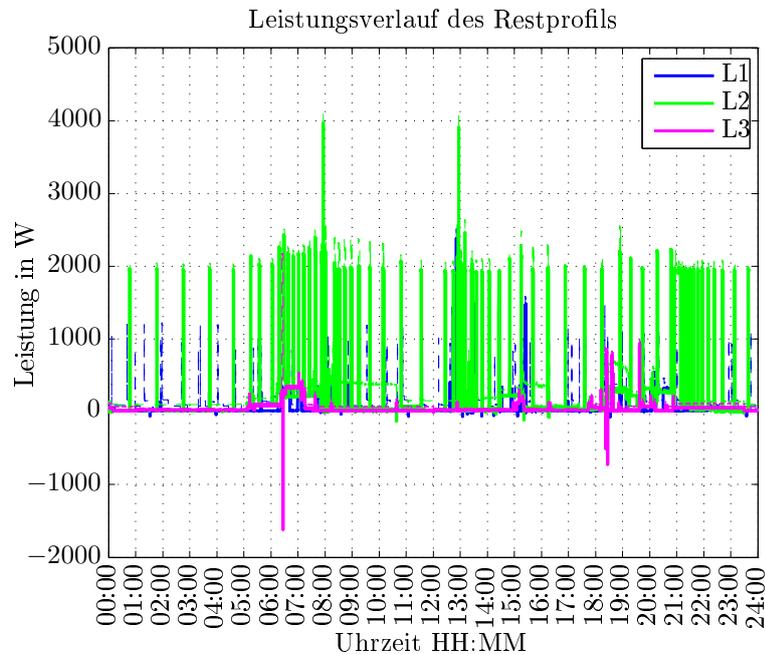


Abbildung 4.128.: Restprofil der drei Phasen für den Beispielhaushalt

#### 4. Auswertungen

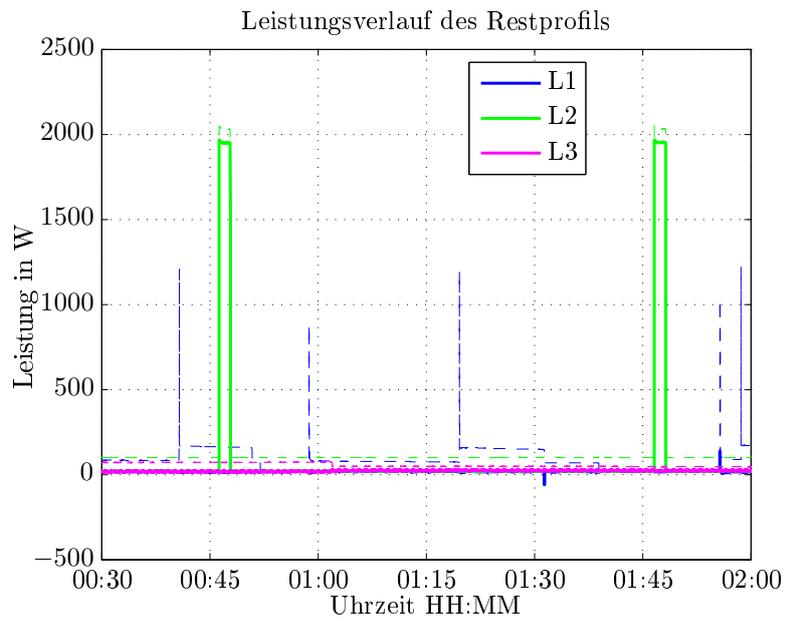


Abbildung 4.129.: Restprofil der drei Phasen für den Beispielhaushalt - Zoom auf Bereich

## Haushalt 2

Betrachtet man das Restprofil eines anderen Haushaltes, so erkennt man dort zwei weitere Phänomene (siehe Abbildung 4.130). Erstens wurden manche Zyklen anscheinend zweimal vom Summenprofil subtrahiert, da ihr Verlauf genau gespiegelt nochmals auftritt. Siehe dazu auch den in Abbildung 4.131 vergrößerten Ausschnitt. Ein zweites Phänomen ist der in Abbildung 4.136 dargestellte Verlauf. Auf Grund des Summenprofils würde man auf einen dreiphasigen Verbraucher schließen, dessen Leistungsverlauf aber im Restprofil nur mehr in der Phase L3 übriggeblieben ist. Auf Grund der Leistungsaufnahme von rund 500 W pro Phase und der kurzen Betriebszeit von knapp 2 Minuten könnte man auf einen Durchlauferhitzer schließen.

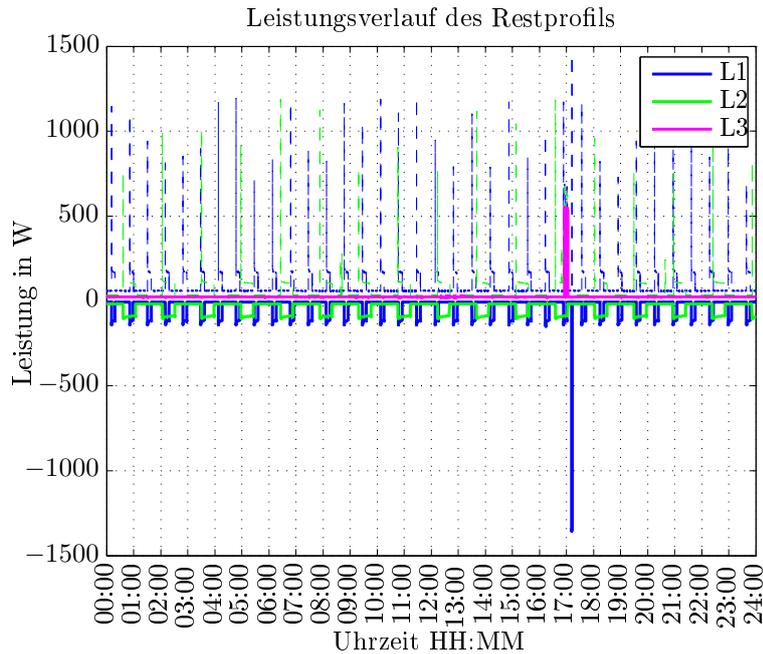


Abbildung 4.130.: Restprofil der drei Phasen für den 2. Beispielhaushalt

#### 4. Auswertungen

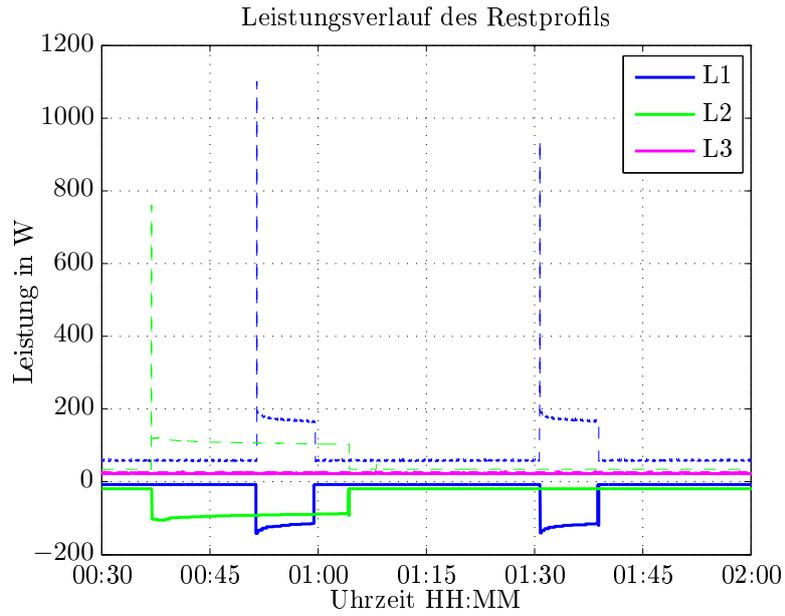


Abbildung 4.131.: Restprofil der drei Phasen für den 2. Beispielhaushalt - Zoom auf Bereich 1

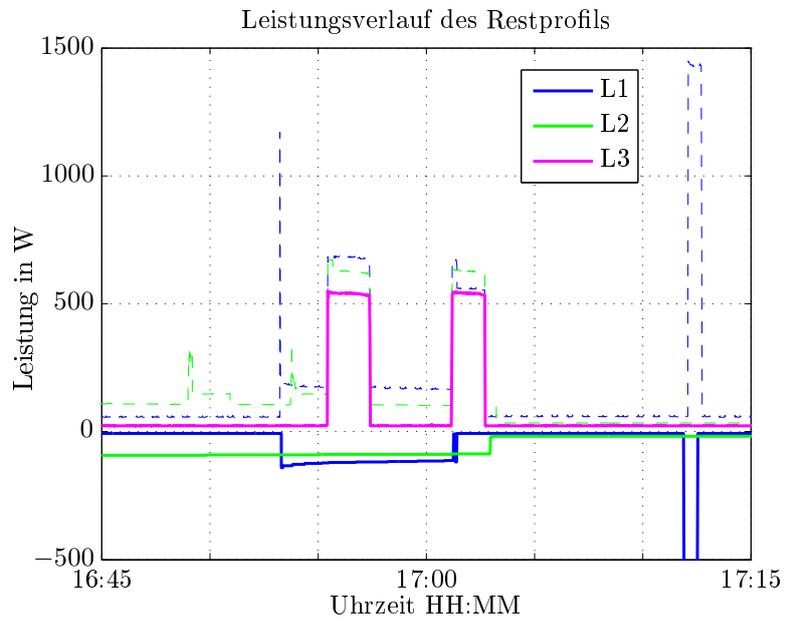


Abbildung 4.132.: Restprofil der drei Phasen für den 2. Beispielhaushalt - Zoom auf Bereich 2

## 4. Auswertungen

### Haushalt 3

Beim Haushalt 3 sieht man im Leistungsverlauf der Phase 3 immer wieder relativ kurze Verbrauchsspitzen mit rund 2 kW Leistung und einer Dauer von wenigen Minuten. Auf Grund des mehr oder weniger periodischen Einsatzes würde ich auf einen Elektroheizkörper zur Raumheizung schließen. Auch die Betriebsdauer mit erstem Einsatz um 08:00 Uhr und letzten Einsatz um ca. 21:30 würde diese Vermutung untermauern. Der Verlauf ist typisch für Heizstäbe mit einer leichten Spitze und dann langsamen Absinken der Leistung auf Grund der Erwärmung des Stabes und damit verbunden einer Erhöhung des elektrischen Widerstandes.

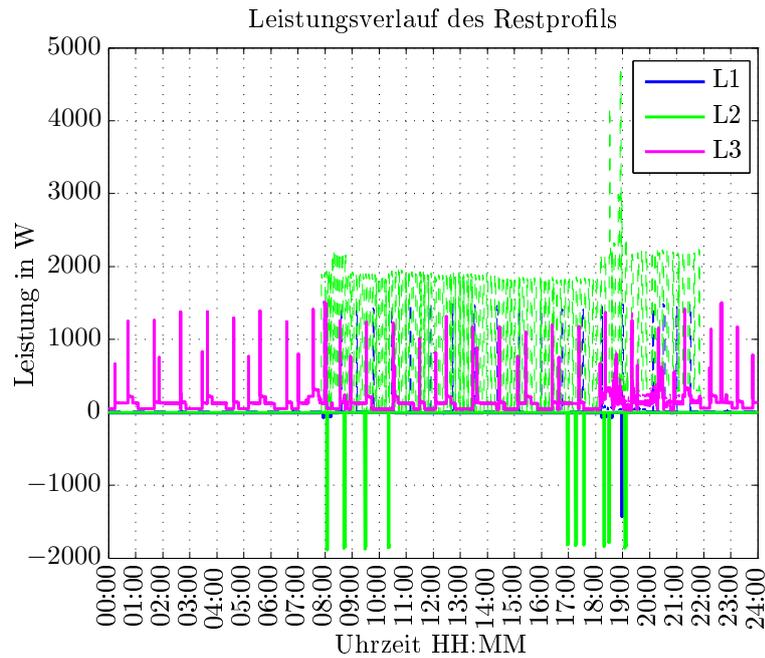


Abbildung 4.133.: Restprofil der drei Phasen für den 3. Beispielhaushalt

#### 4. Auswertungen

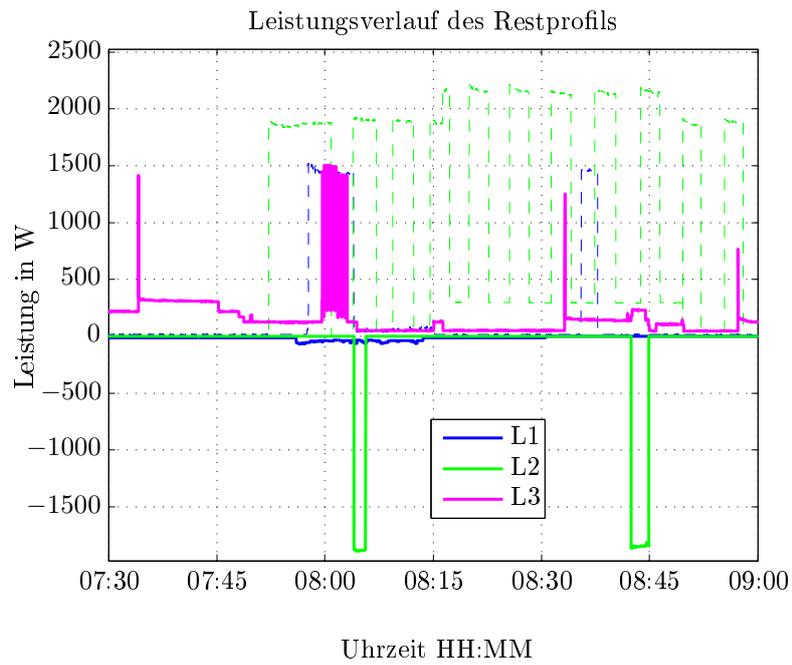


Abbildung 4.134.: Restprofil der drei Phasen für den 3. Beispielhaushalt - Zoom auf Bereich

## Haushalt 4

Bei einem vierten zufällig gewählten Haushalt erkennt man deutlich große Verbraucher in den Phasen L1 und L3. Auf Grund des charakteristischen Verlaufes für ein temperaturgesteuertes Gerät, der Gleichzeitigkeit des Verbrauches und des Zeitpunktes kurz vor Mittag lässt hier auf einen E-Herd schließen. Man kann annehmen, dass hier zuerst eine größere Platte des Herdes mit etwa 2kW in Betrieb genommen wurde und ca 25 Minuten später eine zweite kleinere Kochplatte mit etwa 1250W. Beide Kochfelder wurden dann gleichzeitig abgeschaltet.

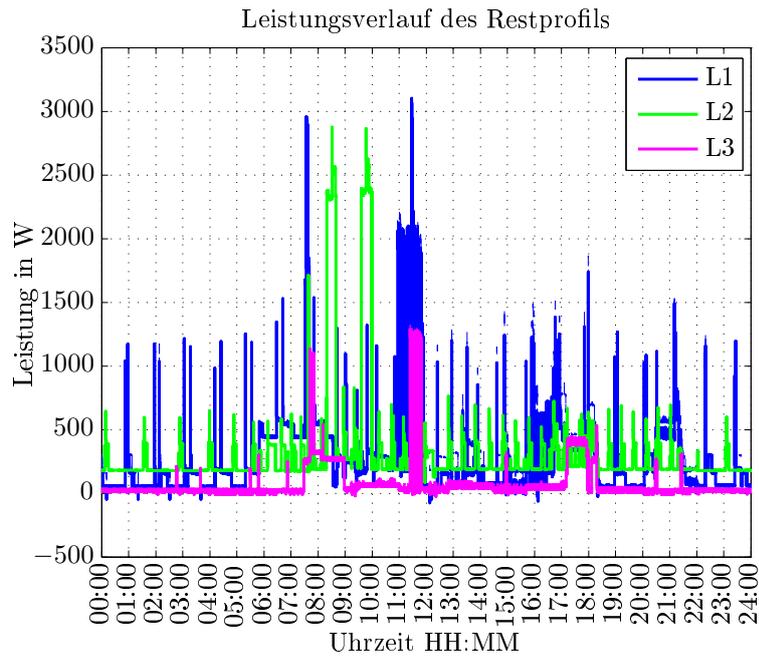


Abbildung 4.135.: Restprofil der drei Phasen für den 4. Beispielhaushalt

#### 4. Auswertungen

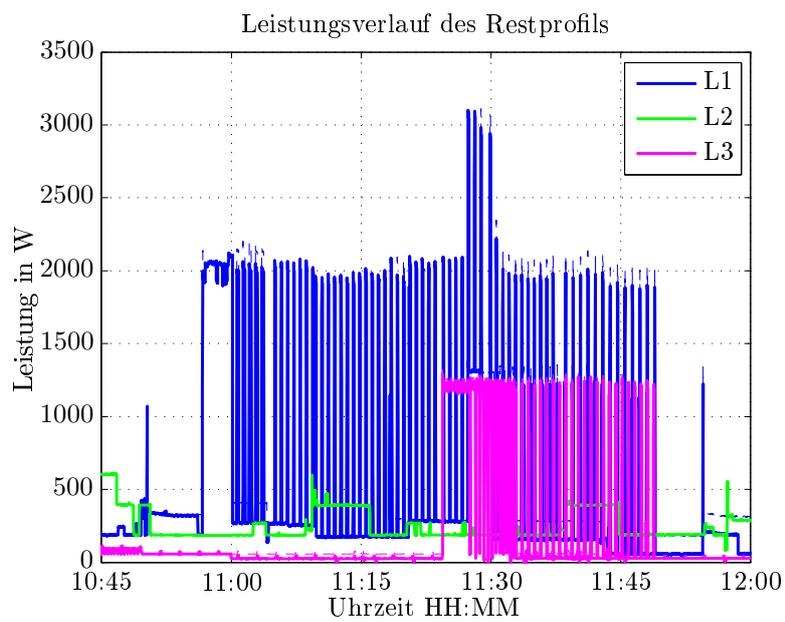


Abbildung 4.136.: Restprofil der drei Phasen für den 4. Beispielhaushalt - Zoom auf Bereich

### 4.3.3. prozentuell vermessene Leistung

In diesem Abschnitt wird eine Darstellung über die prozentual vermessene Leistung in den einzelnen Haushalten sowie dem Mittelwert über alle Haushalte dargestellt. In Abbildung 4.137 wird der Anteil der jeweiligen Geräteklassen am Gesamtverbrauch dargestellt. Man erkennt, dass die größte Gruppe in der Detailmessung die Geräteklasse „Sonstiges“ darstellt. Dies ist nicht weiter verwunderlich, da sich in dieser Klasse die mit Abstand größte Anzahl an Geräten befindet. Die Aufteilung auf die einzelnen Klassen ist in Abbildung 4.138 noch besser erkennbar. Dabei stellen die ersten 42 Balken die mittlere Verteilung pro Haushalt dar. Der 43. und somit äußerst rechte Balken in dieser Abbildung ist die mittlere Aufteilung auf die einzelnen Klassen für alle Haushalte. In dieser Grafik erkennt man auch einige deutliche Ausreißer, die auf einen Fehler im Detailmesssystem hinweisen. So erkennt man beispielsweise beim Haushalt mit der laufenden Nummer 32, dass der Wäschetrockner einen überproportionalen Anteil am Gesamtverbrauch aufweist. Bei genauerer Betrachtung ergibt sich, dass bei der Wintermessung in diesem Haushalt der gemessene Verbrauch über die gesamte Messdauer konstant rund 3.5kW war und diese Messung somit eindeutig falsch ist und ausgeklammert werden muss.

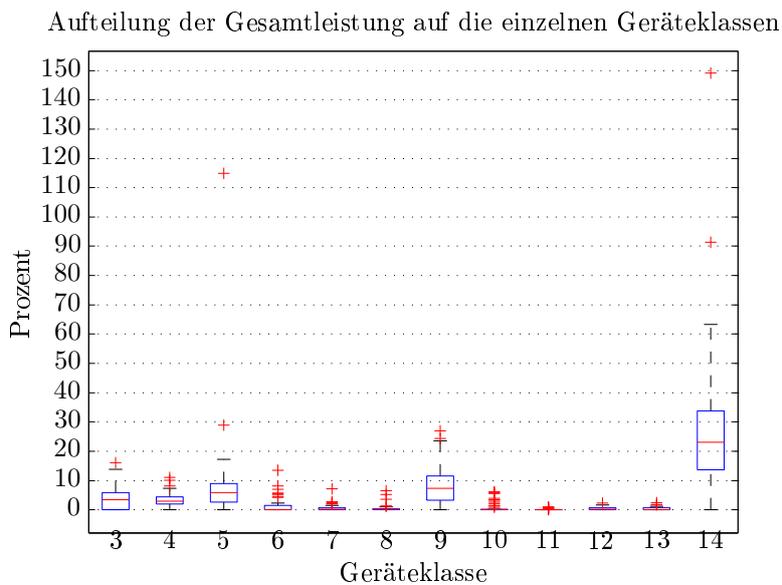


Abbildung 4.137.: prozentuelle Aufteilung der Summenleistung auf die einzelnen Geräteklassen

In Abbildung 4.139 wird der Anteil der Detailmessungen an der Summenmessung angegeben. Man erkennt dass der Mittelwert des prozentuellen Anteils bei etwa 50% liegt.

In der folgenden Abbildung 4.140 wird nochmals die mittlere Verteilung der Gesamt-

## 4. Auswertungen

Aufteilung der Gesamtleistung auf die einzelnen Geräteklassen

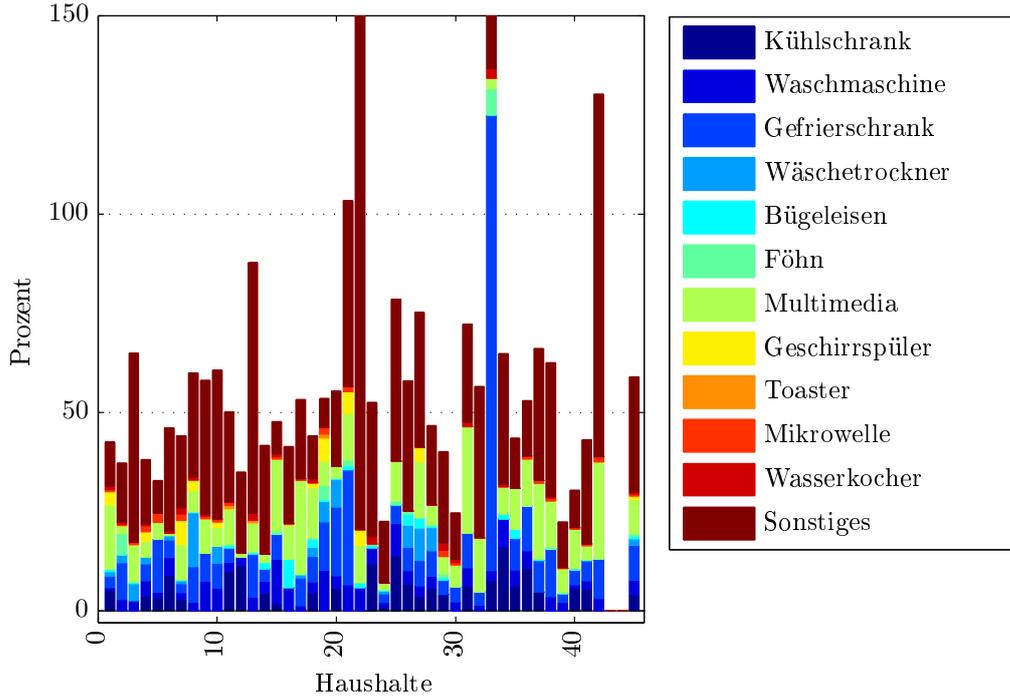


Abbildung 4.138.: prozentuelle Aufteilung der Summenleistung auf die einzelnen Geräteklassen

leistung auf die einzelnen Geräteklassen dargestellt. Weiters wird als Vergleich die Aufteilung laut Statistik Austria (Quelle [5]) dargestellt. Man erkennt doch deutliche Unterschiede in der Aufteilung. Der größte Unterschied ist beim Geschirrspüler zu beobachten. Dieser ist dadurch erklärbar, dass bei dieser Messreihe nur in wenigen Haushalten ein Geschirrspüler vermessen wurde und dadurch die prozentuale Aufteilung zu niedrig ausfällt.

## 4. Auswertungen

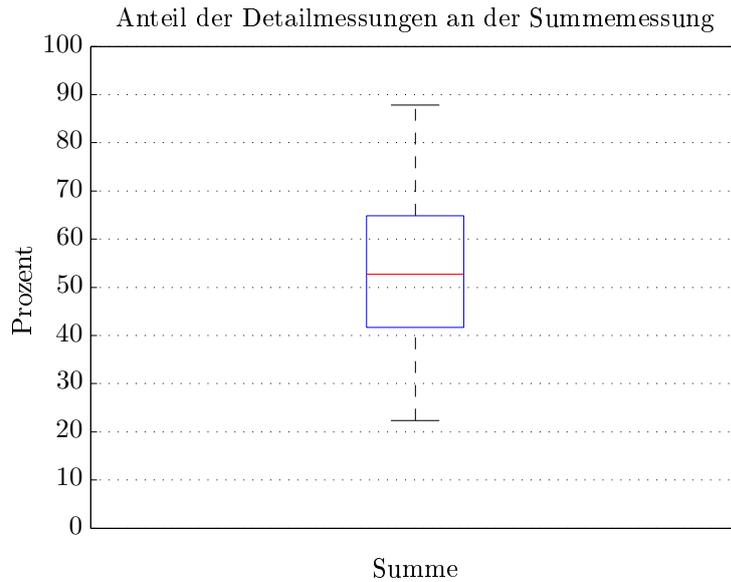


Abbildung 4.139.: prozentueller Anteil der Detailmessungen an der Summenleistung

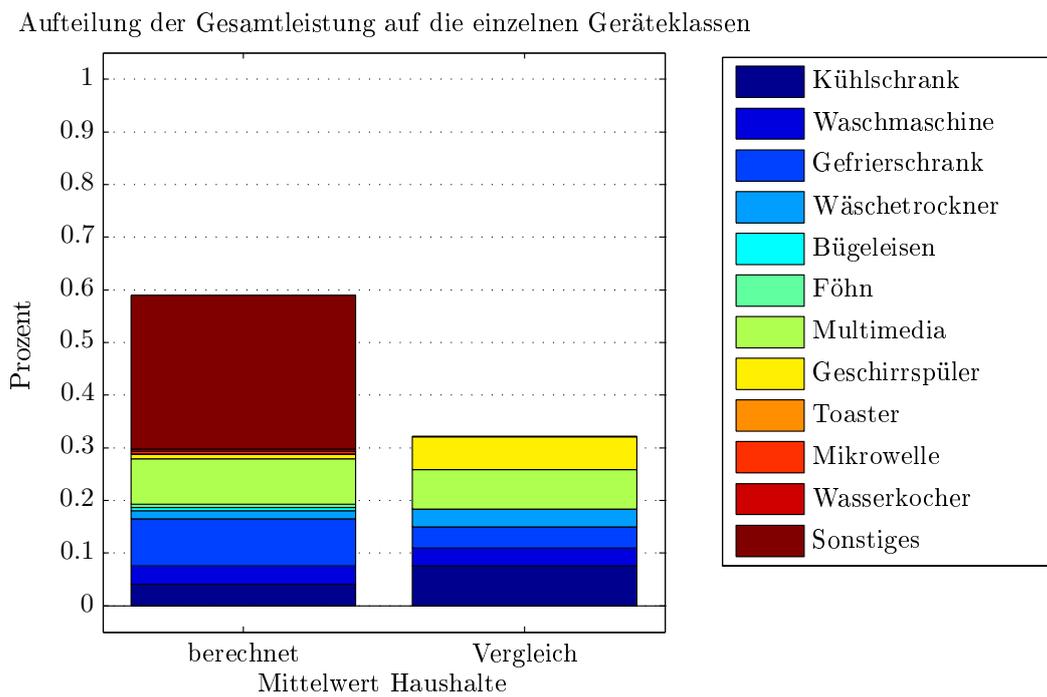


Abbildung 4.140.: prozentuelle Aufteilung der Summenleistung auf die einzelnen Geräteklassen - Mittelwert über alle Haushalte

## 5. Schlussfolgerung

Den Energiebedarf von Haushalten möglichst genau prognostizieren und somit den Gesamtbedarf an Leistung mit relativ wenig Fehlern bestimmen zu können, ist in Zeiten von dezentralen Einspeisepunkten und stark schwankendem Dargebot an erneuerbaren Energien sehr wichtig. Da man die einspeisende Leistung vielerorts nicht mehr im Voraus kennt und sie auf Grund von Abschattungen von Photovoltaik-Anlagen in Sekundenbruchteilen von der Nennleistung der Anlage auf null zurückgehen kann, ist es umso wichtiger zumindest den Bedarf an Energie zu kennen. Da mit falsch prognostizierten Bedarfswerten oder falschen Einspeiseleistungen, die auf Grund der Priorisierung von Erneuerbarer Energie meistens nicht in der Hand der Energieversorger liegen, teilweise relativ hohe Kosten für benötigte Ausgleichsenergie verbunden sind, soll hier der bereits seit Jahren mit dem gleichen Modell ermittelte Haushaltsbedarf kritisch hinterfragt werden.

Um diesem Ziel näher zu kommen, wurde zuerst der Einsatz einzelner Geräte sehr genau beleuchtet. Dabei wurden einerseits die Einsatzwahrscheinlichkeiten, die mittlere Nutzungsdauer, Nennleistung der Geräte und andererseits auch der mittlere Leistungsverbrauch der jeweiligen Geräte über einen Tag verteilt ermittelt. Ein besonderes Augenmerk wurde auch auf den Leistungsverlauf während der Betriebszyklen des betrachteten Gerätes gelegt. Dabei wurde versucht, aus der Summenmessung der Blindleistung auch eine Abschätzung des Leistungsfaktors vorzunehmen. Da über den Leistungsverlauf großer Haushaltsgeräte kaum Informationen verfügbar sind, dieser Verlauf aber für Inselösungen besonders interessant ist. Bei solchen kleinen Netzen ist zu beachten, dass eine Mittlung der Verläufe mehrere Geräte nur in vermindertem Ausmaß beobachtet wird und zur Netzstabilität eine sekundengenaue Bedarfsdeckung unbedingt erforderlich ist.

Im zweiten Schritt wurden dann die Summenmessungen der Haushalte mit einem H0-Profil verglichen. Dabei wurden für eine unterschiedliche Anzahl an Haushalten jeweils der mittlere Leistungsverlauf ermittelt und verglichen. Je größer diese Anzahl war, desto näher schmiegt sich der Verlauf an das H0-Profil an. Damit konnte bestätigt werden, dass im Mittel zur Bedarfsermittlung das H0-Profil herangezogen werden kann.

## 5. Schlussfolgerung

Im dritten und letzten Schritt wurde ein Versuch unternommen, alle nicht im Detail vermessenen Geräte etwas näher zu beschreiben. Auf Grund von einigen unerwarteten Schwierigkeiten bei der Differenzbildung zwischen Summenmessung und Detailmessung konnte aus Zeitgründen nicht näher auf das so genannte Restprofil eingegangen werden. Einige exemplarische Auswertungen werden dennoch angeführt. Weiters wird die Aufteilung der Gesamtleistung auf die einzelnen Geräteklassen in diesem Abschnitt beleuchtet. Auch hier sind einige Ungereimtheiten auf Grund einiger Messfehler in der Detailmessung erkennbar.

# Abbildungsverzeichnis

1.1. Baumdiagramm der Rohdaten . . . . .	3
2.1. Baumdiagramm der Informationsdatenstruktur <code>geraete</code> . . . . .	8
2.2. Baumdiagramm der Informationsdatenstruktur <code>cycle_info</code> . . . . .	9
2.3. Baumdiagramm der Informationsdatenstruktur <code>hh_data</code> . . . . .	12
2.4. Grafisches User-Interface (GUI) . . . . .	23
4.1. Einschalthäufigkeit für Kühlschränke . . . . .	28
4.2. Einschaltdauer für Kühlschränke . . . . .	29
4.3. Nennleistung für Kühlschränke . . . . .	30
4.4. Nennleistung für Kühlschränke - Boxplot . . . . .	31
4.5. Nennleistung für Kühlschränke - Ausschnitt aus dem Boxplot . . . . .	32
4.6. Energieverbrauch der untersuchten Kühlschränke . . . . .	33
4.7. Zyklen pro Tag für Kühlschränke . . . . .	34
4.8. Zyklen pro Tag für Kühlschränke (korrigiert) . . . . .	34
4.9. mittlerer Leistungsverlauf eines Kühlschranks über den Tag verteilt . . . . .	35
4.10. Vergleich der Einschalthäufigkeit eines Kühlschranks für verschiedene Tagestypen . . . . .	35
4.11. Blindleistungsverlauf eines Kühlschranks . . . . .	36
4.12. Zeigerdiagramm für den betrachteten Zeitpunkt eines Kühlschranks . . . . .	36
4.13. Einschalthäufigkeit für Gefrierschränke . . . . .	37
4.14. Einschaltdauer für Gefrierschränke . . . . .	38
4.15. Nennleistung für Gefrierschränke . . . . .	39
4.16. Nennleistung für Gefrierschränke - Boxplot . . . . .	40
4.17. Energieverbrauch pro Jahr für Gefrierschränke mit Energieeffizienzklassen nach [12] . . . . .	41
4.18. Zyklen pro Tag für Gefrierschränke . . . . .	41
4.19. mittlere Tagesleistung für Gefrierschränke . . . . .	42
4.20. Vergleich der Einschalthäufigkeit von Sommer und Winter für Gefrier- schränke . . . . .	42
4.21. Zeigerdiagramm für den betrachteten Zeitpunkt eines Gefrierschranks . . . . .	43
4.22. Leistungsverlauf für einen typischen Zyklus für Gefrierschränke . . . . .	43
4.23. Einschaltzeitpunkte von Waschmaschinen . . . . .	44
4.24. Einschaltzeitpunkte von Waschmaschinen im Sommer . . . . .	45
4.25. Einschaltzeitpunkte von Waschmaschinen im Winter . . . . .	46
4.26. Programmlaufzeit von Waschmaschinen . . . . .	47
4.27. Programmlaufzeit von Waschmaschinen (korrigiert auf maximal 6 Stunden) . . . . .	48

## Abbildungsverzeichnis

4.28. Waschmaschine, zwei Zyklen als ein zusammenhängender erkannt . . . . .	49
4.29. Waschmaschine, Nennleistung Maximalwert . . . . .	50
4.30. Waschmaschine, Boxplot von maximaler sowie durchschnittlicher Nennleistung . . . . .	51
4.31. Waschmaschine, Energieverbrauch je Zyklus und Energieeffizienzklassen laut [12] . . . . .	51
4.32. Waschmaschine, Leistungsverlauf eines Zyklus . . . . .	52
4.33. Waschmaschine, Wirk-, und Blindleistungsverlauf eines Zyklus . . . . .	52
4.34. Waschmaschine, Verlauf der mittleren Tagesleistung mit Vergleichsverlauf nach [9] . . . . .	53
4.35. Waschmaschine, Einschalthäufigkeit pro Tag . . . . .	53
4.36. Waschmaschine, Einschalthäufigkeit pro Tag mit Tagen ohne Zyklus . . . . .	54
4.37. Waschmaschine, Vergleich der Einschalthäufigkeit pro Tag für Sommer und Winter . . . . .	54
4.38. Einschaltzeitpunkte von Wäschetrocknern . . . . .	55
4.39. Programmlaufzeit von Wäschetrocknern . . . . .	56
4.40. Nennleistung von Wäschetrocknern . . . . .	57
4.41. Nennleistung von Wäschetrocknern - Boxplot . . . . .	58
4.42. Einschaltzyklen / Tag für Wäschetrockner . . . . .	59
4.43. Energieverbrauch von Wäschetrocknern mit Referenzklassen nach [12] . . . . .	59
4.44. Leistungsaufnahme während eines Zyklus von Wäschetrocknern . . . . .	60
4.45. Vergleich der Verwendung von Trocknern zwischen Sommer und Winter . . . . .	60
4.46. mittlere Tagesleistung für Wäschetrockner mit Vergleichsverlauf nach [9] . . . . .	61
4.47. Einschaltzeitpunkt von Geschirrspülern . . . . .	62
4.48. Programmlaufzeit von Geschirrspülern . . . . .	63
4.49. Nennleistung von Geschirrspülern . . . . .	64
4.50. Nennleistung von Geschirrspülern - Boxplot . . . . .	65
4.51. Energieverbrauch von Geschirrspülern mit den Energieeffizienzklassen nach [12] . . . . .	66
4.52. Einschaltzyklen pro Tag . . . . .	66
4.53. Leistungsverlauf eines Geschirrspülers . . . . .	67
4.54. Leistungsverlauf eines Geschirrspülers . . . . .	67
4.55. Leistungsverlauf eines Geschirrspülers . . . . .	68
4.56. mittlerer Tagesleistungsverlauf eines Geschirrspülers mit dem Vergleichsverlauf nach [9] . . . . .	68
4.57. Einschaltzeitpunkte für Wasserkocher . . . . .	69
4.58. Einschaltdauer für Wasserkocher . . . . .	70
4.59. Nennleistung für Wasserkocher . . . . .	71
4.60. Nennleistung für Wasserkocher - Boxplot . . . . .	71
4.61. Energieverbrauch pro Zyklus für Wasserkocher . . . . .	72
4.62. mittlerer Tagesleistungsverbrauch für Wasserkocher . . . . .	72
4.63. Einschaltzeitpunkte für Toaster . . . . .	73
4.64. Einschaltdauer für Toaster . . . . .	74
4.65. Nennleistung für Toaster . . . . .	74

## Abbildungsverzeichnis

4.66. Energieverbrauch pro Zyklus für Toaster . . . . .	75
4.67. Leistungsverlauf während eines Zyklus für Toaster . . . . .	75
4.68. Leistungsverlauf während eines Zyklus für Toaster . . . . .	76
4.69. Einschaltzeitpunkte für Mikrowellen . . . . .	77
4.70. Einschaltdauer für Mikrowellen . . . . .	78
4.71. Nennleistung für Mikrowellen . . . . .	79
4.72. Leistungsverlauf für Mikrowellen . . . . .	79
4.73. Leistungsverlauf für Mikrowellen . . . . .	80
4.74. Leistungsverlauf für Mikrowellen . . . . .	80
4.75. Vergleich der Einschalthäufigkeit für Mikrowellen im Sommer . . . . .	81
4.76. Vergleich der Einschalthäufigkeit für Mikrowellen im Winter . . . . .	81
4.77. mittlere Tagesleistung für Mikrowellen . . . . .	82
4.78. Einschaltzeitpunkte für Multimedia-Geräte . . . . .	83
4.79. Einschaltzeitpunkte für Multimedia-Geräte - Ausschnitt bis 200 Zyklen .	84
4.80. mittlere Leistungsaufnahme für Multimedia-Geräte mit Vergleichsverlauf nach [9] . . . . .	85
4.81. Vergleich der Einschalthäufigkeit für Multimedia-Geräte Sommer vs. Winter . . . . .	85
4.82. Vergleich der Einschalthäufigkeit für Multimedia-Geräte Winter Werktag vs. Wochenende . . . . .	86
4.83. Vergleich der Einschalthäufigkeit für Multimedia-Geräte Sommer Werktag vs. Wochenende . . . . .	86
4.84. Vergleich der Nennleistung für Multimedia-Geräte . . . . .	87
4.85. Vergleich der Einschaltdauer für Multimedia-Geräte . . . . .	87
4.86. Vergleich der Einschaltdauer für Multimedia-Geräte - Ausschnitt bis 1 Stunde . . . . .	88
4.87. Einschaltzeitpunkte für die untersuchten Bügeleisen . . . . .	90
4.88. Einsatzdauer für die untersuchten Bügeleisen . . . . .	90
4.89. Nennleistungen für die untersuchten Bügeleisen . . . . .	91
4.90. Leistungsverlauf 1 für die untersuchten Bügeleisen . . . . .	91
4.91. Leistungsverlauf 2 für die untersuchten Bügeleisen . . . . .	92
4.92. Vergleich des Nutzungsverhaltens im Sommer der untersuchten Bügeleisen	92
4.93. Vergleich des Nutzungsverhaltens im Winter für die untersuchten Bügeleisen	93
4.94. mittlere Tagesleistung für die untersuchten Bügeleisen . . . . .	93
4.95. Einsatzzeitpunkte für Föhns . . . . .	94
4.96. Einschaltdauer für Föhns . . . . .	95
4.97. Nennleistung für Föhns . . . . .	95
4.98. Leistungsverlauf für Föhns . . . . .	96
4.99. Leistungsverlauf 2 für Föhns . . . . .	96
4.100 mittlere Tagesleistung für Föhns . . . . .	97
4.101 Einschaltzeitpunkte für nicht näher unterteilte Geräte . . . . .	98
4.102 Nennleistung für nicht näher unterteilte Geräte . . . . .	99
4.103 Nennleistung für nicht näher unterteilte Geräte - Boxplot . . . . .	99
4.104 Lastprofil von einem Haushalt und H0-Vergleichsprofil . . . . .	102

4.105	Lastprofil von 5 zufällig gewählten Haushalten und H0-Vergleichsprofil . .	103
4.106	Lastprofil von 10 zufällig gewählten Haushalten und H0-Vergleichsprofil .	103
4.107	Lastprofil von 15 zufällig gewählten Haushalten und H0-Vergleichsprofil .	104
4.108	Lastprofil von 20 zufällig gewählten Haushalten und H0-Vergleichsprofil .	104
4.109	Lastprofil von 30 zufällig gewählten Haushalten und H0-Vergleichsprofil .	105
4.110	Lastprofil von allen 42 Haushalten und H0-Vergleichsprofil . . . . .	105
4.111	Differenz zwischen H0-Vergleichsprofil und Haushaltslastgang für verschie- dene Anzahl an Haushalten und Tagen . . . . .	106
4.112	Lastprofil von einem zufällig gewählten Haushalt . . . . .	107
4.113	Lastprofil von einem zufällig gewählten Haushalt . . . . .	108
4.114	Ausschnitt aus dem Lastprofil von einem zufällig gewählten Haushalt . .	108
4.115	Lastprofil von einem zufällig gewählten Haushalt . . . . .	109
4.116	Ausschnitt aus dem Lastprofil von einem zufällig gewählten Haushalt . .	110
4.117	Leistungsfaktor $\lambda$ und Wirkleistung P von allen Haushalten . . . . .	111
4.118	Summenlastprofil und Einzelgerätemessung Gefrierschrank vor der Kor- rektur . . . . .	113
4.119	Summenlastprofil und Einzelgerätemessung Gefrierschrank - Detailansicht eines Zyklus vor der Korrektur . . . . .	114
4.120	Summenlastprofil und Einzelgerätemessung Gefrierschrank nach der Kor- rektur . . . . .	115
4.121	Summenlastprofil und Einzelgerätemessung Gefrierschrank nach der Kor- rektur - Detailansicht eines Zyklus . . . . .	116
4.122	Summenlastprofil 3 Phasen und Einzelgerätemessung Kühlschränk . . . .	117
4.123	Summenlastprofil 3 Phasen und Einzelgerätemessung Kühlschränk De- tailansicht eines Zyklus . . . . .	118
4.124	Restprofil abzüglich Kühlschränke und Gefrierschrank . . . . .	119
4.125	Restprofil abzüglich Kühlschränke und Gefrierschrank - Zoom auf inter- essanten Bereich . . . . .	120
4.126	Restprofil abzüglich Kühlschränke und Gefrierschrank sowie Geschirrspüler	120
4.127	Restprofil abzüglich Kühlschränke und Gefrierschrank sowie Geschirrspü- ler - Zoom auf interessanten Bereich . . . . .	121
4.128	Restprofil der drei Phasen für den Beispielhaushalt . . . . .	122
4.129	Restprofil der drei Phasen für den Beispielhaushalt - Zoom auf Bereich .	123
4.130	Restprofil der drei Phasen für den 2. Beispielhaushalt . . . . .	124
4.131	Restprofil der drei Phasen für den 2. Beispielhaushalt - Zoom auf Bereich	125
4.132	Restprofil der drei Phasen für den 2. Beispielhaushalt - Zoom auf Bereich	125
4.133	Restprofil der drei Phasen für den 3. Beispielhaushalt . . . . .	126
4.134	Restprofil der drei Phasen für den 3. Beispielhaushalt - Zoom auf Bereich	127
4.135	Restprofil der drei Phasen für den 4. Beispielhaushalt . . . . .	128
4.136	Restprofil der drei Phasen für den 4. Beispielhaushalt - Zoom auf Bereich	129
4.137	prozentuelle Aufteilung der Summenleistung auf die einzelnen Geräteklassen	130
4.138	prozentuelle Aufteilung der Summenleistung auf die einzelnen Geräteklassen	131
4.139	prozentueller Anteil der Detailmessungen an der Summenleistung . . . .	132

*Abbildungsverzeichnis*

4.140prozentuelle Aufteilung der Summenleistung auf die einzelnen Geräteklas-  
sen - Mittelwert über alle Haushalte . . . . . 132

## Tabellenverzeichnis

4.1. Zusammenstellung der wichtigsten Daten für zyklische Geräte . . . . .	40
4.2. Zusammenfassung der wichtigsten Geräteparameter . . . . .	100
A.1. Geräteübersicht der ADRES Messreihe . . . . .	147

## Literaturverzeichnis

- [1] E.ON Mitte AG. *E.ON Mitte AG | Standardlastprofil- verfahren*. [online, letzter Aufruf: 29.Jan. 2014]. URL: [http://www.eon-mitte.com/de/netz/veroeffentlichungen/strom\\_/standardlastprofil\\_verfahren](http://www.eon-mitte.com/de/netz/veroeffentlichungen/strom_/standardlastprofil_verfahren).
- [2] Anatol Amon. *Analyse von Kundenlastprofilen und automatische Zerlegung in Gerätelastgänge*. Parallelt. [Übers. des Autors] *Analyse of household load profiles and automatic detection of device load profiles*. 2011. URL: <http://media.obvsg.at/AC07812062-3001>.
- [3] Statistik Austria. *STATISTIK AUSTRIA - Ausstattung privater Haushalte*. [online, letzter Aufruf: 02.Jan. 2014]. 2014. URL: [http://www.statistik.at/web\\_de/statistiken/soziales/ausstattung\\_privater\\_haushalte/021850.html](http://www.statistik.at/web_de/statistiken/soziales/ausstattung_privater_haushalte/021850.html).
- [4] Statistik Austria. *STATISTIK AUSTRIA - Energiebilanzen*. [online, letzter Aufruf: 02.Jan. 2014]. URL: [http://www.statistik.at/web\\_de/static/bilanz\\_der\\_elektrischen\\_energie\\_022711.pdf](http://www.statistik.at/web_de/static/bilanz_der_elektrischen_energie_022711.pdf).
- [5] Statistik Austria. *STATISTIK AUSTRIA - Energieeinsatz der Haushalte*. URL: [http://www.statistik.at/web\\_de/static/durchschnittlicher\\_stromverbrauch\\_der\\_haushalte\\_2012\\_nach\\_verbrauchskatego\\_071028.pdf](http://www.statistik.at/web_de/static/durchschnittlicher_stromverbrauch_der_haushalte_2012_nach_verbrauchskatego_071028.pdf).
- [6] Alfred Einfalt et al. "Konzeptentwicklung für ADRES - Autonome Dezentrale Erneuerbare Energie Systeme, FFG-Forschungsprojekt, Energie der Zukunft, 1. AS, Projektnummer: 815674, Endbericht," in: (2011). URL: [http://www.ea.tuwien.ac.at/fileadmin/t/ea/projekte/ADRES\\_Concept/PublizierbarerEndberichtADRES\\_815674.pdf](http://www.ea.tuwien.ac.at/fileadmin/t/ea/projekte/ADRES_Concept/PublizierbarerEndberichtADRES_815674.pdf).
- [7] Sara Ghaemi. *Efficiency potential in private sector in ADRES*. Zsfassung in dt. Sprache. 2011. URL: <http://media.obvsg.at/AC07810532-3001>.
- [8] EWE NETZ GmbH. *EWE NETZ GmbH - Lastprofile*.
- [9] Christoph Groiss. *Power Demand Side Management*. Wien, Techn. Univ., Dipl.-Arb., 2008. 2008. URL: <http://media.obvsg.at/AC07132280-2001>.
- [10] Haier.com. *Waschmaschine GEBRAUCHSANLEITUNG*. [online, letzter Aufruf: 29.Jan. 2014]. 2014. URL: <http://image.haier.com/manual/Laundry/wm/201108/P020110824794025835553.pdf>.
- [11] <http://www.ersatzteilprofi.at>. *Ersatzteilprofi.at - Ersatzteile Waschmaschinen*. [online, letzter Aufruf: 04.Dez. 2013]. 2013. URL: [http://www.ersatzteilprofi.at/index.php?menu=1&smenu=11&mpid=comm\\_sresult1&lpid=111\\_left&rubric-a=21&rubric-b=00&rubric-c=00](http://www.ersatzteilprofi.at/index.php?menu=1&smenu=11&mpid=comm_sresult1&lpid=111_left&rubric-a=21&rubric-b=00&rubric-c=00).

- [12] <http://www.miraconsult.at/haushaltsgeraete-einkaugshilfenumweltzeichenlabels.pdf>. [online, letzter Aufruf: 04.Dez. 2013]. 2013. URL: <http://www.miraconsult.at/wp-content/uploads/haushaltsgeraete-einkaugshilfenumweltzeichenlabels.pdf>.
- [13] kuhlschrank.org. *Leistungsaufnahme eines Kühlschranks und Energieeffizienzklassen*. [online, letzter Aufruf: 04.Dez. 2013]. 2013. URL: <http://kuhlschrank.org/leistungsaufnahme-eines-kuhlschranks-und-energieeffizienzklassen/>.
- [14] Medienforschung ORF. *Medienforschung ORF, medienforschung.orf.at*. 2013. URL: [http://mediaresearch.orf.at/c\\_radio/console/console.htm?y=4&z=3](http://mediaresearch.orf.at/c_radio/console/console.htm?y=4&z=3).
- [15] Medienforschung ORF. *Medienforschung ORF, medienforschung.orf.at*. 2013. URL: [http://mediaresearch.orf.at/c\\_fernsehen/console/console.htm?y=1&z=4](http://mediaresearch.orf.at/c_fernsehen/console/console.htm?y=1&z=4).
- [16] Zebb Prime. *matlabfrag - File Exchange - MATLAB Central*. [online, letzter Aufruf: 26.01.2014]. 2008. URL: <http://www.mathworks.com/matlabcentral/fileexchange/21286-matlabfrag>.
- [17] Thomas Siebenhüner. *Demand-Side-Management von Kleinverbrauchern*. Wien, Techn. Univ., Dipl.-Arb., 2006. 2006, 54 Bl. URL: <http://permalink.obvsg.at/AC05792406>.
- [18] Wikipedia. *Boxplot — Wikipedia, Die freie Enzyklopädie*. [Online; Stand 7. Februar 2014]. 2013. URL: <http://de.wikipedia.org/w/index.php?title=Boxplot&oldid=125106667>.
- [19] Wikipedia. *Einphasen-Reihenschlussmotor — Wikipedia, Die freie Enzyklopädie*. [Online; Stand 11. Februar 2014]. 2014. URL: <http://de.wikipedia.org/w/index.php?title=Einphasen-Reihenschlussmotor&oldid=127189815>.

## A. Übersicht über die vermessenen Geräte

Diese hier beschriebenen Klassen bilden auch die Grundlage der weiteren Auswertungen, da jede Klasse für sich auf ihre charakteristischen Eigenschaften untersucht wird. Die Nummern in der folgenden Aufzählung korrespondieren mit den Spaltenüberschriften in der unten stehenden Tabelle.

1. Haushaltsname
2. Sommer-, oder Wintermessung
3. Kühlschrank: Kühlschränke und Kühl-, Gefrierkombigeräte
4. Waschmaschine
5. Gefrierschrank
6. Wäschetrockner
7. Föhn
8. Multimedia: TV, HiFi, SAT-Receiver, DVD-Player, Switch, Router, Radio, Laptop, PC, Drucker,...
9. Geschirrspüler
10. Toaster
11. Mikrowelle
12. Wasserkocher
13. Sonstiges: alle übrigen Geräte
14. Gesamt
15. Gesamt L1
16. Gesamt L2
17. Gesamt L3

*A. Übersicht über die vermessenen Geräte*

A. Übersicht über die vermessenen Geräte

filename	Messung	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADRES001.mat	Sommer	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ADRES001.mat	Winter	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ADRES002.mat	Sommer		X	X	X		X	X		X	X	X	X	X	X	X	X
ADRES002.mat	Winter		X	X			X	X		X			X	X	X	X	X
ADRES003.mat	Winter		X		X	X	X	X			X		X	X	X	X	X
ADRES003.mat																	
ADRES004.mat	Sommer	X	X	X	X			X	X	X	X	X	X	X	X	X	X
ADRES004.mat	Winter	X	X	X	X			X		X	X		X	X	X	X	X
ADRES005.mat	Winter	X	X	X				X			X		X	X	X	X	
ADRES005.mat																	
ADRES006.mat	Sommer	X	X	X	X			X					X	X	X	X	X
ADRES006.mat	Winter	X	X	X	X							X	X	X	X	X	X
ADRES007.mat	Sommer	X	X	X	X	X		X	X		X	X	X	X	X	X	X
ADRES007.mat	Winter	X	X	X	X	X		X	X		X	X	X	X	X	X	X
ADRES008.mat	Sommer		X	X		X	X	X	X	X		X	X	X	X	X	X
ADRES008.mat	Winter		X	X	X	X		X		X		X	X	X	X	X	X
ADRES009.mat	Sommer		X	X	X		X	X		X	X	X	X	X	X	X	X
ADRES009.mat	Winter			X	X		X			X	X	X	X	X	X	X	X
ADRES010.mat	Sommer		X	X	X			X			X		X	X	X	X	X
ADRES010.mat	Winter		X	X	X			X	X		X		X	X	X	X	X
ADRES011.mat	Sommer	X	X	X		X	X	X		X	X		X	X	X	X	X
ADRES011.mat	Winter	X	X	X		X	X	X		X	X		X	X	X	X	X
ADRES012.mat	Winter	X	X					X					X	X	X	X	X
ADRES012.mat																	
ADRES013.mat	Sommer		X	X		X	X	X		X		X	X	X	X	X	X
ADRES013.mat	Winter		X	X		X		X		X		X	X	X	X	X	X
ADRES014.mat	Sommer	X	X	X		X	X	X					X	X	X	X	X
ADRES014.mat	Winter	X	X	X		X		X					X	X	X	X	X
ADRES015.mat	Sommer	X	X	X			X	X			X	X	X	X	X	X	X
ADRES015.mat	Winter	X	X	X			X	X			X	X	X	X	X	X	X
ADRES016.mat	Sommer		X			X		X		X			X	X	X	X	X
ADRES016.mat	Winter		X			X		X					X	X	X	X	X
ADRES017.mat	Sommer		X	X	X		X	X			X		X	X	X	X	X
ADRES017.mat	Winter		X	X	X		X	X			X		X	X	X	X	X
ADRES018.mat	Sommer	X	X	X	X	X	X	X	X			X	X	X	X	X	X
ADRES018.mat	Winter	X	X	X	X	X	X	X				X	X	X	X	X	X
ADRES019.mat	Sommer		X	X	X	X	X	X	X	X	X		X	X	X	X	X
ADRES019.mat	Winter		X	X	X	X	X	X	X	X	X		X	X	X	X	X

Fortsetzung auf nächster Seite

A. Übersicht über die vermessenen Geräte

<i>Fortsetzung der letzten Seite</i>																	
filename	Messung	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADRES020.mat	Sommer	X		X	X	X	X	X					X	X	X		
ADRES020.mat	Winter	X	X	X	X	X	X	X			X		X	X	X		
ADRES021.mat	Sommer		X	X		X	X	X	X		X		X	X	X	X	X
ADRES021.mat	Winter		X	X		X	X	X	X		X		X	X	X	X	X
ADRES022.mat	Sommer		X			X		X				X	X	X	X	X	X
ADRES022.mat	Winter		X			X		X	X			X	X	X	X	X	X
ADRES023.mat	Winter	X	X			X	X	X				X	X	X	X	X	
ADRES023.mat																	
ADRES024.mat	Sommer	X	X	X	X	X		X					X	X	X	X	X
ADRES024.mat	Winter		X	X	X	X		X					X	X	X	X	X
ADRES026.mat	Sommer	X	X	X			X	X					X	X	X	X	X
ADRES026.mat	Winter	X	X	X			X	X					X	X	X	X	X
ADRES027.mat	Sommer	X	X	X	X	X	X						X	X	X	X	X
ADRES027.mat	Winter	X	X	X	X	X	X						X	X	X	X	X
ADRES028.mat	Sommer	X	X	X	X		X	X	X				X	X	X	X	X
ADRES028.mat	Winter	X	X	X	X	X		X	X		X		X	X	X	X	X
ADRES029.mat	Sommer	X	X	X	X	X	X	X					X	X	X	X	X
ADRES029.mat	Winter	X	X	X	X	X		X					X	X	X	X	X
ADRES030.mat	Sommer	X		X		X	X	X			X	X	X	X	X	X	X
ADRES030.mat	Winter	X		X		X	X	X			X	X	X	X	X	X	X
ADRES031.mat	Sommer		X	X			X	X			X	X	X	X	X	X	X
ADRES031.mat	Winter		X	X				X			X	X	X	X	X	X	X
ADRES032.mat	Sommer	X		X			X	X				X	X	X	X	X	
ADRES032.mat	Winter	X	X	X				X				X	X	X	X		X
ADRES033.mat	Sommer		X					X			X		X	X	X	X	X
ADRES033.mat	Winter		X	X				X			X		X	X	X	X	X
ADRES034.mat	Sommer	X		X			X	X				X	X	X	X	X	X
ADRES034.mat	Winter	X	X	X			X					X	X	X	X	X	X
ADRES035.mat	Sommer	X	X			X	X	X				X	X	X	X	X	X
ADRES035.mat	Winter	X	X			X	X	X				X	X	X	X	X	X
ADRES036.mat	Sommer	X		X		X	X	X					X	X	X	X	X
ADRES036.mat	Winter	X	X	X		X		X					X	X	X	X	X
ADRES037.mat	Sommer	X	X	X		X		X			X		X	X	X	X	X
ADRES037.mat	Winter	X	X	X		X		X			X		X	X	X	X	X
ADRES038.mat	Sommer	X		X		X	X	X			X		X	X	X		
ADRES038.mat	Winter	X		X		X	X	X			X		X	X	X		
ADRES039.mat	Winter		X	X			X	X			X	X	X	X	X	X	X
ADRES039.mat																	
ADRES040.mat	Winter		X	X		X	X	X				X	X	X	X	X	X

*Fortsetzung auf nächster Seite*

A. Übersicht über die vermessenen Geräte

<i>Fortsetzung der letzten Seite</i>																	
filename	Messung	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADRES040.mat																	
ADRES041.mat	Winter	X		X		X	X	X			X		X	X	X	X	X
ADRES041.mat																	
ADRES042.mat	Sommer	X		X			X	X			X		X	X	X	X	X
ADRES042.mat																	
ADRES043.mat	Sommer		X	X			X	X			X		X	X	X	X	X
ADRES043.mat																	
Summe		46	64	65	31	45	47	70	15	17	39	32	75	75	75	70	68

Tabelle A.1.: Geräteübersicht der ADRES Messreihe

## B. MATLAB Quellcode

### B.1. build\_infostructure()

```
1 function ret = build_infostructure()
2 tic
3     % Funktion generiert die Infostruktur für die zukünftige Verwendung
4     % mit dem GUI und speichert eine Übersicht in Excel (falls aktiviert)
5
6     % In der Infostruktur werden je nach Einstellung auch die
7     % Geräteeinsatzdaten gespeichert sowie Summenwerte ermittelt
8
9     % config:
10    %Suche nach Bereichen, in denen das jeweilige Gerät in Betrieb war
11    %sehr zeitintensiv
12    find_areas = true;
13    %Übersicht über gefundene Geräte nach Excel exportieren
14    export_excel = false;
15
16    global geraete;
17    % "alte" Data-Structure laden
18    load('Data_Structure.mat');
19    % Gerätenamen laden
20    load('ger.mat');
21    % Haushaltsnamen ermitteln
22    field_names = fields(Data_Structure);
23    % return variable initialisieren
24    ret = cell(numel(field_names),1);
25    % Geräteübersichtsvariable initialisieren
26    geraete = cell(numel(field_names)*2,numel(ger)); %#ok<*USENS>
27    geraeteuebersicht = cell(numel(field_names)*2,numel(ger));
28    for i=1:numel(ger)
29        geraete{1,i} = ger{i};
30        geraeteuebersicht{1,i} = ger{i};
31    end
32
33    geraete{1,2} = 'Messung';
34    geraeteuebersicht{1,2} = 'Messung';
35
36    % alle Haushalte durchgehen
37    for i=1:numel(field_names)
38
39        infos = Data_Structure.(field_names{i});
40        % Dateiname der Rohdaten
```

## B. MATLAB Quellcode

```
41     ret{i}.filename = infos.Data_Filename;
42     geraete{i*2,1} = infos.Data_Filename;
43     geraete{i*2+1,1} = infos.Data_Filename;
44     geraeteuebersicht{i*2,1} = infos.Data_Filename;
45     geraeteuebersicht{i*2+1,1} = infos.Data_Filename;
46
47     infos = rmfield(infos, 'Data_Filename');
48     % Nur Detailmessungen von Interesse
49     if (isfield(infos, 'Sommer_Topas'))
50         infos = rmfield(infos, 'Sommer_Topas');
51     end
52     if (isfield(infos, 'Winter_Topas'))
53         infos = rmfield(infos, 'Winter_Topas');
54     end
55
56     messungen = fields(infos);
57     for g=1:numel(messungen)
58         disp(strcat('      ',messungen{g}, 'messung Detail Haushalt',ret{
59             i}.filename, ' :'));
60         geraete{i*2+g-1,2} = messungen{g};
61         geraeteuebersicht{i*2+g-1,2} = messungen{g};
62         if isfield(infos,messungen{g})
63             details = infos.(messungen{g});
64             % die Zeitfelder sichern und dann aus der Struktur
65             % entfernen:
66             ret{i}.start_time(g) = details.Time_Start;
67             ret{i}.end_time(g) = details.Time_End;
68             details = rmfield(details, {'Time_Start','Time_End'});
69             % Die Überschriften der einzelnen Messspalten bereinigen
70             % und standardisieren
71             days = fields(details);
72             det = details.(days{1});
73             % alle Tage durchgehen und index_data_array sichern
74             varname = strcat('day_index_',messungen{g});
75             ret{i}.(varname) = [];
76             for k=1:numel(days)
77                 ret{i}.(varname) = [ret{i}.(varname); details.(days{k}
78                     ).Index_Data_Array];
79             end
80             found = zeros(1,40);
81             % Jede Überschrift einzeln analysieren
82             geraete_temp = zeros(1,numel(det.Names_Datafields));
83             for k = 1:numel(det.Names_Datafields)
84                 temp = det.Names_Datafields{k};
85                 name = regexp(temp, '_', 'start');
86                 ret{i}.columns{g,k}.messung_name = messungen{g};
87                 ret{i}.columns{g,k}.fullname = temp;
88                 if (numel(name) > 2);
89                     ret{i}.columns{g,k}.name = temp(name(1)+1:name(end
90                         -1)-1);
91                 elseif (numel(name) == 2)
92                     ret{i}.columns{g,k}.name = temp(1:name(1)-1);
```

## B. MATLAB Quellcode

```
89     end
90     ret{i}.columns{g,k}.phase = regexp(temp, '[L]\d', 'match
91     ', 'once');
92     ret{i}.columns{g,k}.messung = temp(name(end)+1:end);
93
94     %mit bekannten Gerätetypen vergleichen
95     if (strfind(lower(det.Names_Datafields{k}), 'energie')
96         ≠ 0)
97         geraete_type = 40;
98     elseif (find_kuehlschrank(det.Names_Datafields{k}) ==
99         1)
100         geraete_type = 3;
101     elseif (find_waschmaschine(det.Names_Datafields{k}) ==
102         1)
103         geraete_type = 4;
104     elseif (find_gefrierschrank(det.Names_Datafields{k})
105         == 1)
106         geraete_type = 5;
107     elseif (find_trockner(det.Names_Datafields{k}) == 1)
108         geraete_type = 6;
109     elseif (find_bueegeleisen(det.Names_Datafields{k}) ==
110         1)
111         geraete_type = 7;
112     elseif (find_foehn(det.Names_Datafields{k}) == 1)
113         geraete_type = 8;
114     elseif (find_multimedia(det.Names_Datafields{k}) == 1)
115         geraete_type = 9;
116     elseif (find_geschirrspueler(det.Names_Datafields{k})
117         == 1)
118         geraete_type = 10;
119     elseif (find_toaster(det.Names_Datafields{k}) == 1)
120         geraete_type = 11;
121     elseif (find_mikrowelle(det.Names_Datafields{k}) == 1)
122         geraete_type = 12;
123     elseif (find_wasserkocher(det.Names_Datafields{k}) ==
124         1)
125         geraete_type = 13;
126     elseif (find_gesamt_L1(det.Names_Datafields{k}) == 1)
127         geraete_type = 16;
128     elseif (find_gesamt_L2(det.Names_Datafields{k}) == 1)
129         geraete_type = 17;
130     elseif (find_gesamt_L3(det.Names_Datafields{k}) == 1)
131         geraete_type = 18;
132     elseif (find_gesamt(det.Names_Datafields{k}) == 1)
133         geraete_type = 15;
134     else
135         % Sonstiges: Klasse 14
136         geraete_type = 14;
137     end %if Suchfunktion nach bekannten Gerätetypen
138     % temporäre Variable für die Funktion
139     % find_relevant_areas()
140     geraete_temp(1,k) = geraete_type;
```

## B. MATLAB Quellcode

```
133 % Geräte Typ
134 ret{i}.columns{g,k}.geraete_type = geraete_type;
135 % Geräteklasse
136 ret{i}.columns{g,k}.geraete_klasse = ger{geraete_type
    };
137 % Anzahl der gefundenen Geräte dieser Klasse
138 found(geraete_type) = found(geraete_type) + 1;
139 ret{i}.columns{g,k}.geraete_count = found(geraete_type
    );
140 % Spaltenindex innerhalb der Rohdaten
141 geraete{i*2+g-1,geraete_type}(found(geraete_type)).
    column_index = k;
142 % Phaseninformation
143 geraete{i*2+g-1,geraete_type}(found(geraete_type)).
    phase = regexp(temp, '[L]\d', 'match', 'once');
144 % User-freundliche Übersicht
145 geraeteuebersicht{i*2+g-1,geraete_type} = 'X';
146 end
147
148 % Suche nach relevanten Zeitpunkten
149 if (find_areas == true)
150     % Existiert schon eine Suche und deren Ergebnisse?
151     if (exist(strcat('cycle_infos_',ret{i}.filename,'_',
        messungen{g},'.mat'),'file') ≠ 0)
152         load(strcat('cycle_infos_',ret{i}.filename,'_',
            messungen{g},'.mat'));
153         if (isfield(cycle_infos,'start_time') == 0)
154             disp('Startzeiten ergänzen');
155             cycle_infos = extend_infostructure(ret{i}.
                filename,messungen{g},false,cycle_infos);
156         end
157     else
158         disp('Detailinfos ermitteln');
159         cycle_infos = find_relevant_areas(ret{i}.filename
            ,0,ret{i}.(varname),1:k,geraete_temp,false);
160         save(strcat('cycle_infos_',ret{i}.filename,'_',
            messungen{g},'.mat'),'cycle_infos');
161     end % if (exists)
162     for k=1:numel(det.Names_Datafields)
163         ret{i}.columns{g,k}.cycle_infos = cycle_infos(:,k)
            ;
164         geraete_type = ret{i}.columns{g,k}.geraete_type;
165         c = ret{i}.columns{g,k}.geraete_count;
166         geraete{i*2+g-1,geraete_type}(c).cycle_infos =
            cycle_infos(:,k);
167         %disp (k);
168         data = calculate_overall_data(cycle_infos(:,k));
169         pairs = [fieldnames(ret{i}.columns{g,k}),
            struct2cell(ret{i}.columns{g,k}); fieldnames(
                data), struct2cell(data)].';
170         ret{i}.columns{g,k} = struct(pairs{:});
171         a = fieldnames(data);
```

## B. MATLAB Quellcode

```
172         for n = 1:numel(a)
173             geraete{i*2+g-1,geraete_type}(c).(a{n}) = data
                .(a{n});
174         end %for n = 1:numel(a)
175     end %for k=1:numel(det.Names_Datafields)
176     end %if (find_areas == true)
177     end %if isfield(infos,messungen{g})
178     end %for g=1:numel(messungen)
179 end %for i=1:numel(field_names)
180
181 if (find_areas == true)
182     save('geraeteuebersicht2.mat','geraete','geraeteuebersicht');
183 else
184     save('geraeteuebersicht.mat','geraete','geraeteuebersicht');
185 end
186
187 if (export_excel == true)
188     % Initialisation of POI Libs
189     % Add Java POI Libs to matlab javapath
190     javaaddpath('poi_library/poi-3.8-20120326.jar');
191     javaaddpath('poi_library/poi-ooxml-3.8-20120326.jar');
192     javaaddpath('poi_library/poi-ooxml-schemas-3.8-20120326.jar');
193     javaaddpath('poi_library/xmlbeans-2.3.0.jar');
194     javaaddpath('poi_library/dom4j-1.6.1.jar');
195     javaaddpath('poi_library/stax-api-1.0.1.jar');
196     % Ergebnisfiles schreiben:
197     xlwrite('Geräteübersicht.xlsx',geraeteuebersicht,'Übersicht');
198 end
199 toc
200 end %function
```

### B.2. find\_relevant\_areas(...)

```
1 function ret = find_relevant_areas(filename,Data,day_index,column_index,
    geraete_type,do_plots)
2     close all
3     %Falls die Rohdaten nicht übergeben wurden, lade diese nachträglich
4     if (iscell(Data) == 0)
5         load(filename);
6     end;
7
8     % Rückgabeveriable initialisieren
9     %ret = struct(numel(day_index),numel(column_index));
10
11     % alle Tage(day_index) sowie Spalten(column_index) durchgehen und nach
12     % interessanten Daten suchen
13     for j=1:numel(day_index)
14         for k=1:numel(column_index)
15             % gibt an, ob gerade nach Start oder Ende gesucht wird
16             start = 0;
```

## B. MATLAB Quellcode

```
17
18 % counter für gefundene Zyklen
19 found = 0;
20
21 % Variablen für einfacheren Zugriff umbenennen
22 data = Data{day_index(j)}.data(:,column_index(k)); %#ok<*USENS
    >
23 time = Data{day_index(j)}.time;
24
25 % Setze diverse Infos in der Informationsstruktur
26 ret(j,k).filename = filename;
27 ret(j,k).day_index = day_index(j);
28 ret(j,k).column_index = column_index(k);
29 ret(j,k).geraete_type = geraete_type(k);%geraete_type(j,k);
30 ret(j,k).phase = regexp(Data{day_index(j)}.names{
    column_index(k)}, '[L]\d', 'match', 'once');
31 ret(j,k).power_max = max(data);
32 ret(j,k).power_mean = mean(data);
33 ret(j,k).on_time = 0;
34
35 % alle Zeitpunkte durchgehen und nach Ein-, bzw Ausschalten
    suchen
36 % falls am Anfang Ein ist (Bedingung: Leistung größer 5W)
37 if (data(1) ≥ 5)
38     start = 1;
39     found = found + 1;
40     ret(j,k).start(found) = 1;
41     ret(j,k).start_time(found) = time(1);
42 end
43 i = 2;
44 while (i<numel(data))
45     % Einschaltzeitpunkt gefunden, Ausschaltzeitpunkt suchen
46     if (start == 1)
47         % Kriterium für Ausschalten (Leistungsrückgang auf
            40%)
48         % -> bei Programmen von Waschmaschinen,
            Geschirrspülern
49         % und Trocknern wird ein eigenes Kriterium definiert.
50         % Ebenso bei alternierender Leistungsaufnahme bei
51         % temperaturgesteuerten Geräten wie Toaster und
52         % Bügeleisen.
53         switch (geraete_type(k))
54             % Waschmaschine, Trockner
55             case {4,6}
56                 if (i+130 < numel(data))
57                     if (sum(data(i:i+130)/130) ≤ 4)
58                         on_time = time(i-1) - time(ret(j,k).
                            start(found));
59                         %Länger als eine halbe Stunde in
60                         %Betrieb
61                         if (on_time ≥ 1/48)
62                             start = 0;
```

## B. MATLAB Quellcode

```
63         ret(j,k).ende(found) = i-1;
64         ret(j,k).end_time(found) = time(i
65         -1);
65         ret(j,k).on_times(found) = time(i
66         -1) - time(ret(j,k).start(found)
67         ));
66         ret(j,k).on_time = ret(j,k).
68         on_time + time(i-1)-time(ret(j,
69         k).start(found));
67         else
68         start = 0;
69         found = found - 1;
70         end
71     end
72 else
73     if (sum(data(i:end)/200) ≤ 4)
74         on_time = time(i-1) - time(ret(j,k).
75         start(found));
76         if (on_time ≥ 1/48)
77             start = 0;
78             ret(j,k).ende(found) = i-1;
79             ret(j,k).end_time(found) = time(i
80             -1);
81             ret(j,k).on_times(found) = time(i
82             -1) - time(ret(j,k).start(found)
83             ));
84             ret(j,k).on_time = ret(j,k).
85             on_time + time(i-1)-time(ret(j,
86             k).start(found));
87         else
88             start = 0;
89             found = found - 1;
90         end
91     end
92     end
93     % Geschirrspüler
94     case {10}
95         % Suche beginnend 3 Stunden nach dem
96         % Einschalten nach dem Ausschaltzeitpunkt des
97         % Programms
98         i = i + 10800; % + 3h
99         if (i > numel(data))
100             ret(j,k).end_time(found) = time(end);
101             ret(j,k).ende(found) = numel(data); %#ok<*
102             AGROW>
103         else
104             for g=i:-1:i-10800
105                 if (data(g) ≤ (data(g-1)*0.4)-3)
106                     on_time = time(g) - time(ret(j,k).
107                     start(found));
108                     %Dauer größer 1h
109                     if (on_time ≥ 1/24)
```

## B. MATLAB Quellcode

```
102         start = 0;
103         g = g-1;
104         ret(j,k).end_time(found) =
            time(g);
105         ret(j,k).ende(found) = g;
106         ret(j,k).on_times(found) =
            time(g) - time(ret(j,k).
            start(found));
107         ret(j,k).on_time = ret(j,k).
            on_time + time(g)-time(ret(
            j,k).start(found));
108     else
109         start = 0;
110         found = found - 1;
111     end
112     break;
113 end
114 end
115 end
116 % Bügeleisen, Toaster, Mikrowelle (Test)
117 case {7,11,12}
118     % Suche beginnend 1/2 Stunde nach dem
119     % Einschalten nach dem Ausschaltzeitpunkt des
120     % Programms
121     i = i + 1800; % + 1/2 h
122     if (i > numel(data))
123         ret(j,k).end_time(found) = time(end);
124         ret(j,k).ende(found) = numel(data);
125     else
126         for g=i:-1:i-1800
127             if (data(g) ≤ (data(g-1)*0.4)-3)
128                 start = 0;
129                 g = g-1;
130                 ret(j,k).ende(found) = g;
131                 ret(j,k).end_time(found) = time(g)
132                 ;
133                 ret(j,k).on_times(found) = time(g)
134                 - time(ret(j,k).start(found));
135                 ret(j,k).on_time = ret(j,k).
136                 on_time + time(g)-time(ret(j,k)
137                 .start(found));
138                 break;
139             end
140         end
141     end
142 % Standard - Suchalgorithmus
143 otherwise
144     if (data(i) ≤ (data(i-1)*0.4))
145         start = 0;
146         ret(j,k).ende(found) = i-1;
147         ret(j,k).end_time(found) = time(i-1);
```

## B. MATLAB Quellcode

```
144         ret(j,k).on_times(found) = time(i-1) -
145             time(ret(j,k).start(found));
146         ret(j,k).on_time = ret(j,k).on_time + time
147             (i-1)-time(ret(j,k).start(found));
148     end
149     % Einschaltzeitpunkt suchen, Kriterium für Einschalten:
150     % Leistungssteigerung um Faktor 1.1 + 2W
151     elseif (data(i) ≥ (data(i-1)*1.1 + 2))
152         start = 1;
153         found = found + 1;
154         ret(j,k).start(found) = i;
155         ret(j,k).start_time(found) = time(i);
156     end
157     i = i + 1;
158 end
159 % falls Ende am nächsten Tag liegt -> Ende = letztes Element
160 if (start == 1)
161     ret(j,k).ende(found) = numel(data);
162     ret(j,k).end_time(found) = time(end);
163     ret(j,k).on_times(found) = time(end) - time(ret(j,k).start
164         (found));
165 end
166 % durchschnittliche EIN-Leistung ermitteln
167 if (found > 0)
168     if (ret(j,k).start(1) == 1)
169         on_power = data(1:ret(j,k).ende(1));
170     else
171         on_power = data(ret(j,k).start(1) : ret(j,k).ende(1));
172     end
173     for i=2:numel(ret(j,k).ende)
174         on_power = [on_power; data(ret(j,k).start(i) : ret(j,k)
175             .ende(i))];
176     end
177     on_power = 0;
178 end
179 %maximale Leistung
180 ret(j,k).on_power_max = max(on_power);
181 %durchschnittliche EIN-Leistung
182 ret(j,k).on_power_mean = mean(on_power);
183 % minimale EIN-Leistung
184 ret(j,k).on_power_min = min(on_power);
185
186 %Einschaltzyklen
187 ret(j,k).cycles = found;
188
189 %Stand-By Zeit
190 ret(j,k).off_time = time(end) - time(1) - ret(j,k).on_time;
191
```

## B. MATLAB Quellcode

```
192 %Prozent der eingeschalteten Zeit
193 ret(j,k).percent_on = ret(j,k).on_time/(time(end) - time(1));
194
195 % Ein Zeit ermitteln
196 if (found > 0)
197     % durchschnittliche EIN-Zeit
198     ret(j,k).on_time_mean = mean(ret(j,k).on_times);
199     % maximale EIN-Zeit
200     ret(j,k).on_time_max = max(ret(j,k).on_times);
201     % minimale EIN-Zeit
202     ret(j,k).on_time_min = min(ret(j,k).on_times);
203 else
204     ret(j,k).on_time_mean = 0;
205     ret(j,k).on_time_max = 0;
206     ret(j,k).on_time_min = 0;
207 end
208 %Stand-By Leistung ermitteln
209 if (found > 0)
210     if (ret(j,k).start(1) ≠ 1)
211         standby = data(1:ret(j,k).start(1)-1);
212     else
213         standby = [];
214     end
215     for i=2:numel(ret(j,k).start)
216         standby = [standby; data(ret(j,k).ende(i-1)+1:ret(j,k)
217             .start(i)-1)];
218     end
219     standby = data;
220 end
221 % maximale Stand-By Leistung
222 ret(j,k).standby_max = max(standby);
223 % durchschnittliche Stand-By Leistung
224 ret(j,k).standby_mean = mean(standby);
225
226 %Plots der gefundenen Zyklen
227 if (do_plots == true)
228     figure;
229     hist(on_power);
230     figure;
231     hist(standby);
232     load('ger.mat');
233     for i=1:found
234         figure;
235         if (ret(j,k).ende(i)+10 > numel(data))
236             plot_data = data(ret(j,k).start(i)-10:ret(j,k).
237                 ende(i));
238             time_data = time(ret(j,k).start(i)-10:ret(j,k).
239                 ende(i));
240         else
241             plot_data = data(ret(j,k).start(i)-10:ret(j,k).
242                 ende(i)+10);
```

## B. MATLAB Quellcode

```
240         time_data = time(ret(j,k).start(i)-10:ret(j,k).
                ende(i)+10);
241     end
242     plot(time_data,plot_data);
243     title(strcat(ger{ret(j,k).geraete_type},' vom ',
                datestr(time(1),'dd.mm.yyyy')));
244     xlabel('Zeit');
245     ylabel('Leistung in W');
246     end
247     end
248     end
249     end
```

### B.3. show\_usage\_histogram(...)

```
1 % Funktion zur Auswertung der Einschaltzeitpunkte des übergebenen Gerätes
2 % in den übergebenen Haushalten bei wahlweise Sommer-, Wintermessung oder
3 % beiden Messreihen. Zusätzlich muss eine Auflösung für das Diagramm
4 % übergeben werden, wobei 1 für 1 Tag steht also z.B. 1/24 für
   stundenweise
5 % Auflösung übergeben werden muss. Mit der boolean-Variable do_plots kann
6 % die Ausgabe der Grafiken unterdrückt werden.
7 function ret = show_usage_histogram(geraete, hh,sommer,winter,geraete_type
   ,weekdays,aufloesung,do_plots)
8     %falls Gerätedaten nicht übergeben wurden, lade sie nach
9     if (iscell(geraete) == 0)
10        load('geraeteuebersicht2.mat');
11    end
12    load('ger.mat');
13    %Init
14    start_times = [];
15    num_device = 0;
16    num_days = 0;
17    %Alle Zeilen durchgehen
18    for i=2:size(geraete,1)
19        %Falls in Liste gewünschter Haushalte enthalten
20        if any(strcmp(geraete{i,1},hh))
21            %Sommer oder Wintermessung
22            if (sommer == true && strcmp(geraete{i,2},'Sommer') || winter
                == true && strcmp(geraete{i,2},'Winter'))
23                %Falls Messdaten vorhanden
24                if (isstruct(geraete{i,geraete_type}) &= 0)
25                    for k=1:numel(geraete{i,geraete_type})
26                        num_device = num_device + 1;
27                        cycles = geraete{i,geraete_type}(k).cycle_infos;
28                        %alle Tage durchgehen und nach gewünschten
                            Wochentagen
29                        %filtern
30                        for j=1:numel(cycles)
31                            if (cycles(j).cycles &= 0)
```

## B. MATLAB Quellcode

```
32         if any(strcmp(weekdays,datestr(cycles(j).
33             start_time(1),'ddd')))
34             %Nur Uhrzeit des Einschaltzeitpunktes
35             werden betrachtet
36             %(Nachkommastellen des Zeitvektors)
37             num_days = num_days + 1;
38             if (isfield(cycles,'exclude'))
39                 if (cycles(j).exclude ≠ 1)
40                     start_times = [start_times [
41                         cycles(j).start_time] -
42                         datenum(datestr([cycles(j).
43                             start_time],1))']; %#ok<*
44                             AGROW>
45                     end
46                 else
47                     start_times = [start_times [cycles
48                         (j).start_time] - datenum(
49                             datestr([cycles(j).start_time
50                                 ],1))'];
51                     end %if not exclude
52                     end % if weekday
53                     end % if(cyclex(j).cycles) ≠ 0)
54                     end % for j=1:numel(cycles)
55                     end %for k=1:numel(geraete{i,geraete_type})
56                     end %if (isstruct(geraete{i,geraete_type}) ≠ 0)
57                     end %if sommer || winter
58                 end % if hh
59             end % for i=2:size(geraete,1)
60
61 %Ergebnisse vorhanden?
62 if(numel(start_times) ≠ 0)
63     %Wahrscheinlichkeitsdichtefunktion schätzen und plotten
64     [ret.f,ret.xi] = ksdensity(start_times,'npoints',8640);
65     ret.pd = fitdist(start_times,'kernel');
66
67     % Rückgabewerte setzen
68     ret.analysis = 'show_usage_histogram';
69     ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
70     ret.geraete_type = geraete_type; ret.weekdays = weekdays;
71     ret.aufloesung = aufloesung; ret.start_times = start_times;
72     [ret.nelements,ret.centers] = hist(start_times,aufloesung/2:
73         aufloesung:1+aufloesung/2);
74     n = length(start_times);
75
76     if (do_plots == true)
77         % Plot erstellen
78         ret.figHandles(1) = figure;
79         hist(start_times,aufloesung/2:aufloesung:1+aufloesung/2);
80         h = title({'Einschalthäufigkeit %s %d Haushalte: (%d
81             Zyklen)',ger{geraete_type},numel(hh),numel(start_times)};
82             sprintf('%d Geräte, %d Tagesmessungen',num_device,num_days)
83             });
```

## B. MATLAB Quellcode

```
71     axpos = get(gca, 'pos');
72     set(h, 'units', 'normalized');
73     extent = get(h, 'extent');
74     set(gca, 'pos', [axpos(1) axpos(2) axpos(3) axpos(4)-0.2*extent
75                     (4)])
76     xlabel = '';
77     for i=0:24
78         xlabel = strcat(xlabel, sprintf('%02d:00', i), '|');
79     end
80     xlabel = xlabel(1:end-1);
81     set(gca, 'xlim', [0 1]);
82     set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
83     rotateticklabel(gca, 90);
84     switch (aufloesung)
85     case 1/24
86         aufl = '1 Stunde';
87     case 1/48
88         aufl = '1/2 Stunde';
89     case 1/96
90         aufl = '1/4 Stunde';
91     case 1/(24*6)
92         aufl = '10 Minuten';
93     case 1/(24*12)
94         aufl = '5 Minuten';
95     case 1/(24*60)
96         aufl = '1 Minute';
97     otherwise
98         aufl = '';
99     end
100    xlabel(strcat('Einschaltzeitpunkt (hh:mm), Auflösung ', {' '},
101                aufl));
102    ylabel('Anzahl Zyklen');
103    moveLabel('x', 35, ret.figHandles(1), gca);
104    grid on;
105
106    % normierten Plot erstellen
107    ret.figHandles(2) = figure;
108    prob = ret.nelements / (n*aufloesung);
109    bar(ret.centers, prob, 'hist');
110    hold on;
111    plot(ret.xi, ret.f, 'r', 'lineWidth', 3);
112    h = title({sprintf('Einschaltheufigkeit %s %d Haushalte: (%d
113                Zyklen)', ger{geraete_type}, numel(hh), numel(start_times));
114              sprintf('%d Geräte, %d Tagesmessungen', num_device, num_days)
115                });
116    axpos = get(gca, 'pos');
117    set(h, 'units', 'normalized');
118    extent = get(h, 'extent');
119    set(gca, 'pos', [axpos(1) axpos(2) axpos(3) axpos(4)-0.2*extent
120                    (4)])
121    set(gca, 'xlim', [0 1]);
122    set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
```

## B. MATLAB Quellcode

```
117         rotateticklabel(gca,90);
118         xlabel(strcat('Einschaltzeitpunkt (hh:mm), Auflösung ',{' '},
119                     aufl));
119         ylabel('Anzahl Zyklen');
120         moveLabel('x',35,ret.figHandles(2),gca);
121         grid on;
122     end %if (do_plots == true)
123 else
124     ret = false;
125 end %if (numel(start_times) ≠ 0)
126 end %function
```

### B.4. show\_usage\_duration(...)

```
1 % Funktion zur Auswertung der Einschaltdauer des übergebenen Gerätes
2 % in den übergebenen Haushalten bei wahlweise Sommer-, Wintermessung oder
3 % beiden Messreihen. Zusätzlich muss eine Auflösung für das Diagramm
4 % übergeben werden, wobei 1 für 1 Tag steht also z.B. 1/24 für
5 % stundenweise
6 % Auflösung übergeben werden muss. zusätzlich kann der Bereich des ersten
7 % Plots mit den Parametern xmin und xmax eingestellt werden, wobei hier 0
8 % für Mitternacht und 1 für 24:00 steht, also wieder auf einen ganzen Tag
9 % normiert wurde
10 function ret = show_usage_duration(geraete, hh,sommer,winter,geraete_type,
11     weekdays,aufloesung,xmin,xmax)
12     %falls Gerätedaten nicht übergeben wurden, lade sie nach
13     if (iscell(geraete) == 0)
14         load('geraeteuebersicht2.mat');
15     end
16     load('ger.mat');
17     %Init
18     on_times = [];
19     num_device = 0;
20     num_days = 0;
21     %Alle Zeilen durchgehen
22     for i=2:size(geraete,1)
23         %Falls in Liste gewünschter Haushalte enthalten
24         if any(strcmp(geraete{i,1},hh))
25             %Sommer oder Wintermessung
26             if (sommer == true && strcmp(geraete{i,2},'Sommer') || winter
27                 == true && strcmp(geraete{i,2},'Winter'))
28                 %Falls Messdaten vorhanden
29                 if (isstruct(geraete{i,geraete_type}) ≠ 0)
30                     for k=1:numel(geraete{i,geraete_type})
31                         num_device = num_device + 1;
32                         cycles = geraete{i,geraete_type}(k).cycle_infos;
33                         %alle Tage durchgehen und nach gewünschten
34                         Wochentagen
35                         %filtern
36                         for j=1:numel(cycles)
```

## B. MATLAB Quellcode

```
33         if (cycles(j).cycles ≠ 0)
34             if any(strcmp(weekdays,datestr(cycles(j).
35                 start_time(1),'ddd')))
36                 num_days = num_days + 1;
37                 if (isfield(cycles,'exclude'))
38                     if (cycles(j).exclude ≠ 1)
39                         on_times = [on_times cycles(j)
40                             .on_times]; %#ok<*AGROW>
41                     end
42                 else
43                     on_times = [on_times cycles(j).
44                         on_times];
45                 end %if not exclude
46             end %if weekday
47         end %if (cycles(j))
48     end %for j=1:numel(cycles)
49     end %for k=1:numel(geraete{i,geraete_type})
50     end %if (isstruct(geraete{i,geraete_type}))
51     end %if sommer || winter
52 end %if hh
53 end %for i=2:size(geraete,1)
54
55 %Ergebnisse vorhanden?
56 if (numel(on_times) ≠ 0)
57     %Rückgabewerte
58     ret.analysis = 'show_usage_duration';
59     ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
60     ret.geraete_type = geraete_type; ret.weekdays = weekdays;
61     ret.aufloesung = aufloesung; ret.on_times = on_times;
62     [ret.nelements,ret.centers] = hist(on_times,aufloesung/2:
63         aufloesung:1+aufloesung/2);
64     ret.num_days = num_days; ret.num_device = num_device;
65
66     %plot erstellen
67     ret.figHandles(1) = figure;
68     hist(on_times,aufloesung/2:aufloesung:1+aufloesung/2);
69     h = title({'sprintf('Einschaltdauer %s %d Haushalte: (%d Zyklen)',
70         ger{geraete_type},numel(hh),numel(on_times));sprintf('%d Geräte
71         , %d Tagesmessungen',num_device,num_days)});
72     axpos = get(gca,'pos');
73     set(h,'units','normalized');
74     extent = get(h,'extent');
75     set(gca,'pos',[axpos(1) axpos(2) axpos(3) axpos(4)-0.3*extent(4)])
76     %weniger als 3 Stunden anzeigen -> Achsbeschriftung im 10-Minuten
77     %Intervall
78     if (xmax ≤ 3/24+0.01)
79         xlabel = '';
80         for i=0:24
81             for j=0:5
82                 xlabel = strcat(xlabel,sprintf('%02d:%02d',i,j*10),'
83                     |');
84             end
85         end
86     end
87 end
```

## B. MATLAB Quellcode

```
78     end
79     xlabel = xlabel(1:end-1);
80     xticks = 0:1/(24*6):xmax;
81     ret.xticks = xticks;
82     set(gca, 'xtick',xticks, 'xticklabel', xlabel(1:(size(xticks,2))
      *6-1));
83     ret.xlabel = xlabel;
84 else
85     xlabel = '';
86     for i=0:24
87         xlabel = strcat(xlabel, sprintf('%02d:00',i), '|');
88     end
89     xlabel = xlabel(1:end-1);
90     set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
91 end
92 set(gca, 'xlim', [xmin xmax]);
93
94 rotateticklabel(gca, 90);
95 switch (aufloesung)
96     case 1/24
97         aufl = '1 Stunde';
98     case 1/48
99         aufl = '1/2 Stunde';
100    case 1/96
101        aufl = '1/4 Stunde';
102    case 1/(24*6)
103        aufl = '10 Minuten';
104    case 1/(24*12)
105        aufl = '5 Minuten';
106    case 1/(24*60)
107        aufl = '1 Minute';
108    otherwise
109        aufl = '??';
110 end
111 xlabel(strcat('Einschaltdauer (hh:mm), Auflösung ', {' '}, aufl));
112 ylabel('Anzahl Zyklen');
113 moveLabel('x', 35, ret.figHandles(1), gca);
114 grid on;
115 %maximale, minimale, und durchschnittliche Einschaltdauer anzeigen
116 fprintf('Maximale Einschaltdauer: %s\n', datestr(max(on_times), 'HH:
      MM:SS'));
117 ret.max_on_time = datestr(max(on_times), 'HH:MM:SS');
118 fprintf('Durchschnittliche Einschaltdauer: %s\n', datestr(mean(
      on_times), 'HH:MM:SS'));
119 ret.mean_on_time = datestr(mean(on_times), 'HH:MM:SS');
120
121 %zweiten Plot mit begrenzter Einschaltdauer generieren
122 %Begrenzung: 6 Stunden
123 ret.figHandles(2) = figure;
124 [N, ~] = histc(on_times, 0:aufloesung:1/4);
125 bar(0:aufloesung:1/4, N, 'histc');
```

## B. MATLAB Quellcode

```
126     h = title({sprintf('Einschaltdauer %s %d Haushalte: (%d Zyklen)',
127                   ger{geraete_type}, numel(hh), numel(on_times)); sprintf('%d Geräte
128                   , %d Tagesmessungen', num_device, num_days)});
129     xlabel(strcat('Einschaltdauer (hh:mm), Auflösung ', {' '}, aufl));
130     ylabel('Anzahl Zyklen');
131     set(gca, 'xlim', [0 1/4]);
132     xlabels = '';
133     for i=0:24
134         xlabels = strcat(xlabels, sprintf('%02d:00', i), '|');
135     end
136     xlabels = xlabels(1:end-1);
137     set(gca, 'xtick', 0:1/24:1/4, 'xticklabel', xlabels);
138     grid on;
139     on_times2 = on_times(on_times < 1/4);
140     fprintf('Maximale Einschaltdauer: %s\n', datestr(max(on_times2), 'HH
141             :MM:SS'));
142     ret.max_on_time2 = datestr(max(on_times2), 'HH:MM:SS');
143     fprintf('Durchschnittliche Einschaltdauer: %s\n', datestr(mean(
144             on_times2), 'HH:MM:SS'));
145     ret.mean_on_time2 = datestr(mean(on_times2), 'HH:MM:SS');
146 else
147     ret = false;
148 end
149 end
```

### B.5. show\_usage\_count(...)

```
1 % Diese Funktion zeigt ein Histogramm der Einschalthäufigkeit pro Tag an,
2 % berücksichtigt nur die gewünschten Haushalte, Messungen und Wochentage
3 function ret = show_usage_count(geraete, hh, sommer, winter, geraete_type,
4     weekdays, do_plots)
5     %falls Gerätedaten nicht übergeben wurden, lade sie nach
6     if (iscell(geraete) == 0)
7         load('geraeteuebersicht2.mat');
8     end
9     load('ger.mat');
10    %Init
11    on_times = [];
12    num_device = 0;
13    num_days = 0;
14    %Alle Zeilen durchgehen
15    for i=2:size(geraete,1)
16        %Falls in Liste gewünschter Haushalte enthalten
17        if any(strcmp(geraete{i,1}, hh))
18            %Sommer oder Wintermessung
19            if (sommer == true && strcmp(geraete{i,2}, 'Sommer') || winter
20                == true && strcmp(geraete{i,2}, 'Winter'))
21                %Falls Messdaten vorhanden
22                if (isstruct(geraete{i, geraete_type}) &= 0)
23                    for k=1:numel(geraete{i, geraete_type})
```

## B. MATLAB Quellcode

```
22         num_device = num_device + 1;
23         cycles = geraete{i,geraete_type}(k).cycle_infos;
24         %alle Tage durchgehen und nach gewünschten
           Wochentagen
25         %filtern
26         for j=1:numel(cycles)
27             %if (cycles(j).cycles ≠ 0)
28                 if any(strcmp(weekdays,datestr(cycles(j).
           date,'ddd')))
29                     num_days = num_days + 1;
30                     if (isfield(cycles,'exclude'))
31                         if (cycles(j).exclude ≠ 1)
32                             on_times = [on_times cycles(j)
           .cycles];%#ok<*AGROW>
33                         end
34                     else
35                         on_times = [on_times cycles(j).
           cycles];
36                     end %if not exclude
37                 end %if weekday
38             %end %if (cycles(j))
39         end %for j=1:numel(cycles)
40     end %for k=1:numel(geraete{i,geraete_type})
41 end %if (isstruct(geraete{i,geraete_type}))
42 end %if sommer || winter
43 end %if hh
44 end %for i=2:size(geraete,1)
45
46 %Ergebnisse vorhanden?
47 if (numel(on_times) ≠ 0)
48     % Rückgabewerte setzen
49     ret.analysis = 'show_usage_count';
50     ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
51     ret.geraete_type = geraete_type; ret.weekdays = weekdays;
52     ret.on_times = on_times;
53     if (max(on_times) < 10)
54         X = 0:1:10;
55     else
56         X = 10;
57     end
58     [ret.nelements,ret.centers] = hist(on_times,X);
59
60     if (do_plots)
61         %plot erstellen
62         ret.figHandles(1) = figure;
63         if (max(on_times) < 10)
64             X = 0:1:10;
65             set(gca,'xlim',[-0.5 10]);
66             hold on;
67         else
68             X = 10;
69         end

```

## B. MATLAB Quellcode

```
70 hist(on_times,X);
71 h = title({sprintf('Einschaltzyklen pro Tag %s %d Haushalte',
72     ger{geraete_type}, numel(hh));
73     sprintf('%d Geräte, %d Tagesmessungen %d gewertet',
74     num_device,num_days,numel(on_times))});
75 axpos = get(gca,'pos');
76 set(h,'units','normalized');
77 extent = get(h,'extent');
78 set(gca,'pos',[axpos(1) axpos(2) axpos(3) axpos(4)-0.2*extent
79     (4)])
80 xlabel('Einschaltzyklen');
81 ylabel('Häufigkeit');
82 grid on;
83 %maximale, minimale, und durchschnittliche Einschaltzyklen
84 anzeigen
85 fprintf('Maximale Zyklen / Tag: %d\n',max(on_times));
86 fprintf('Durchschnittliche Zyklen / Tag: %f\n',mean(on_times))
87 ;
88
89 %Plot der null ausblendet;
90 ret.figHandles(2) = figure;
91 on_times = on_times(on_times > 0);
92 if (max(on_times) < 10)
93     X = 0:1:10;
94     set(gca,'xlim',[0 10]);
95     hold on;
96 else
97     X = 10;
98 end
99 hist(on_times,X);
100 h = title({sprintf('Einschaltzyklen pro Tag %s %d Haushalte',
101     ger{geraete_type}, numel(hh));
102     sprintf('%d Geräte, %d Tagesmessungen %d gewertet',
103     num_device,num_days,numel(on_times))});
104 axpos = get(gca,'pos');
105 set(h,'units','normalized');
106 extent = get(h,'extent');
107 set(gca,'pos',[axpos(1) axpos(2) axpos(3) axpos(4)-0.2*extent
108     (4)])
109 xlabel('Einschaltzyklen');
110 ylabel('Häufigkeit');
111 grid on;
112 %maximale, minimale, und durchschnittliche Einschaltzyklen
113 anzeigen
114 fprintf('Maximale Zyklen / Tag (ohne 0): %d\n',max(on_times));
115 fprintf('Durchschnittliche Zyklen / Tag (ohne 0): %f\n',mean(
116     on_times));
117
118 if (max(on_times) > 100)
119     % zweiter Plot mit Begrenzung der Zyklen auf 100
120     ret.figHandles(2) = figure;
121     [N,~] = histc(on_times,0:10:100);
```

## B. MATLAB Quellcode

```
112         bar(0:10:100,N,'histc');
113         title(sprintf('Einschaltzyklen pro Tag %s %d Haushalte: (%
114             d Tage)',ger{geraete_type},numel(hh),numel(on_times)));
115         xlabel('Einschaltzyklen');
116         ylabel('Häufigkeit');
117         grid on;
118         %maximale, minimale, und durchschnittliche Einschaltzyklen
119         anzeigen
120         on_times = on_times(on_times < 100);
121         fprintf('Maximale Zyklen / Tag: %d\n',max(on_times));
122         fprintf('Durchschnittliche Zyklen / Tag: %f\n',mean(
123             on_times));
124         ret.on_times2 = on_times;
125     else
126         ret.on_times2 = on_times;
127     end
128 end
129 end
```

### B.6. show\_power\_consumption(...)

```
1 %Wertet die Nennleistung aller Geräte an den gewünschten Wochentagen und
2 %Messreihen aus, stellt den Plot mit der gewünschten Auflösung dar und
3 %ermittelt entweder aus der maximalen oder der durchschnittlichen Leistung
4 %die gewünschte Auswertung
5 function ret = show_power_consumption(geraete, hh,sommer,winter,
6     geraete_type,weekdays,aufloesung,max_mean)
7     %falls Gerätedaten nicht übergeben wurden, lade sie nach
8     if (iscell(geraete) == 0)
9         load('geraeteuebersicht2.mat');
10    end
11    load('ger.mat');
12    %Init
13    power_max = [];
14    power_mean = [];
15    num_device = 0;
16    %Alle Zeilen durchgehen
17    for i=2:size(geraete,1)
18        %Falls in Liste gewünschter Haushalte enthalten
19        if any(strcmp(geraete{i,1},hh))
20            %Sommer oder Wintermessung
21            if (sommer == true && strcmp(geraete{i,2},'Sommer') || winter
22                == true && strcmp(geraete{i,2},'Winter'))
23                %Falls Messdaten vorhanden
24                if (isstruct(geraete{i,geraete_type}) &= 0)
25                    for k=1:numel(geraete{i,geraete_type})
26                        num_device = num_device + 1;
27                    end
28                end
29            end
30        end
31    end
32    ret.power_max = power_max;
33    ret.power_mean = power_mean;
34    ret.num_device = num_device;
```

## B. MATLAB Quellcode

```
25         cycles = geraete{i,geraete_type}(k).cycle_infos;
26         %alle Tage durchgehen und nach gewünschten
           Wochentagen
27         %filtern
28         for j=1:numel(cycles)
29             if (cycles(j).cycles ≠ 0)
30                 if any(strcmp(weekdays,datestr(cycles(j).
           start_time(1),'ddd')))
31                     power_max = [power_max cycles(j).
           on_power_max];
32                     power_mean = [power_mean cycles(j).
           on_power_mean];
33                 end
34             end
35         end
36     end
37 end
38 end
39
40 end
41 end
42
43 if (numel(power_max) ≠ 0)
44     % Rückgabewerte setzen
45     ret.analysis = 'show_power_consumption';
46     ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
47     ret.geraete_type = geraete_type; ret.weekdays = weekdays;
48     ret.aufloesung = aufloesung; ret.max_mean = max_mean;
49     ret.power_max = power_max;
50     ret.power_mean = power_mean;
51
52     ret.figHandles(1) = figure;
53     if (strcmp(max_mean,'max') == 1)
54         hist(power_max,aufloesung/2:aufloesung:max(power_max));
55         [ret.nelements,ret.centers] = hist(power_max,aufloesung/2:
           aufloesung:max(power_max));
56     elseif (strcmp(max_mean,'mean') == 1)
57         hist(power_mean,aufloesung/2:aufloesung:max(power_mean));
58         [ret.nelements,ret.centers] = hist(power_mean,aufloesung/2:
           aufloesung:max(power_mean));
59     else
60         fprintf('Bitte "max" oder "mean" übergeben');
61     end
62     h = title({sprintf('Nennleistung %s %d Haushalte',ger{geraete_type
           },numel(hh));sprintf('%d Geräte, %d Tagesmessungen',num_device,
           numel(power_max))});
63     axpos = get(gca,'pos');
64     set(h,'units','normalized');
65     extent = get(h,'extent');
66     set(gca,'pos',[axpos(1) axpos(2) axpos(3) axpos(4)-0.2*extent(4)])
67     xlabel(strcat('Nennleistung in W Auflösung:',{' '},sprintf('%d',
           aufloesung),'W'));
```

## B. MATLAB Quellcode

```
68     ylabel('Anzahl Geräte');
69     rotateticklabel(gca,90);
70     moveLabel('x',30,ret.figHandles(1),gca);
71     grid on;
72
73     fprintf('durchschnittliche Leistung (Max): %f\n',mean(power_max));
74     fprintf('maximale Leistung (Max): %f\n',max(power_max));
75     fprintf('minimale Leistung (Max): %f\n',min(power_max));
76     fprintf('durchschnittliche Leistung (Mean): %f\n',mean(power_mean)
77         );
78     fprintf('maximale Leistung (Mean): %f\n',max(power_mean));
79     fprintf('minimale Leistung (Mean): %f\n',min(power_mean));
80
81     ret.figHandles(2) = figure;
82     boxplot([ power_max; power_mean]','labels',{ 'Maximalleistung', '
83         Durchschnittsleistung'});
84     h = title({sprintf('Nennleistung %s %d Haushalte',ger{geraete_type
85         }, numel(hh));sprintf('%d Geräte, %d Tagesmessungen', num_device,
86         numel(power_max))});
87     axpos = get(gca, 'pos');
88     set(h, 'units', 'normalized');
89     extent = get(h, 'extent');
90     set(gca, 'pos', [axpos(1) axpos(2) axpos(3) axpos(4)-0.2*extent(4)]);
91     ylabel('Nennleistung in W');
92     %Get the hggroup
93     hg1 = get(gca, 'Children');
94     %Get the text annotations
95     ch2 = findobj(hg1, 'type', 'text');
96     %Modify position settings. This may be done programmatically or
97     interactively
98     set(ch2, 'Units', 'data');
99     set(ch2(1), 'Position', get(ch2(1), 'Position') + [0 -35 0]);
100    set(ch2(2), 'Position', get(ch2(2), 'Position') + [0 -35 0]);
101    %Make a new hggroup with the repositioned text annotations
102    copyobj(hg1,gca)
103    %delete the first hggroup
104    delete(hg1);
105 else
106     ret = false;
107 end %if(numel(power_max) ≠ 0)
108 end %function
```

### B.7. show\_energy\_consumption(...)

```
1 %Wertet den Energieverbrauch für das übergebene Gerät mit den gewünschten
2 %Tagen und Messreihen aus. Aus Konformitätsgründen ist der erste Parameter
3 %nicht verfügbar, da die Infostruktur 'geraete' nicht benötigt wird, aber
4 %die gleiche Parameterreihenfolge angestrebt wird.
5 function ret = show_energy_consumption(~, hh, sommer, winter, geraete_type,
    weekdays, type)
```

## B. MATLAB Quellcode

```
6
7 load('ger.mat');
8 load(strcat('cycles_',ger{geraete_type},' .mat'));
9
10 if (exist(strcat('energy_class_',ger{geraete_type},' .mat'),'file'))
11     load(strcat('energy_class_',ger{geraete_type},' .mat'));
12 end
13 %init
14 energy = [];
15 switch (type)
16     case 1
17         mode = 'Zyklus';
18     case 2
19         mode = 'Tag';
20     case 3
21         mode = 'Jahr';
22 end
23 %Alle Zeilen durchgehen
24 for i=1:size(cycle_info,1)
25     %Alle Elemente der Zeile durchgehen
26     day_count = 0;
27     sum_energy = 0;
28     for j=1:size(cycle_info,2);
29         %Falls in Liste gewünschter Haushalte enthalten
30         if (isstruct(cycle_info{i,j}) &#226; 0)
31             if any(strcmp(cycle_info{i,j}.hh,hh))
32                 %Sommer oder Wintermessung
33                 if (sommer == true && cycle_info{i,j}.sommer == 1 ||
34                     winter == true && cycle_info{i,j}.winter == 1)
35                     %Wochentag
36                     if any(strcmp(weekdays,datestr(cycle_info{i,j}.
37                         date,'ddd'))
38                         %Zyklusweiser Energieverbrauch
39                         switch (type)
40                             %zyklusweiser Energieverbrauch
41                             case 1
42                                 for k=1:size(cycle_info{i,j}.power,2)
43                                     temp = cycle_info{i,j}.power(k);
44                                     energy = [energy sum(temp{:})]; %#
45                                     ok< *AGROW>
46                                 end
47                             %tageweiser Energieverbrauch
48                             case 2
49                                 day_energy = 0;
50                                 day_count = day_count + 1;
51                                 for k=1:size(cycle_info{i,j}.power,2)
52                                     temp = cycle_info{i,j}.power(k);
53                                     day_energy = day_energy + sum(temp
54                                         {:});
55                                 end
56                             energy = [energy day_energy];
57                         %jahresweiser Energieverbrauch
```

## B. MATLAB Quellcode

```
54         case 3
55             day_energy = 0;
56             day_count = day_count + 1;
57             for k=1:size(cycle_info{i,j}.power,2)
58                 temp = cycle_info{i,j}.power(k);
59                 day_energy = day_energy + sum(temp
60                     {:});
61             end
62             sum_energy = sum_energy + day_energy;
63         end
64     end
65 end
66
67
68     end
69 end
70 if (type == 3)
71     year_energy = sum_energy * 365/day_count;
72     energy = [energy year_energy];
73 end
74 end
75
76 %Ergebnisse vorhanden?
77 if (numel(energy) ≠ 0)
78     % Rückgabewerte setzen
79     ret.analysis = 'show_energy_consumption';
80     ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
81     ret.geraete_type = geraete_type; ret.weekdays = weekdays;
82     ret.energy = energy;
83
84     ret.figHandles = figure;
85     hold on;
86     title(sprintf('Energieverbrauch %s pro %s, %d Haushalte: (%d Werte
87         )',ger{geraete_type},mode, numel(hh),numel(energy)));
88     xlabel('Gerät');
89     ylabel('Energieverbrauch in kWh');
90     if (exist('energy_class','var'))
91         for i=2:size(energy_class,1)
92             a = energy_class{i,2};
93             line([0 2],[a(1) a(1)],'color','m');
94             line([0 2],[a(2) a(2)],'color','m');
95             if (mod(i,2) == 0)
96                 b = 0.3;
97             else
98                 b = 0.4;
99             end
100             text(0.4+b,a(1) + (a(2)-a(1))/2,energy_class{i,1}, '
101                 BackgroundColor',[1 1 1]);
102         end
103     end
104     boxplot(energy./3.6e6);
```

## B. MATLAB Quellcode

```
103     fprintf('durchschnittlicher Energieverbrauch pro %s: %f kWh \n',
            mode,mean(energy)/3.6e6);
104     fprintf('maximaler Energieverbrauch pro %s: %f kWh \n',mode,max(
            energy)/3.6e6);
105     fprintf('minimaler Energieverbrauch pro %s: %f kWh \n',mode,min(
            energy)/3.6e6);
106     else
107         ret = false;
108     end %if (numel(energy) ≠ 0)
109 end
```

### B.8. show\_cycles(...)

```
1 % Diese Funktion zeigt die gewünschte Anzahl an Zyklen des jeweiligen
2 % Gerätes mit Hilfe des GUIs comparison.m an. Damit wird auch eine Auswahl
3 % der tatsächlich gewünschten Plots erlaubt und eine einfache Anpassung
4 % der
5 % Achsen sowie Speicherung der Daten ermöglicht.
6 function ret = show_cycles(~,hh,sommer,winter,geraete_type,weekdays,
7     cycle_count,pq,filename)
8     %Gerätenamen laden
9     load('ger.mat');
10    %Zyklen für diesen Gerätetypen laden
11    load(strcat('cycles_',ger{geraete_type}));
12    %Variablen initialisieren
13    %Leistungsverlauf
14    power2 = zeros(86400,1000);
15    reactive_power = zeros(86400,1000);
16    %Nummer an Zyklen
17    count = 0;
18    %Maximale Länge
19    max_duration = 0;
20    %Alle Zeilen und Spalten von cylce_info durchgehen
21    for i=1:size(cycle_info,1)
22        for k=1:size(cycle_info,2)
23            %Falls in diesem Element Daten vorhanden sind
24            if (isstruct(cycle_info{i,k}))
25                %Falls in Liste gewünschter Haushalte enthalten
26                if any(strcmp(cycle_info{i,k}.hh,hh))
27                    if (sommer == 1 && cycle_info{i,k}.sommer == 1) || (
28                        winter == 1 && cycle_info{i,k}.winter == 1)
29                        if any(strcmp(weekdays,datestr(cycle_info{i,k}.
30                            date,'ddd'))
31                            switch (pq)
32                                case 'p' %Wirkleistung
33                                    %Alle Zyklen durchgehen
34                                    for j=1:numel(cycle_info{i,k}.power)
35                                        %Dauer ermitteln
36                                        duration = numel(cycle_info{i,k}.
37                                            power{j});
```

## B. MATLAB Quellcode

```
33         %Anzahl um 1 erhöhen
34         count = count + 1;
35         %Verlauf speichern
36         power2(1:duration,count) =
            cycle_info{i,k}.power{j};
37         %Legendeneintrag erstellen
38         %legende{count} = sprintf('
            Haushalt %s, Tag %d, Zyklus %d
            ',cycle_info{i,k}.hh,cycle_info
            {i,k}.day_index,j);
39         legende{count} = sprintf('Haushalt
            %s',cycle_info{i,k}.hh);
40         %maximale Dauer ermitteln
41         max_duration = max(duration,
            max_duration);
42         %Diagrammtitel
43         titel = strcat('
            Wirkleistungsverlauf während
            eines Zyklus einer(s) ',{' '},
            ger{geraete_type});
44     end %for j=1:numel(cycle_info{i,k}.
        power)
45     case {'q','pq'} %Blindleistung oder beides
46         %Alle Zyklen durchgehen
47         if (isfield(cycle_info{i,k},'
            reactive_power'))
48             for j=1:numel(cycle_info{i,k}.
                reactive_power)
49                 %Dauer ermitteln
50                 duration = numel(cycle_info{i,
                    k}.power{j});
51                 duration2 = numel(cycle_info{i
                    ,k}.reactive_power{j});
52                 %Anzahl um 1 erhöhen
53                 count = count + 1;
54                 %Verlauf speichern
55                 power2(1:duration,count) =
                    cycle_info{i,k}.power{j};
56                 reactive_power(1:duration2,
                    count) = cycle_info{i,k}.
                    reactive_power{j};
57                 %Legendeneintrag erstellen
58                 %legende{count} = sprintf('P
                    Haushalt %s, Tag %d, Zyklus
                    %d',cycle_info{i,k}.hh,
                    cycle_info{i,k}.day_index,j
                    );
59                 %legende2{count} = sprintf('Q
                    Haushalt %s, Tag %d, Zyklus
                    %d',cycle_info{i,k}.hh,
                    cycle_info{i,k}.day_index,j
                    );
```

## B. MATLAB Quellcode

```
60         legende{count} = sprintf('P
           Haushalt %s', cycle_info{i,k
           }.hh);
61         legende2{count} = sprintf('Q
           Haushalt %s', cycle_info{i,k
           }.hh);
62         %maximale Dauer ermitteln
63         max_duration = max([duration,
           duration2], max_duration);
64         %Diagrammtitel
65         titel = strcat('
           Blindleistungsverlauf
           während eines Zyklus einer(
           s) ', {' '}, ger{geraete_type
           });
66         end %for j=1:numel(cycle_info{i,k
           }.power)
67         else
68             fprintf('Zyklus %d,%d:
           Blindleistung fehlt!\n', i,k);
69         end %if (isfield(cycle_info{i,k}, '
           reactive_power'))
70         end %switch (pq)
71         end %if weekdays
72         end %if sommer || winter
73         end %if any(strcmp(cycle_info{i,k}.hh,hh))
74         end %if (isstruct(cycle_info{i,k}))
75         end %for k=1:size(cycle_info,2)
76     end %for i=1:size(cycle_info,1)
77
78     %Falls Zyklen gefunden
79     if (count > 0)
80         %Anzahl der Zyklen im Plot ermitteln
81         switch (cycle_count)
82             case 1
83                 cycle = randi(count);
84                 power = power2(1:max_duration,cycle);
85                 react_power = reactive_power(1:max_duration,cycle);
86                 legende{1} = legende{cycle};
87                 if(strcmp(pq, 'q') == 1)
88                     legende2{1} = legende2{cycle};
89                 end
90             case 'all'
91                 power = power2(1:max_duration,1:count);
92                 react_power = reactive_power(1:max_duration,1:count);
93             otherwise
94                 cycle_index = 1:count;
95                 if (numel(cycle_index) < cycle_count)
96                     cycle_count = numel(cycle_index);
97                 end
98                 for i=1:cycle_count
99                     a = randi(numel(cycle_index),1);
```

## B. MATLAB Quellcode

```
100         cycles(i) = cycle_index(a);
101         cycle_index = [ cycle_index(1:a-1) cycle_index(a+1:end
102             )];
103     end
104     power = power2(1:max_duration,cycles);
105     react_power = reactive_power(1:max_duration,cycles);
106     legende = legende(cycles);
107     if (strcmp(pq, 'q') == 1) || (strcmp(pq, 'pq') == 1)
108         legende2 = legende2(cycles);
109     end
110 end %switch(cycle_count)
111
112 %Plot erstellen (unter Verwendung von comparison.m)
113 switch (pq)
114     case 'p'
115         plotting = power;
116         legenden = legende;
117         type = 2;
118     case 'q'
119         plotting = react_power;
120         legenden = legende2;
121         type = 2;
122     case 'pq'
123         plotting = [power react_power];
124         legenden = [legende, legende2];
125         type = 3;
126     end
127 x_label = 'Zeit in Sekunden';
128 y_label = 'Leistung in W';
129 comparison(1:max_duration,plotting',legenden,titel,filename,type,
130     x_label,y_label);
131 ret.figHandles = gcf;
132 ret.power = power2(1:max_duration,1:count);
133 ret.legenden = legenden; ret.analysis = 'show_cycles';
134 ret.hh = hh; ret.weekdays = weekdays;
135 ret.sommer = sommer; ret.winter = winter;
136 ret.plotting = plotting;
137 else
138     ret = false;
139 end
140 end
```

### B.9. show\_mean\_power(...)

```
1 % Diese Funktion ermittelt die mittlere Tagesleistung pro Person für die
2 % übergebenen Messreihen, Wochentage sowie das gewählte Gerät
3 function ret = show_mean_power(~,hh,sommer, winter,geraete_type,weekdays)
4     %Gerätenamen laden
5     load('ger.mat');
6     %Zyklen für diesen Gerätetypen laden
```

## B. MATLAB Quellcode

```
7 load(strcat('cycles_',ger{geraete_type}));
8 load('hh_data3.mat');
9 %Variablen initialisieren
10 %Leistungsverlauf
11 power2 = zeros(86400,1000);
12 %Nummer an Tagen
13 count = 1;
14 %Alle Zeilen und Spalten von cycle_info durchgehen
15 for i=1:size(cycle_info,1)
16     for k=1:size(cycle_info,2)
17         %Falls in diesem Element Daten vorhanden sind
18         if (isstruct(cycle_info{i,k}))
19             %Falls in Liste gewünschter Haushalte enthalten
20             if any(strcmp(cycle_info{i,k}.hh,hh))
21                 %Sommer oder Wintermessung
22                 if (sommer == true && cycle_info{i,k}.sommer == 1 ||
23                     winter == true && cycle_info{i,k}.winter == 1)
24                     %Wochentag
25                     if any(strcmp(weekdays,datestr(cycle_info{i,k}.
26                         date,'ddd')))
27                         %Haushaltsgröße ermitteln (Standardwert 2)
28                         persons = 2;
29                         for j=1:size(hh_data,2)
30                             if strcmp(hh_data{j}.filename,cycle_info{i
31                                 ,k}.hh) ≠ 0
32                                 persons = hh_data{j}.num_persons;
33                             end
34                         end
35                         %alle Zyklen an diesem Tag durchgehen
36                         for j=1:min(size(cycle_info{i,k}.start,2),size
37                             (cycle_info{i,k}.power,2))
38                             temp = cycle_info{i,k}.power{j};
39                             power2(cycle_info{i,k}.start(j):cycle_info
40                                 {i,k}.start(j) + size(temp,1) - 1,count
41                                 ) = temp / persons;
42                         end
43                     end
44                     count = count + 1;
45                 end % if (weekday)
46             end %if (sommer || winter)
47         end %if any(strcmp(cycle_info{i,k}.hh,hh))
48     end %if (isstruct(cycle_info{i,k}))
49 end %for k=1:size(cycle_info,2)
50 count = count - 1;
51 %Falls Zyklen gefunden – Plot erstellen
52 if (count > 0)
53     %Rückgabewerte
54     ret.analysis = 'show_mean_power';
55     ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
56     ret.geraete_type = geraete_type; ret.weekdays = weekdays;
57 end
58 %sekundenweise Auflösung
```

## B. MATLAB Quellcode

```
53     mean_power = mean(power2(:,1:count),2);
54     ret.figHandles(1) = figure();
55     plot(1/86400:1/86400:1,mean_power(1:86400));
56     ret.power = power2(:,1:count);
57     ret.mean_power = mean_power;
58     title({sprintf('mittlere Leistung pro Person für %s',ger{
        geraete_type}),sprintf('%d Haushalte: (%d Zyklen)', numel(hh),
        count)}});
59     xlabel = '';
60     for i=0:24
61         xlabel = strcat(xlabel,sprintf('%02d:00',i),'|');
62     end
63     xlabel = xlabel(1:end-1);
64     set(gca,'xlim',[0 1]);
65     set(gca,'xtick',0:1/24:1,'xticklabel',xlabel);
66     rotateticklabel(gca,90);
67     xlabel('Uhrzeit hh:mm');
68     ylabel('mittlere Leistung in W');
69     moveLabel('x',40,ret.figHandles(1),gca);
70     grid on;
71
72     %Minuten – Auflösung
73     ret.figHandles(2) = figure();
74     mean_power = decimate(mean_power,60);
75     plot(1/1440:1/1440:1,mean_power(1:1440));
76     ret.mean_power_minute = mean_power;
77     title({sprintf('mittlere Leistung pro Person für %s',ger{
        geraete_type}),sprintf('%d Haushalte: (%d Zyklen)', numel(hh),
        count)}});
78     set(gca,'xlim',[0 1]);
79     set(gca,'xtick',0:1/24:1,'xticklabel',xlabel);
80     rotateticklabel(gca,90);
81     xlabel('Uhrzeit hh:mm');
82     ylabel('mittlere Leistung in W');
83     moveLabel('x',35,ret.figHandles(2),gca);
84     grid on;
85
86     %Viertelstunden Auflösung
87     ret.figHandles(3) = figure();
88     mean_power = decimate(mean_power,15);
89     plot(1/96:1/96:1,mean_power(1:96));
90     ret.power = power2(:,1:count);
91     ret.mean_power_1_4 = mean_power;
92     title(sprintf('mittlere Leistung pro Person für %s %d Haushalte',
        ger{geraete_type}, numel(hh)));
93     set(gca,'ylim',[0 max(mean_power)*1.5]);
94     set(gca,'xlim',[0 1]);
95     set(gca,'xtick',0:1/24:1,'xticklabel',xlabel);
96     rotateticklabel(gca,90);
97     xlabel('Uhrzeit hh:mm');
98     ylabel('mittlere Leistung in W');
99     moveLabel('x',35,ret.figHandles(3),gca);
```

## B. MATLAB Quellcode

```
100     grid on;
101
102     %Stundenauflösung
103     ret.figHandles(4) = figure();
104     mean_power = decimate(mean_power,4);
105     if (size(mean_power,1) ≠ 25)
106         mean_power(25) = mean_power(24);
107     end
108     plot(0:1/24:1,mean_power(1:25));
109     ret.mean_power_1 = mean_power;
110     title(sprintf('mittlere Leistung pro Person für %s %d Haushalte',
111                 ger{geraete_type}, numel(hh)));
112     set(gca, 'ylim', [0 max(mean_power)*1.5]);
113     set(gca, 'xlim', [0 1]);
114     set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
115     rotateticklabel(gca, 90);
116     xlabel('Uhrzeit hh:mm');
117     ylabel('mittlere Leistung in W');
118     moveLabel('x', 40, ret.figHandles(4), gca);
119     grid on;
120 end %if (count > 1)
end
```

### B.10. show\_usage\_comparison(...)

```
1 function ret=show_usage_comparison(geraete, hh, sommer, winter, geraete_type,
2   weekdays, aufloesung, filename)
3     %init
4     load('ger.mat');
5     j=0; i=0;
6     summe = 0;
7     %Sommer-Messreihe
8     if (sommer == 1)
9         %Für jeden Wochentag einzeln bestimmen
10        for i=1:numel(weekdays)
11            ret(i) = show_usage_histogram(geraete, hh, sommer, 0,
12                geraete_type, weekdays{i}, aufloesung, false);
13            plotting(i, :) = ret(i).nelements;
14            legende{i} = strcat(weekdays{i}, {' '}, 'Sommer');
15            summe = summe + ret(i).nelements;
16        end
17        %Mittelwert ermitteln
18        ret(i+1).nelements = summe./i;
19        plotting(i+1, :) = summe./i;
20        ret(i+1).centers = ret(i).centers;
21        legende{i+1} = 'Sommer';
22        %Falls alle Wochentage übergeben wurden, Wochentage und Wochenende
23        %mitteln
24        if (i==7)
25            %Wochentage
```

## B. MATLAB Quellcode

```
24     plotting(i+2,:) = sum(plotting(1:5,:),1)./5;
25     legende{i+2} = 'Sommer Wochentag';
26     ret(i+2).nelements = plotting(i+2,:);
27     ret(i+2).centers = ret(i).centers;
28     %Wochenende
29     plotting(i+3,:) = sum(plotting(6:7,:))./2;
30     legende{i+3} = 'Sommer Wochenende';
31     ret(i+3).nelements = plotting(i+3,:);
32     ret(i+3).centers = ret(i).centers;
33     end
34     i = i + 3;
35 end
36 summe = 0;
37 %Winter-Messreihe
38 if (winter == 1)
39     %Für jeden Wochentag einzeln bestimmen
40     for j=1:numel(weekdays)
41         ret(i+j) = show_usage_histogram(geraete, hh,0,winter,
42             geraete_type,weekdays{j},aufloesung,false);
43         plotting(i+j,:) = ret(i+j).nelements;
44         legende{i+j} = strcat(weekdays{j},{' ','Winter'});
45         summe = summe + ret(i+j).nelements;
46     end
47     %Mittelwert ermitteln
48     ret(i+j+1).nelements = summe/j;
49     plotting(i+j+1,:) = summe/j;
50     ret(i+j+1).centers = ret(i).centers;
51     legende{j+i+1} = 'Winter';
52     %Falls alle Wochentage übergeben wurden, Wochentage und Wochenende
53     %mitteln
54     if (j==7)
55         %Wochentage
56         plotting(i+j+2,:) = sum(plotting(11:15,:),1)./5;
57         legende{i+j+2} = 'Winter Wochentag';
58         ret(i+j+2).nelements = plotting(i+j+2,:);
59         ret(i+j+2).centers = ret(i).centers;
60         %Wochenende
61         plotting(i+j+3,:) = sum(plotting(16:17,:))./2;
62         legende{i+j+3} = 'Winter Wochenende';
63         ret(i+j+3).nelements = plotting(i+j+3,:);
64         ret(i+j+3).centers = ret(i).centers;
65     end
66     j = j + 3;
67 end
68 k = i + j;
69 ret(k).plotting = plotting;
70 ret(k).legenden = vertcat(legende{:});
71 x_label = 'Uhrzeit HH:MM';
72 y_label = 'Einschalhäufigkeit';
73 titel = sprintf('Einschalhäufigkeit %s %d Haushalte',ger{geraete_type
74     },numel(hh));
```

## B. MATLAB Quellcode

```
73     comparison(ret(k).centers,plotting,ret(k).legenden,titel,filename,1,  
74     x_label,y_label);  
ret(1).figHandles = gcf;
```

### B.11. show\_usage\_comparison\_day(...)

```
1 function ret=show_usage_comparison_day(geraete,hh,sommer,winter,  
   geraete_type,weekdays,filename)  
2     %init  
3     load('ger.mat');  
4     j=0;i=0;  
5     summe = 0;  
6     %Sommer-Messreihe  
7     if (sommer == 1)  
8         %Für jeden Wochentag einzeln bestimmen  
9         for i=1:numel(weekdays)  
10            ret(i) = show_usage_count(geraete, hh,sommer,0,geraete_type,  
   weekdays{i},false);  
11            plotting(i,:) = ret(i).nelements;  
12            legende{i} = strcat(weekdays{i},{ ' '},'Sommer');  
13            summe = summe + ret(i).nelements;  
14        end  
15        %Mittelwert ermitteln  
16        ret(i+1).nelements = summe./i;  
17        plotting(i+1,:) = summe./i;  
18        ret(i+1).centers = ret(i).centers;  
19        legende{i+1} = 'Sommer';  
20        %Falls alle Wochentage übergeben wurden, Wochentage und Wochenende  
21        %mitteln  
22        if (i==7)  
23            %Wochentage  
24            plotting(i+2,:) = sum(plotting(1:5,:),1)./5;  
25            legende{i+2} = 'Sommer Wochentag';  
26            ret(i+2).nelements = plotting(i+2,:);  
27            ret(i+2).centers = ret(i).centers;  
28            %Wochenende  
29            plotting(i+3,:) = sum(plotting(6:7,:))./2;  
30            legende{i+3} = 'Sommer Wochenende';  
31            ret(i+3).nelements = plotting(i+3,:);  
32            ret(i+3).centers = ret(i).centers;  
33        end  
34        i = i + 3;  
35    end  
36    summe = 0;  
37    %Winter-Messreihe  
38    if (winter == 1)  
39        %Für jeden Wochentag einzeln bestimmen  
40        for j=1:numel(weekdays)  
41            ret(i+j) = show_usage_count(geraete, hh,0,winter,geraete_type,  
   weekdays{j},false);
```

## B. MATLAB Quellcode

```
42     plotting(i+j,:) = ret(i+j).nelements;
43     legende{i+j} = strcat(weekdays{j},{' ','Winter'});
44     summe = summe + ret(i+j).nelements;
45     end
46     %Mittelwert ermitteln
47     ret(i+j+1).nelements = summe/j;
48     plotting(i+j+1,:) = summe/j;
49     ret(i+j+1).centers = ret(i).centers;
50     legende{j+i+1} = 'Winter';
51     %Falls alle Wochentage übergeben wurden, Wochentage und Wochenende
52     %mitteln
53     if (j==7)
54         %Wochentage
55         plotting(i+j+2,:) = sum(plotting(11:15,:),1)./5;
56         legende{i+j+2} = 'Winter Wochentag';
57         ret(i+j+2).nelements = plotting(i+j+2,:);
58         ret(i+j+2).centers = ret(i).centers;
59         %Wochenende
60         plotting(i+j+3,:) = sum(plotting(16:17,:))./2;
61         legende{i+j+3} = 'Winter Wochenende';
62         ret(i+j+3).nelements = plotting(i+j+3,:);
63         ret(i+j+3).centers = ret(i).centers;
64     end
65     j = j + 3;
66     end
67     k = i + j;
68     ret(k).plotting = plotting;
69     ret(k).legenden = vertcat(legende{:});
70     x_label = 'Zyklen / Tag';
71     y_label = 'Anzahl';
72     titel = sprintf('Einschaltheufigkeit %s für verschiedene Tage %d
73     Haushalte',ger{geraete_type},numel(hh));
74     comparison(ret(k).centers,plotting',ret(k).legenden,titel,filename,4,
75     x_label,y_label);
76     ret(1).figHandles =(gcf;
```

### B.12. show\_load\_profile(...)

```
1 function ret = show_load_profile(hh,sommer,winter,weekdays,detail_plot,
2     mean_plot,sum_plot,pq)
3     % note that there is no difference in the shape of the mean and sum plots
4     % consider removing one of them!
5     % note that the detail plot may contain lots of lines so consider if you
6     % can really use that one?!
7     load('hh_data.mat');
8     load('hh2.mat');
9     global H0dyn;
10    if (~istable(H0dyn))
11        load('H0dyn.mat');
12    end
```

## B. MATLAB Quellcode

```
12 %Variablen sind global, um bei mehrmaligem Aufruf nicht jedes Mal alles
    neu
13 %laden zu müssen, da dies sehr zeitaufwändig ist.
14 global load_profile;
15 global load_profile_Q;
16 global power_factor;
17 c = cellstr(H0dyn.date);
18 switch (pq)
19     %Wirkleistung
20     case 'p'
21         if(~iscell(load_profile))
22             load('load_profile.mat');
23         end
24         load_profile2 = load_profile;
25     %Blindleistung
26     case 'q'
27         if(~iscell(load_profile_Q))
28             load('load_profile_Q.mat');
29         end
30         load_profile2 = load_profile_Q;
31     %Leistungsfaktor
32     case 'pf'
33         if(~iscell(power_factor))
34             load('power_factor');
35         end
36         load_profile2 = power_factor;
37 end
38
39 %global h0data;
40 %global data;
41
42 k = 1;
43 data = zeros(86400,1000);
44 h0data = zeros(96,1000);
45 legende = cell(1,1000);
46 legende2 = cell(1,1000);
47
48 done = false;
49
50 index = 2;
51 %alle übergebenen Haushalte durchgehen
52 for i=1:numel(hh)
53     m = index;
54     while (strcmp(load_profile2{m,1},hh{i}) ≠ 1)
55         m = m+1;
56     end
57     %so lange der gleiche Haushalt vorne steht
58     while (strcmp(load_profile2{m,1},hh{i}) == 1)
59         %falls Sommermessung gewünscht
60         if (sommer == true)
61             %falls Montag gewünscht und vorhanden
```

## B. MATLAB Quellcode

```
62     if (any(strcmp(weekdays, 'Mon')) && numel(load_profile2{m
63         ,3}) ≠ 0)
64         %Daten kopieren
65         data(:,k) = [load_profile2{m,3}];
66         done = 1;
67     end
68     %falls Dienstag gewünscht
69     if (any(strcmp(weekdays, 'Tue')) && numel(load_profile2{m
70         ,4}) ≠ 0)
71         %Daten kopieren
72         data(:,k) = [load_profile2{m,4}];
73         done = 1;
74     end %if (Tuesday)
75     %falls Mittwoch gewünscht
76     if (any(strcmp(weekdays, 'Wed')) && numel(load_profile2{m
77         ,5}) ≠ 0)
78         %Daten kopieren
79         data(:,k) = [load_profile2{m,5}];
80         done = 1;
81     end %if (wednesday)
82     %falls Donnerstag gewünscht
83     if (any(strcmp(weekdays, 'Thu')) && numel(load_profile2{m
84         ,6}) ≠ 0)
85         %Daten kopieren
86         data(:,k) = [load_profile2{m,6}];
87         done = 1;
88     end %if (thursday)
89     %falls Freitag gewünscht
90     if (any(strcmp(weekdays, 'Fri')) && numel(load_profile2{m
91         ,7}) ≠ 0)
92         %Daten kopieren
93         data(:,k) = [load_profile2{m,7}];
94         done = 1;
95     end %if (friday)
96     %falls Samstag gewünscht
97     if (any(strcmp(weekdays, 'Sat')) && numel(load_profile2{m
98         ,8}) ≠ 0)
99         %Daten kopieren
100        data(:,k) = [load_profile2{m,8}];
101        done = 1;
102    end %if (saturday)
103    %falls Sonntag gewünscht
104    if (any(strcmp(weekdays, 'Sun')) && numel(load_profile2{m
105        ,9}) ≠ 0)
106        %Daten kopieren
107        data(:,k) = [load_profile2{m,9}];
108        done = 1;
109    end %if (sunday)
110    if (done == 1)
111        %Legendeneinträge für den Detailplot
112        legende{k} = strcat(hh{i}, ' # ', datestr(load_profile2{
113            m,2}, 'dd.mm.yyyy.'));
```

## B. MATLAB Quellcode

```
106         legende2{k} = strcat('H0dyn # ',datestr(load_profile2{
           m,2},'dd.mm.yyyy.'));
107         %Suche nach gleichem Tag im H0 Profil
108         found = 0;
109         a = strcmp(datestr(load_profile2{m,2},'dd.mm'),c);
110         for n=1:numel(a)
111             if (a(n) == 1)
112                 found = n;
113                 break;
114             end
115         end
116         if (found ≠ 0)
117             h0data(:,k) = H0dyn.power(n:n+95);
118         end
119         k = k + 1;
120         done = 0;
121     end %if(done == 1)
122 end %if(sommer == true)
123
124 %falls Wintermessung gewünscht
125 if (winter == true)
126     %falls Montag gewünscht und vorhanden
127     if (any(strcmp(weekdays,'Mon')) && numel(load_profile2{m
           ,13}) ≠ 0)
128         %Daten kopieren
129         data(:,k) = [load_profile2{m,13}];
130         done = 1;
131     end
132     %falls Dienstag gewünscht
133     if (any(strcmp(weekdays,'Tue')) && numel(load_profile2{m
           ,14}) ≠ 0)
134         %Daten kopieren
135         data(:,k) = [load_profile2{m,14}];
136         done = 1;
137     end %if (Tuesday)
138     %falls Mittwoch gewünscht
139     if (any(strcmp(weekdays,'Wed')) && numel(load_profile2{m
           ,15}) ≠ 0)
140         %Daten kopieren
141         data(:,k) = [load_profile2{m,15}];
142         done = 1;
143     end %if (wednesday)
144     %falls Donnerstag gewünscht
145     if (any(strcmp(weekdays,'Thu')) && numel(load_profile2{m
           ,16}) ≠ 0)
146         %Daten kopieren
147         data(:,k) = [load_profile2{m,16}];
148         done = 1;
149     end %if (thursday)
150     %falls Freitag gewünscht
151     if (any(strcmp(weekdays,'Fri')) && numel(load_profile2{m
           ,17}) ≠ 0)
```

## B. MATLAB Quellcode

```
152         %Daten kopieren
153         data(:,k) = [load_profile2{m,17}];
154         done = 1;
155     end %if (friday)
156     %falls Samstag gewünscht
157     if (any(strcmp(weekdays, 'Sat')) && numel(load_profile2{m
158         ,18}) ≠ 0)
159         %Daten kopieren
160         data(:,k) = [load_profile2{m,18}];
161         done = 1;
162     end %if (saturday)
163     %falls Sonntag gewünscht
164     if (any(strcmp(weekdays, 'Sun')) && numel(load_profile2{m
165         ,19}) ≠ 0)
166         %Daten kopieren
167         data(:,k) = [load_profile2{m,19}];
168         done = 1;
169     end %if (sunday)
170     if (done == 1)
171         %Legendeneinträge für den Detailplot
172         legende{k} = strcat(hh{i}, ' # ', datestr(load_profile2{
173             m,2}, 'dd.mm.yyyy. '));
174         legende2{k} = strcat('H0dyn # ', datestr(load_profile2{
175             m,2}, 'dd.mm.yyyy. '));
176         %Suche nach gleichem Tag im H0 Profil
177         found = 0;
178         a = strcmp(datestr(load_profile2{m,2}, 'dd.mm'), c);
179         for n=1:numel(a)
180             if (a(n) == 1)
181                 found = n;
182                 break;
183             end
184         end
185         if (found ≠ 0)
186             h0data(:,k) = H0dyn.power(n:n+95);
187         end
188         k = k + 1;
189         done = 0;
190     end %if(done == 1)
191     end %if(winter == true)
192     if (m < size(load_profile2,1))
193         m = m + 1;
194     else
195         break;
196     end
197     end %while(hh{i})
198     if (m < size(load_profile2,1))
199         index = m;
200     end
201 end %for i=1:numel(hh)
202 %Mittelwert der Lastgänge ermitteln
```

## B. MATLAB Quellcode

```
200 data(:,k) = nanmean(data(:,1:k-1),2)';
201 h0data(:,k) = mean(h0data(:,1:k-1),2)';
202 %H0 Lastprofil normieren
203 s = sum(data(:,k));
204 h0s = sum(h0data(:,k));
205 h0data = h0data * (s/(h0s*900));
206 %Summenwert der Lastgänge ermitteln
207 sumdata = sum(data(:,1:k-1),2)';
208 sumh0data = sum(h0data(:,1:k-1),2)';
209 %H0 Lastprofil normieren
210 s = sum(sumdata);
211 h0s = sum(sumh0data);
212 sumh0data = sumh0data*(s/(h0s*900));
213 %Legendeneintrag erstellen
214 legende{k} = 'mittleres Lastprofil Sommer';
215 legende2{k} = 'mittleres H0 Profil Sommer';
216 xlabel = '';
217 for i=0:24
218     xlabel = strcat(xlabel, sprintf('%02d:00',i), '|');
219 end
220 xlabel = xlabel(1:end-1);
221 weekdays = strcat(' ', ' ');
222
223 %Rückgabewerte setzen
224 ret.analysis = 'show_laod_profile';
225 ret.hh = hh; ret.sommer = sommer; ret.winter = winter;
226 ret.weekdays = weekdays;
227 ret.load_profiles = data(:,1:k-1);
228 ret.time_mean_load = 1/86400:1/86400:1;
229 ret.mean_load = data(:,k);
230 ret.time_mean_h0 = 1/96:1/96:1;
231 ret.mean_h0 = h0data(:,k);
232 ret.sum_load = sumdata;
233 ret.sumh0 = sumh0data;
234 ret.figHandles = [];
235 ret.num_days = k-1;
236
237 %Falls plot der einzelnen Tageslastgänge erwünscht
238 if (detail_plot)
239     ret.figHandles(end+1) = figure;
240     hold on;
241     set(gca, 'xlim', [0 1]);
242     set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
243     rotateticklabel(gca, 90);
244     %plotten
245     plot(1/86400:1/86400:1, data(:,1:k));
246     plot(1/96:1/96:1, h0data(:,1:k));
247     legend(legende{1:k}, legende2{1:k});
248 end %if(detail_plot)
249
250 %Falls plot der mittleren Lastgänge erwünscht
251 if (mean_plot && k > 1)
```

## B. MATLAB Quellcode

```
252     ret.figHandles(end+1) = figure;
253     hold on;
254     set(gca, 'xlim', [0 1]);
255     set(gca, 'ylim', [0, max(data(:,k))]);
256     set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
257     rotateticklabel(gca, 90);
258     %plotten
259     plot(1/86400:1/86400:1, data(:,k));
260     switch (pq)
261         case {'p', 'q'}
262             plot(1/96:1/96:1, h0data(:,k), 'r');
263         case {'pf'}
264             set(gca, 'ylim', [0, 1]);
265     end
266     legend(legende{k}, legende2{k});
267     grid on;
268     title(strcat('mittleres Haushaltslastprofil', {' '}, cell2mat(
        weekdays), ' von', sprintf(' %d ', k-1), ' Tagen'));
269     xlabel('Uhrzeit (hh:mm)');
270     ylabel('Leistung in W');
271     moveLabel('x', 30, ret.figHandles(end), gca);
272 end %if(mean_plot)
273
274 %Falls plot der Summenlastgänge erwünscht
275 if (sum_plot)
276     ret.figHandles(end+1) = figure;
277     hold on;
278     set(gca, 'xlim', [0 1]);
279     set(gca, 'ylim', [0, max(sumdata)]);
280     set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
281     rotateticklabel(gca, 90);
282     %plotten
283     plot(1/86400:1/86400:1, sumdata);
284     plot(1/96:1/96:1, sumh0data, 'r');
285     legend(legende{k}, legende2{k});
286     grid on;
287     title(strcat('Summenhaushaltslastprofil', {' '}, cell2mat(weekdays),
        ' von', sprintf(' %d ', k-1), ' Tagen'));
288 end %if(sum_plot)
289 end
```

### B.13. show\_restprofil(...)

```
1 %Diese Funktion zeigt das vorher berechnete Restprofil des gewählten
2 %Haushaltes für die übergebenen Tagestypen an. Dabei werden zufällig
   num_days
3 %Tage ausgewählt und angezeigt, falls mehr vorhanden sind.
4 function ret = show_restprofil2(hh, sommer, winter, weekdays, num_days)
5 %Haushaltsübersicht laden
6 load('hh_data3.mat');
```

## B. MATLAB Quellcode

```
7
8 %Haushaltsdaten nachladen, falls noch nicht geschehen
9 global Data;
10 global Data_hh;
11 if strcmp(Data_hh,hh) ≠ 1
12     load(strcat(hh, '+rest.mat'));
13     Data_hh = hh;
14 end
15
16 %Gewünschten Haushalt suchen und index ermitteln
17 for i=1:numel(hh_data)
18     if strcmp(hh_data{i}.filename,hh)==1
19         hh_index = i;
20         hh_data2 = hh_data{i};
21         break;
22     end
23 end
24
25 %init
26 ret.restprofil_L1 = zeros(86400,50);
27 ret.original_L1 = zeros(86400,50);
28 ret.restprofil_L2 = zeros(86400,50);
29 ret.original_L2 = zeros(86400,50);
30 ret.restprofil_L3 = zeros(86400,50);
31 ret.original_L3 = zeros(86400,50);
32 count = 0;
33 %Falls Sommermessung gewünscht
34 if (sommer == true)
35     %Alle Tage durchgehen
36     for i=2:size(hh_data2.days,1)
37         %Falls richtiger Wochentag
38         day = hh_data2.days(i,:);
39         if any(strcmp(weekdays,datestr(day{1},'ddd'))) && ~isempty(day{3})
40             %Alle drei Phasen getrennt speichern
41             count = count + 1;
42             ret.restprofil_L1(:,count) = Data{1,day{3}}.data
43                 (300:86699,24);
44             ret.original_L1(:,count) = Data{1,day{3}}.data(300:86699,8);
45             ret.restprofil_L2(:,count) = Data{1,day{3}}.data
46                 (300:86699,25);
47             ret.original_L2(:,count) = Data{1,day{3}}.data(300:86699,9);
48             ret.restprofil_L3(:,count) = Data{1,day{3}}.data
49                 (300:86699,26);
50             ret.original_L3(:,count) = Data{1,day{3}}.data(300:86699,10);
51         end %weekdays
52     end %for i=2:size(days,2)
53 end %if (sommer == true)
54
55 %Falls Wintermessung gewünscht
56 if (winter == true)
57     %Alle Tage durchgehen
58     for i=2:size(hh_data2.days,1)
```

## B. MATLAB Quellcode

```
56     %Falls richtiger Wochentag
57     day = hh_data2.days(i,:);
58     if any(strcmp(weekdays,datestr(day{1},'ddd'))) && ~isempty(day{5})
59         %Alle drei Phasen getrennt speichern
60         count = count + 1;
61         ret.restprofil_L1(:,count) = Data{1,day{5}}.data
62             (300:86699,24);
63         ret.original_L1(:,count) = Data{1,day{5}}.data(300:86699,8);
64         ret.restprofil_L2(:,count) = Data{1,day{5}}.data
65             (300:86699,25);
66         ret.original_L2(:,count) = Data{1,day{5}}.data(300:86699,9);
67         ret.restprofil_L3(:,count) = Data{1,day{5}}.data
68             (300:86699,26);
69         ret.original_L3(:,count) = Data{1,day{5}}.data(300:86699,10);
70     end %weekdays
71 end %for i=2:size(days,2)
72 end %if (winter == true)
73
74 %Falls Zyklen gefunden
75 if (count > 0)
76     %Anzahl der Zyklen im Plot ermitteln
77     switch (num_days)
78     case 1
79         cycle = randi(count);
80         ret.restprofil_L1_plot = ret.restprofil_L1(:,cycle);
81         ret.original_L1_plot = ret.original_L1(:,cycle);
82         ret.restprofil_L2_plot = ret.restprofil_L2(:,cycle);
83         ret.original_L2_plot = ret.original_L2(:,cycle);
84         ret.restprofil_L3_plot = ret.restprofil_L3(:,cycle);
85         ret.original_L3_plot = ret.original_L3(:,cycle);
86         ret.cycle = cycle;
87         ret.figHandles = figure;
88         ret.plot_handle(1,:) = plot(1/86400:1/86400:1,ret.
89             original_L1_plot,'b-');
90         hold on;
91         ret.plot_handle(2,:) = plot(1/86400:1/86400:1,ret.
92             restprofil_L1_plot,'b-','lineWidth',1.5);
93
94         ret.plot_handle(3,:) = plot(1/86400:1/86400:1,ret.
95             original_L2_plot,'g-');
96         ret.plot_handle(4,:) = plot(1/86400:1/86400:1,ret.
97             restprofil_L2_plot,'g-','lineWidth',1.5);
98         ret.plot_handle(5,:) = plot(1/86400:1/86400:1,ret.
99             original_L3_plot,'m-');
100        ret.plot_handle(6,:) = plot(1/86400:1/86400:1,ret.
101            restprofil_L3_plot,'m-','lineWidth',1.5);
102        title('Leistungsverlauf des Restprofils')
103        ylabel('Leistung in W');
104        xlabel('Uhrzeit HH:MM');
105        xlabels = '';
106        for k=0:24
107            xlabels = strcat(xlabel,'%02d:00',k),'|';
108        end
109    end
110 end
```

## B. MATLAB Quellcode

```

99     end
100    xlabel = xlabel(1:end-1);
101    set(gca, 'xlim', [0 1]);
102    set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
103    rotateticklabel(gca, 90);
104    moveLabel('x', 35, ret.figHandles(1), gca);
105    legend(ret.plot_handle([2 4 6]), {'L1', 'L2', 'L3'});
106    grid on;
107    case 'all'
108        ret.restprofil_L1_plot = ret.restprofil_L1(:, 1:count);
109        ret.original_L1_plot = ret.original_L1(:, 1:count);
110        ret.restprofil_L2_plot = ret.restprofil_L2(:, 1:count);
111        ret.original_L2_plot = ret.original_L2(:, 1:count);
112        ret.restprofil_L3_plot = ret.restprofil_L3(:, 1:count);
113        ret.original_L3_plot = ret.original_L3(:, 1:count);
114        for i=1:count
115            ret.figHandles(i) = figure;
116            ret.plot_handle(1, :) = plot(1/86400:1/86400:1, ret.
                original_L1(:, i), 'b-');
117            hold on;
118            ret.plot_handle(2, :) = plot(1/86400:1/86400:1, ret.
                restprofil_L1(:, i), 'b-', 'lineWidth', 1.5);
119
120            ret.plot_handle(3, :) = plot(1/86400:1/86400:1, ret.
                original_L2(:, i), 'g-');
121            ret.plot_handle(4, :) = plot(1/86400:1/86400:1, ret.
                restprofil_L2(:, i), 'g-', 'lineWidth', 1.5);
122            ret.plot_handle(5, :) = plot(1/86400:1/86400:1, ret.
                original_L3(:, i), 'm-');
123            ret.plot_handle(6, :) = plot(1/86400:1/86400:1, ret.
                restprofil_L3(:, i), 'm-', 'lineWidth', 1.5);
124            title('Leistungsverlauf des Restprofils')
125            ylabel('Leistung in W');
126            xlabel('Uhrzeit HH:MM');
127            xlabel = '';
128            for k=0:24
129                xlabel = strcat(xlabel, sprintf('%02d:00', k), '|');
130            end
131            xlabel = xlabel(1:end-1);
132            set(gca, 'xlim', [0 1]);
133            set(gca, 'xtick', 0:1/24:1, 'xticklabel', xlabel);
134            rotateticklabel(gca, 90);
135            moveLabel('x', 35, ret.figHandles(i), gca);
136            legend(ret.plot_handle([2 4 6]), {'L1', 'L2', 'L3'});
137            grid on;
138        end
139    otherwise
140        cycle_index = 1:count;
141        if (numel(cycle_index) < num_days)
142            num_days = numel(cycle_index);
143        end
144        for i=1:num_days

```

## B. MATLAB Quellcode

```
145     a = randi(numel(cycle_index),1);
146     cycles(i) = cycle_index(a);
147     cycle_index = [ cycle_index(1:a-1) cycle_index(a+1:end)];
148
149     ret.figHandles(i) = figure;
150     ret.plot_handle(1,:) = plot(1/86400:1/86400:1,ret.
        original_L1(:,cycle_index(a))','b-');
151     hold on;
152     ret.plot_handle(2,:) = plot(1/86400:1/86400:1,ret.
        restprofil_L1(:,cycle_index(a)),'b-','lineWidth',1.5);
153
154     ret.plot_handle(3,:) = plot(1/86400:1/86400:1,ret.
        original_L2(:,cycle_index(a)),'g-');
155     ret.plot_handle(4,:) = plot(1/86400:1/86400:1,ret.
        restprofil_L2(:,cycle_index(a)),'g-','lineWidth',1.5);
156     ret.plot_handle(5,:) = plot(1/86400:1/86400:1,ret.
        original_L3(:,cycle_index(a)),'m-');
157     ret.plot_handle(6,:) = plot(1/86400:1/86400:1,ret.
        restprofil_L3(:,cycle_index(a)),'m-','lineWidth',1.5);
158     title('Leistungsverlauf des Restprofils')
159     ylabel('Leistung in W');
160     xlabel('Uhrzeit HH:MM');
161     xlabels = '';
162     for k=0:24
163         xlabels = strcat(xlabel,sprintf('%02d:00',k),'|');
164     end
165     xlabels = xlabels(1:end-1);
166     set(gca,'xlim',[0 1]);
167     set(gca,'xtick',0:1/24:1,'xticklabel',xlabels);
168     rotateticklabel(gca,90);
169     moveLabel('x',35,ret.figHandles(i),gca);
170     legend(ret.plot_handle([2 4 6]),{'L1','L2','L3'});
171     grid on;
172     end
173     ret.restprofil_L1_plot = ret.restprofil_L1(:,cycles);
174     ret.original_L1_plot = ret.original_L1(:,cycles);
175     ret.restprofil_L2_plot = ret.restprofil_L2(:,cycles);
176     ret.original_L2_plot = ret.original_L2(:,cycles);
177     ret.restprofil_L3_plot = ret.restprofil_L3(:,cycles);
178     ret.original_L3_plot = ret.original_L3(:,cycles);
179     ret.cycles = cycles;
180     end %switch(cycle_count)
181
182     %x_label = 'Zeit in Sekunden';
183     %y_label = 'Leistung in W';
184     %comparison(1/86400:1/86400:1,[ret.original_L1_plot, ret.
        restprofil_L1_plot],legenden,titel,filename,type,x_label,y_label);
185     %ret.figHandles = gcf;
186     ret.analysis = 'show_restprofil';
187     ret.hh = hh; ret.weekdays = weekdays;
188     ret.sommer = sommer; ret.winter = winter;
189
```

## B. MATLAB Quellcode

```
190
191 else
192     ret = false;
193     return;
194 end %if(count > 0)
```