Anwendung von KI und ML zur Messdatenverarbeitung aus Energieversorgungskomponenten

Bachelorarbeit

Sergej Lamert - 727245
Fakultät angewandte Informatik

15. Februar 2021

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Buck Wandler sind heutzutage in aller Munde, sie stellen beispielsweise Spannung in Notebook Prozessoren und Ladegeräten zu verfügung, Regeln den Strom an Stepper-Motoren. Durch die weitläufigen Anwendungsfälle, ist es besonders in der heutigen Zeit, in der Künstliche Intelligenz und Machine Learning Algorithmen immer wichtiger werden, zu evaluieren, inwieweit sich solche Wandler für Machine Learning Anwendungen eignen, wenn Daten über Strom, Spannung, Leistung und Temperatur bekannt sind. Deshalb ist das Thema dieser Arbeit die Analyse eines Hybrid-Buck-Wandlers. Im Folgenden wird solch ein Wandler hinsichtlich seiner Anwendbarkeit für Themen im Bereich Maschine Learning und Zustandsüberwachung analysiert werden. Die Analyse erfolgt dabei in mehreren Schritten. Zuerst werden Daten erhoben, während der Wandler an unterschiedliche statische Lasten angeschlossen ist, um einen Überblick über das Verhalten des Wandlers zu gewinnen. Anschließend wird der Wandler an eine dynamische Last angeschlossen und es werden periodische Signale gemessen. Die gemessenen Daten werden dann in Echtzeit mit bereits bekannten Daten verglichen und ebenfalls gegen andere, parallel laufende Wandler, abgestimmt. Das Letztendliche Ziel dieser Arbeit ist es, zu evaluieren, wie präzise und zuverlässig die Daten sind und inwieweit diese sich für Anwendungen in den am Anfang genannten Bereichen eignen. Da der vorliegende Wandler über einen Mikrocontroller mit I2C Schnittstelle verfügt, aus welchem Strom, Spannungs und Temperaturdaten erhoben werden können, wird das Erfassen der Daten durch das vorhandene Protokoll stark vereinfacht und sehr portierbar gemacht, was es ermöglicht die Daten aus beliebigen Platformen (Win 10, Linux, etc.) oder einfach durch andere Mikrocontroller (z.B. Raspberry Pi) auszulesen.

2 Theoretischer Hintergrund

Dadurch bedingt, das diese Arbeit sich an vielen Konzepten aus der Signaltheorie, Elektrotechnik und Informatik bedient, ist es sinnvoll vorab einige Begriffe und Konzepte zu definieren und zu erläutern. Aus diesem Grund, werden im Folgenden das I²C Protokoll, die Grundlagen von Neuronalen Netzen sowie die verwendete Software näher erläutert.

2.1 Hybrid-Buck-Wandler

Bei dem Buck-Wandler handelt es sich um einen DC-DC Wandler, d. h. dieser nimmt eine Gleichstrom Eingangsspannung von 15 bis 40 Volt entgegen und setzt diese herunter auf eine Gleichstrom Ausgangsspannung von 12 Volt. Des Weiteren ist der hier verwendete Wandler auf einen Stromfluss von 10 Ampere limitiert, was bedeutet, dass die maximale Leistung am Ausgang des Wandlers 120 Watt beträgt. Es bedeutet ebenfalls, dass, wenn der Strom 10 A übersteigt, die Ausgangsspannung gedrosselt wird. Der Wandler besitzt ebenfalls eine I²C Schnittstelle, aus welcher Ausgangsspannung, Eingangsspannung, Ausgangsstrom und Temperatur des Boards gemessen und ausgelesen werden können. Die Datenleitungen des I²C Busses sind bereits auf dem Wandler selbst mit Pull-Up Widerständen ausgestattet, wodurch diese nicht in externen Schaltungen realisiert werden müssen.

2.2 I2C

I2C steht für Inter-Integrated Circuit und wurde von Philips Semiconductors 1982 entwickelt. Es handelt sich dabei um einen seriellen Datenbus im Master-Slave Stil. Dieses Protokoll setzt auf zwei Leitungen für den Datenaustausch, davon ist ein Kanal explizit für die Daten vorgesehen und der andere Kanal für den Takt. Der Takt beträgt im klassischen 100 KHz oder 400 KHz.

Es ist in Abb. 1 zu erkennen, dass es eine Master- und mehrere Slave Komponenten gibt. Der Master gibt allen Slaves vor, was sie zu senden haben, und wie schnell sie es tun sollen. Des Weiteren ist zu erkennen, das die beiden Leitungen SDA und SCL an einem Pull-Up Widerstand angeschlossen sind. Dieser dient dazu, die Leitungen, wenn weder Master noch Slave sendet, auf die Versorgungsspannung zu schalten, sodass keine undefinierten Zustände entstehen und die Leitung im unbenutzten Zustand eine logische 1 besitzt. Die Kommunikation erfolgt durch Adressen, so besitzt jeder Sensor eine I2C Slave Adresse, über die ein Master auf diese zugreifen kann. Das auslesen der Daten eines Slaves erfolgt per Registeradressen, so hat beispielsweise ein beliebiger I²C fähiger Sensor ein Register, in dem bestimmte Werte gespeichert sind. In dieser Arbeit handelt es sich bei dem I2C Slave um einen Mikrocontroller auf dem Buck-Wandler. Der Master ist dabei ein

Raspberry Pi, welcher direkt per I2C mit dem Netzteil verbunden werden kann.

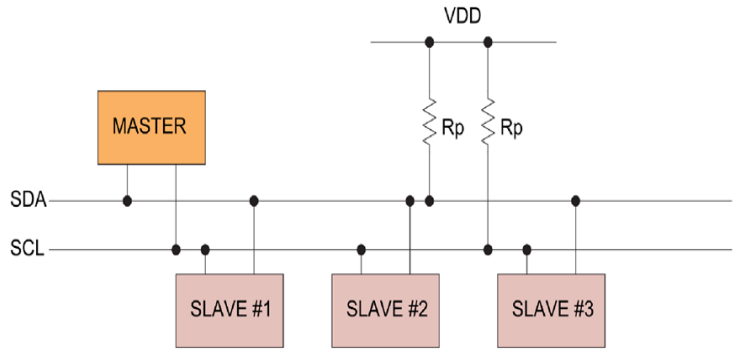

Abbildung 1: I2C Bus Beispiel Quelle: https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html

2.3 Neuronale Netze

Zur durchführung einiger simpler Testfälle unter den Punkten 3.2.1 und 3.2.2 ist es notwendig, eines der Grundlegenden Konzepte von Machine Learning zu erläutern: das Neuronale Netz. Neuronale Netze gehören zur Maschine Learning Kategorie des ßupervised learning". Beim supervised learning wird ein neuronales Netz anhand von Eingabedaten (Input) und bereits definierten Lösungen (Output) daraufhin optimiert, bei gegebenen Input eine passende Lösungsstrategie für den bereits definierten Output zu finden.

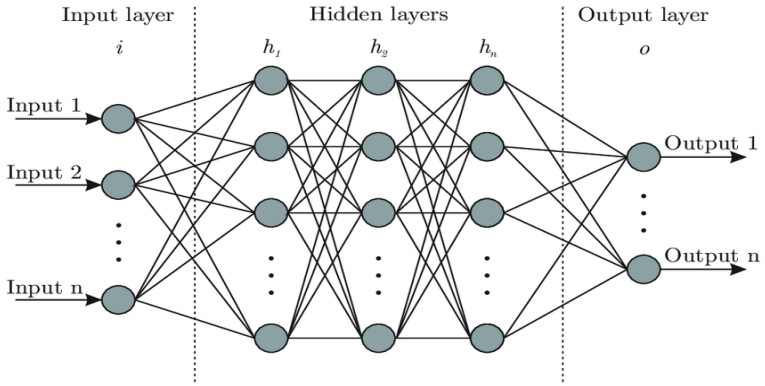

Abbildung 2: Beispiel eines Neuronalen Netzes: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o$_fig1_321259051$

In Abb. 2 erkennt man eine Abbildung der Architektur eines möglichen

neuronalen Netzes. Dieses ist aufgeteilt in mehrere Schichten: Input layer,hidden layer, Output layer. Das Input layer nimmt die Daten auf, welche verarbeitet werden müssen. Diese Daten können von verschiedener Art sein, so können es z. B. Pixeldaten eins Bildes sein, oder der Zeitverlauf von einem Signal. Im hidden layer werden die Daten durch diverse mathematische Operationen verarbeitet, um im Output layer Aussagen über die Input Daten machen zu können, d. h. um die Input Daten zu klassifizieren. Neben den einzelnen Schichten ist zu erkennen, dass es mehrere Knoten gibt, welche untereinander durch Kanten vollvermascht sind. Die Knoten werden im Allgemeinen "Neuronen"genannt und die Kanten sind die sog. "weightsöder Gewichtungen. Die Gewichtungen sind dabei die Hauptparameter eines Neuronalen Netzes. Bei Neuronalen Netzen wird zwischen zwei Algorithmen unterschieden, dem Feedforward Algorithmus, welcher bei gegebenen Input Daten und Gewichtungen einen bestimmten Output liefert. Und dem Backpropagation Algorithmus, welcher bei gegebenem Input und Output neue Gewichtungen für das Neuronale Netz berechnet. Im Folgenden werden beide Algorithmen näher erläutert.

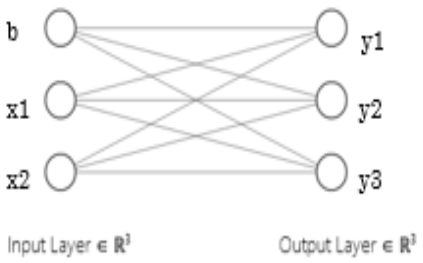

Abbildung 3: Konzept des Feedforward Algorithmus

$$\mathrm{A}\left(\begin{bmatrix} w_{10} w_{11} \\ w_{20} w_{21} \\ w_{30} w_{31} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Abb. 3 zeigt ein vereinfachtes Neuronales Netzt mit zwei Schichten. Das Input layer nimmt zwei Eingangswerte entgegen und verrechnet diese per Matrixmultiplikation mit den Gewichtungen, wie in Gleichung X dargestellt. Nach dieser Matrixmultiplikation entsteht ein neuer Vektor mit der gleichen Dimension wie der Output layer. Zu diesem Vektor wird noch der sog. "Biasäddiert. Der Bias dient als feste Skalierung dazu, ein Neuron gezielt mehr, bzw. weniger Aktiv zu machen, das bedeutet mathematisch, dass der numerische Wert gezielt gesteigert bzw. verringert wird. Nachdem der Input mit den Gewichtungen und dem Bias verrechnet wurde, wird auf das Ergebnis eine Aktivierungsfunktion angewendet. Eine Aktivierungsfunktion dient

dazu zu bestimmen, wie äktivëin Neuron ist. Es existieren mehrere Aktivierungsfunktionen. So zeigen Abbildungen 4 und 5 zwei Mögliche Funktionen.

Die Formel für die Sigmoid Funktion:

$$[H]s(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

In Abb. 4 ist eine der beliebtesten Aktivierungsfunktionen zu sehen, die ReLu Funktion. Wenn der Input der Funktion ¡= 0 ist, dann ist der Output immer 0. Für Werte ¿ 0, ist der Output immer die Identitätsfunktion. Diese Funktion hat den Vorteil, dass sie negative Werte ignoriert, und positive Werte einfach durchlässt. Des Weiteren ist zu erkennen, dass diese Funktion an der Stelle x = 0 nicht differenzierbar ist, da dort ein Knick vorzufinden ist.

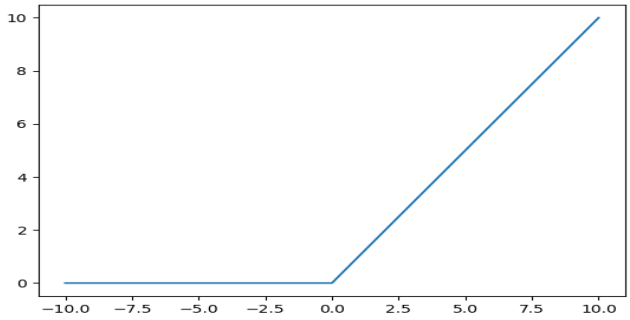

Abbildung 4: Rectified Linear Unit

$$y = \begin{cases} 0 & \text{if } x <= 0 \\ x & \text{if } x > 0 \end{cases}$$

In Abb. 5 ist die Sigmoid Funktion zu erkennen, diese lässt sowohl positive als auch negative Wert durch. Sigmoid ist im Vergleich zu ReLu jedoch an allen Stellen differenzierbar.

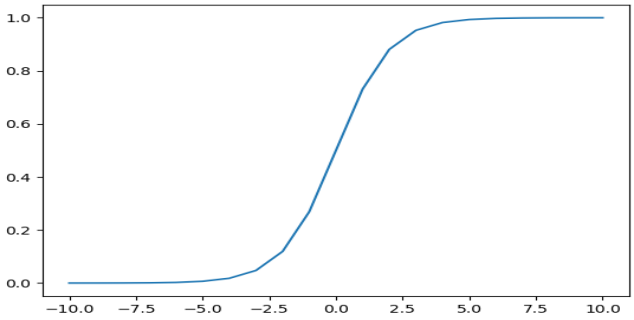

Abbildung 5: Sigmoid Funktion

2.4 Verwendeten Platformen, Programmiersprachen und Bibliotheken

Da sich herausgestellt hat, dass ein PicKit eine zu geringe Abtastrate ermöglicht, wurde für die dynamischen Tests ein Raspberry Pi 4 (4 GB RAM) mit der Programmiersprache Python für Datenerfassung- und Verarbeitung verwendet. Des Weiteren wurde für das Erstellen von Neuronalen Netzen die library "Tensorflow"verwendet, wobei Keras als High-Level API dient. Die graphische Visualisierung der Daten erfolgt ausschließlich per Pyplot library.

3 Methodik

3.1 Statische Datenerfassung

Die statische Datenerfassung dient dem Zweck, die Funktionalität der Wandler unter einer konstanten Last zu überprüfen. Es wurden zwei simple Testfälle aufgebaut. Bei beiden handelt es sich um Parallelschaltungen von 4,7 Ohm Widerständen. Der erste Fall schaltet zwei davon parallel, der zweite drei. Durch das Anwenden des Ohmschen Gesetzes, kann ein theoretischer Stromfluss und somit auch die theoretische Leistungsaufnahme berechnet werden. Der Wandler wird durch ein Netzteil mit 24 Volt Ausgangsspannung betrieben. Diese 24 Volt werden von dem Wandler auf 12 V herabgesetzt. Somit kann der Strom berechnet werden, der durch die Widerstände fließt. Bei zwei parallel-geschaltenen Widerständen ergibt das einen theoretischen Stromfluss von 5.1 A und bei drei parallel geschaltenen Widerständen 7.65 A.

$$\frac{U}{R} = I \tag{2}$$

$$\frac{12V}{4.7Ohm} * 2 = 5.10A \tag{3}$$

$$frac12V4.7Ohm * 3 = 7.65A \tag{4}$$

Da es sich in diesen Testfällen um statische Lasten handelt, sind keine hohen Abtastraten erforderlich, deshalb wird zur Datenerfassung ein PicKit mitsamt I²C Schnittstelle verwendet und kein Raspberry Pi.

Des Weiteren ist zu erwähnen, dass mehrere Wandler getestet wurden, einer davon mit welcher fehler. Bei der Messung der Daten, zeigt sich, dass sowohl Temperatur als auch Ausgangsstrom, dieses Wandlers unter gleicher Last höher waren als die der anderen, was man in Abb. X erkennen kann

3.2 Dynamische Datenerfassung

Das Ziel der dynamischen Datenerfassung ist es, zu evaluieren, wie der Wandler sich unter Lasten verhält, die sich mit der Zeit ändern. Sowie mögliche Abweichungen zwischen den einzelnen Wandlern selbst. Der Testaufbau beinhaltet den Hybridwandler, einen Stereoverstärker, welcher per Wandler mit Strom versorgt wird und in den ein Audiosignal eingespeist wird. In Abb. 5 ist die Schaltung des Testaufbaus abgebildet. Es ist zu erkennen, das am Eingang des Buck-Wandlers 24 V eingespeist werden und das die am Ausgang liegenden 12 V an den Stereoverstärker angeschlossen sind. Des Weiteren ist ersichtlich, dass nur ein Kanal des Stereoverstärkers genutzt wird. Dieser Kanal wird mit einem Audiosignal aus einem externen Rechner versorgt. Als Signale werden hierbei Sinussignale mit variierender

Frequenz verwendet, welche mit dem Programm Audacity generiert werden (https://www.audacityteam.org/). Das Ausgangssignal fällt über einen Lastwiderstand ab. Obwohl es ein Stereoverstärker ist, wurde auf einen Lautsprecher verzichtet, da durch einen Lastwiderstand die Reproduzierbarkeit gewährleistet wird.

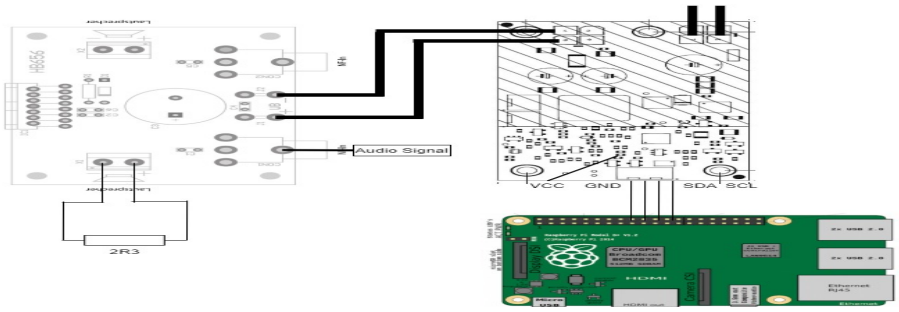

Abbildung 6: Versuchsaufbau der dynamischen Datenerfassung

Der erste Testschritt ist dabei die Überprüfung, wie hoch die maximale Frequenz eines Signals sein kann, sodass interpretierbare Daten erzeugt werden können. Die maximale Frequenz kann durch praktische Tests empirisch bestimmt werden. Der Umfang dieser Tests besteht darin, die Daten von mehreren Wandlern über einen Zeitraum T zu messen und anschließend die Anzahl der gemessenen Samples durch die gemessene Zeit zu teilen, um so die durchschnittliche Sample-Rate pro Sekunde zu erhalten.

Dadurch, dass die Datenerfassung auf einem Raspberry Pi abläuft, ist es möglich, die Daten nach jeder Abfrage sofort in eine Textdatei abzuspeichern. Es werden Daten zur Zeit, Temperatur, Ausgangsstrom, Ausgangsspannung und Eingangsspannung aufgezeichnet und tabellarisch abgespeichert.

$$f_{\text{nyquist}} = 1/2 * f_{\text{abtastrate}} \tag{5}$$

Die Nyquist-Frequenz stammt aus der Signaltheorie und setzt die maximale Abtastrate in Verhältnis mit der maximalen Frequenz, die ohne Verzerrungen dargestellt werden kann. Die Gleichung besagt, dass die maximale Frequenz, die ohne Verzerrungen dargestellt werden soll, mit einer Abtastrate abgetastet werden soll, die mindestens doppelt so hoch ist, wie die abgetastete Frequenz. Sobald die Abtastrate experimentell bestimmt werden kann, kann das Nyquist-Frequenz theorem darauf angewendet werden, um so die Frequenz zu bestimmen, die maximal mit dem Wandler abgetastet werden kann.

3.2.1 Echtzeitmonitoring

Das Ziel des Echtzeitmonitoring ist es, in laufenden Betrieb Aussagen darüber zu treffen welche Last gerade an dem Wandler angeschlossen ist. Eine Last ist in diesem Fall ein Audiosignal, welches durch einen Stereo Verstärker, welcher an den Wandler angeschlossen ist, verstärkt wird. Zur Klassifizierung, welche Last, bzw. welches Signal gerade am Stereoverstärker angeschlossen ist, wird ein neuronales Netz verwendet, welches per Tensorflow API generiert wird. Die gemessenen Signale werden dann vom Raspberry Pi, zwecks Verminderung der Rechenleistung, per Socket an einen Rechner geschickt, welcher dann die Daten per Neuronales Netz interpretiert und eine entsprechende Klassifizierung berechnet. Als Testfälle wurden Sinussignal mit verschiedenen Frequenzen verwendet, um die allgemeine Anwendbarkeit von Deep-Learning Algorithmen zu verifizieren.

3.2.2 Vergleichsmonitoring mehrerer Wandler im laufenden Betrieb

Das Ziel des Vergleichsmonitoring ist es, eine Software Struktur aufzubauen, welche es ermöglicht, mehrere Wandler gleichzeitig auf einem Raspberry Pi per I^2C zu betreiben und die Daten in Echtzeit zu visualisieren und in Textdateien abzuspeichern. Dadurch, dass die verwendeten Wandler unterschiedlich Adressen besitzen, müssen keine Ergänzungen an der Schaltung durchgeführt werden, es reicht einzig die Datenleitungen (SDA, SCL) zusammenzuschalten. Während der Messung der Wandler, werden die Daten von jedem Messpunkt in Echtzeit auf ein Koordinatensystem per PyPlot übertragen.

3.3 Anwendbarkeitstest - Neuronales Netz

Das Ziel des Anwenbarkeitstest ist es zu evaluieren, ob und in welchem Ausmaß sich die vom Wandler generierten Daten sich für Anwendungsfälle im Bereich des Maschine Learning, im speziellen dem der Neuronalen Netze, eignen. Hierfür wird ein simpler Testfall generiert, in dem die Daten mehrerer Signale mit verschiedenen Frequenzen als Input-Daten dienen und der Output entsprechend eine Klassifizierung der Signalfrequenz herausgeben soll.

4 Ergebnisse

4.1 Ergebnisse der statischen Tests

Im Folgenden werden die Ergebnisse der statischen Tests mit Bezug auf Temperatur- und Leistungsverlauf mehrerer Wandler dargestellt und näher erläutert. Der Versuchsaufbau und die Testumgebung waren bei allen drei Wandlern nahezu identisch. In Abb. 10 ist der Temperaturverlauf [in °C] von drei unterschiedlichen Wandlern über einen Zeitverlauf von 3600 Sekunden aufgetragen. Es ist zu erkennen, dass der Wandler mit der Bezeichnung 'Board_10' eine wesentlich höhere Einschwingtemperatur besitzt als die anderen zwei. Dies lässt sich darauf zurückführen, dass besagter Wandler einen Hardware defekt besitzt. Des Weiteren ist zu erkennen, dass die beiden verbleibenden Wandler trotz identischer Hardware und gleichen Testbedingungen unterschiedliche Temperaturen vorweisen. Somit kann behauptet werden, das unter gleichen Bedingungen Abweichende Resultate herauskommen

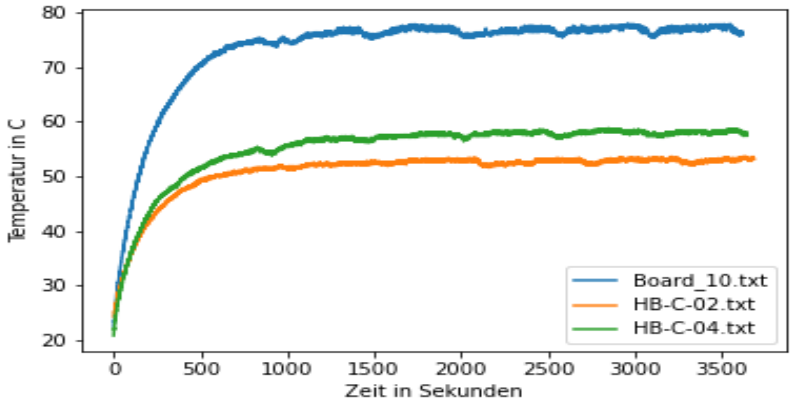

Abbildung 7: Temperaturvergleich mehrerer Wandler

In Abb. x ist der Leistungsverlauf [in Watt] der drei Wandler geplotet gegen die [Zeit in Sekunden] zu sehen. Aus diesem Graphen ist sofort ersichtlich, dass alle Signale rauschen. Wie man Tabelle X entnehmen kann, beträgt die Standardabweichung vom Mittelwert bei allen Signalen Un 300 Milliwatt. Die theoretische Leistung, welche bei einer Spannung von 12 V und zwei Parallel geschalteten Widerständen (siehe 3.1) generiert werden müsste, beträgt 61.2 Watt:

$$[H]12V * 5.1A = 61.2W \tag{6}$$

Somit liegt der Wandler mit der Bezeichnung "HB-C-02äm nächsten zum theoretisch berechneten Wert. Diese praktische Abweichung kann durch einen Leistungsabfall in den Leitungskabeln und der Hardware zumindest teilweise er werden.
Aufgrund dieser Resultate ist es für Maschine Learning Applikationen sinnvoll auf diese Daten einen Frequenzfilter anzuwenden, um das Rauschen zu eliminieren, da ansonsten Inkonsistenzen bei den K.I Applikationen vorkommen können, da die Daten unter gleichen Bedingungen variieren.

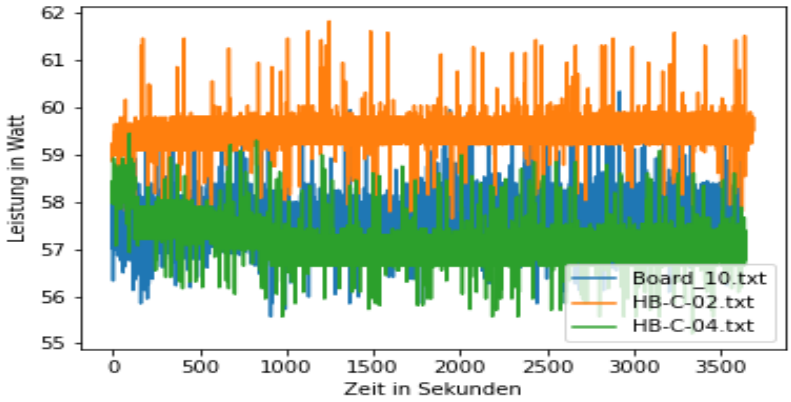

Abbildung 8: Leistungsvergleich mehrerer Wandler

In Abb. x ist der Leistungsverlauf zu erkennen, auf den ein Tiefpassfilter angewendet wurde. Bei dem Tiefpassfilter handelt es sich um Blablabla details. Dieser Datensatz besitzt nun kein Rauschen mehr, die einzelnen Signale sind jedoch trotzdem noch unterscheidbar.

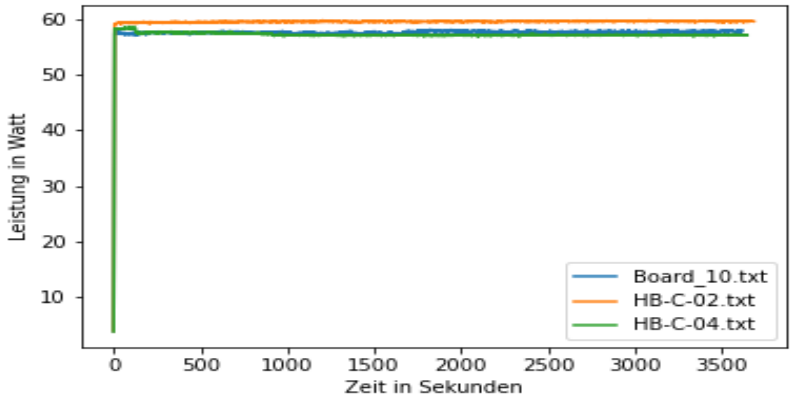

Abbildung 9: Gefilterter Leistungsverlauf der drei Wandler

4.2 Ergebnisse der dynamischen Tests

Zur dynamischen Datenerfassung ist zu sagen, dass das I²C Protokoll sich limitierend darauf ausgewirkt hat. Dadurch, dass stabile Programmausführung nur bei einer I²C Taktrate von 70 KHz gewährleistet war, war die Konsequenz daraus, dass die Abtastrate ebenfalls reduziert wurde. Die Ermittlung der Abtastrate erfolgte durch Messungen mehrerer Wandler, die mit verschiedenen Signalen gespeist wurden.Wie in Abb. 13 zu erkennen ist, wurden zwei verschiedene Wandler mit jeweils 6 verschiedenen Signalen verwendet. Zur Bestimmung der jeweiligen Abtastrate wurde zu durch Zeitstempel das Zeitdelta bestimmt, d. h. der Zeitraum in dem die Messungen durchgeführt wurden. Anschließend wurde die Anzahl der aufgenommenen Samples durch das Zeitdelta geteilt, um so auf die durchschnittliche Abtastrate zu kommen. Darauffolgend wurde der Mittelwert von allen Wandler-Signal Paaren berechnet, welcher 110 Hz beträgt. Wenn auf dieses Ergebnis die oben genannten Konzepte der Nyquist Frequenz angewendet werden, so zeigt sich, dass die maximale Signalfrequenz, welche ohne Verzerrungen durch den Wandler abgetastet werden kann, bei 55 Hz liegt. Aus diesem Grund wurden im nachfolgenden Test mit einem Neuronalen Netz nur Signalfrequenzen verwendet, welche einen geringere Frequenz als 55 Hz besitzen.

Board/Signal	Abtastrate in Hz
B1_S1	109
B1_S2	113
B1_S3	110
B1_S4	111
B1_S5	113
B1_S6	107
B2_S1	110
B2_S2	108
B2_S3	111
B2_S4	110
B2_S5	111
B2_S6	109
ø	110,1666667

Abbildung 10: Berechnung der Abtastrate

In Abb. 4 ist das generierte Signal zu sehen, welches in den Verstärker eingespeist wird. Dabei handelt es sich in diesem Beispiel um ein 10 Hz Sinussignal, welches in Audacity generiert wurde und als Signalinput für den Stereo-Verstärker dient. Dadurch, dass der Verstärker sowohl positive als auch negative Signale verstärkt, ergibt sich für die reine Leistungsaufnahme des Wandlers eine theoretische Leistungskurve wie sie in Abb. 11 dargestellt ist, nämlich der Betrag einer Sinusfunktion:

$$[H]f(x) = |sin(2 * pi * x * 10)| \tag{7}$$

Dadurch, dass es vorher ein Sinussignal mit einer Frequenz von 10 Hz war, ist es nun ein Sinussignal mit einer effektiven Frequenz von 20 Hz.
Das Messen der Daten erfolgte auf einem Raspberry Pi per I^2C Protokoll und einer eingestellten Taktrate von 70 KHz, da diese Stabilität gewährleistet. Die Datenerhebung und Verarbeitung erfolgte durch die Programmiersprache Python.

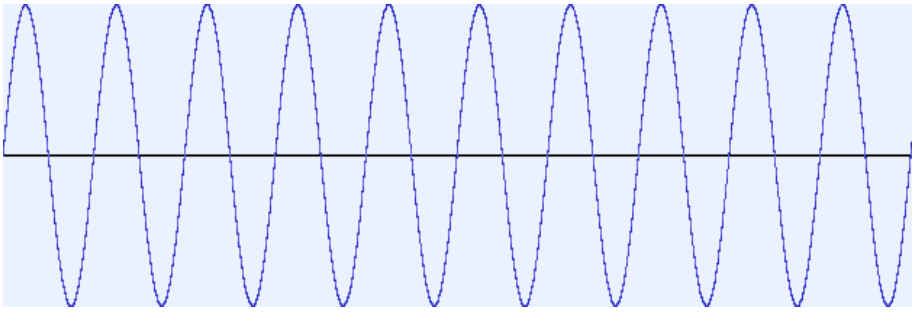

Abbildung 11: In Audacity generiertes 10 Hz Sinussignal

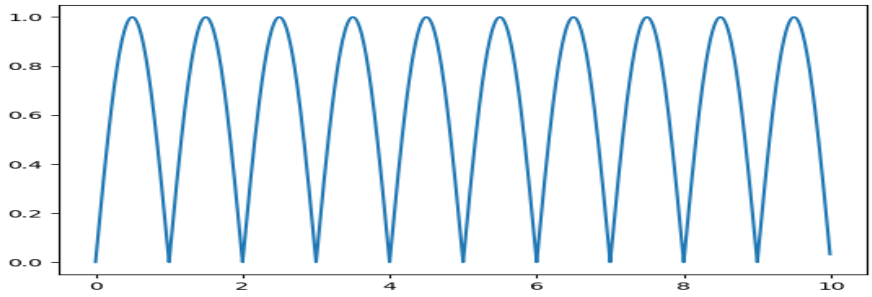

Abbildung 12: Effektive Frequenz eines 10 Hz Signals ist 20 Hz, weil, weil es für den Verstärker egal ist ob + oder - Signal

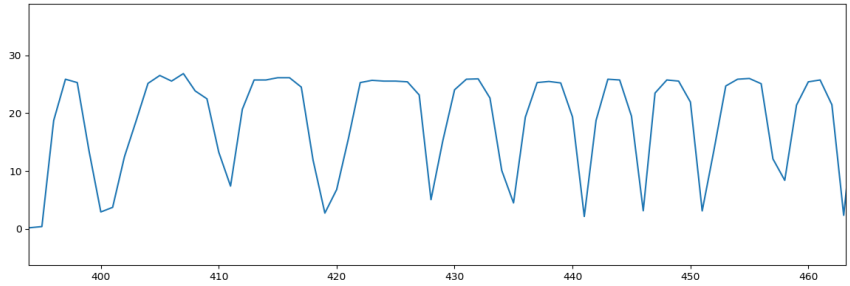

Abbildung 13: Tatsächlich gemessene Leistung des Wandlers

In Abb. 6 ist der ungefilterte Leistungsverlauf des Wandler dargestellt. Es ist ersichtlich, dass das umgeklappte 20 Hz Sinussignal in Abb. 5, dem gemessenen Sinussignal, entspricht. Der unsaubere Verlauf der Leistung ist auf Rauschen innerhalb der Hardware und auf die geringe Abtastrate von 110 Hz zurückzuführen.

4.3 Ergebnis des Anwendbarkeitstest - Neuronales Netz

Als Architektur des Neuronalen Netzes wurde ein Netz mit einem Input layer, zwei Hidden layern, sowie einem Output layer gewählt. Als Aktivierungsfunktion der Neuronen wurde ReLu ausgewählt, die Output Neuronen werden durch die softmax Funktion Aktiviert. Der verwendete Optimierungsalgorithmus ist Gradient Descent. Als Input Daten wurden 50 Samples des zeitlichen Leistungsverlaufs eines Signals verwendet. Die Ausgabe des Neuronalen Netzes ist eine Klassifizierung, um was für eine Signalfrequenz es sich bei den Input Daten handelt.
Abb. 13 zeigt die Genauigkeit des Neuronalen Netzes in der Vorhersage der verschiedenen Frequenzen anhand der Trainingsdaten. Es ist aus dem Graphen ersichtlich, dass die Genauigkeit über 1000 Trainingsepochen 99% beträgt.

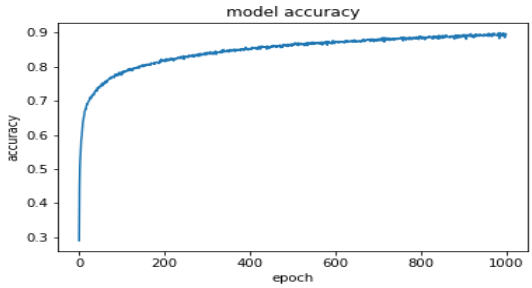

Abbildung 14: Genauigkeit der Anwendbarkeitstests

5 Literaturverzeichnis