Technische Universität Berlin
Fakulät IV
Institut für Energie und Automatisierungstechnik
Fachgebiet Elektronische Mess- und Diagnosetechnik

Praktikum Messdatenverarbeitung
Betreuer: José-Luis Bote-Garcia
SS2018

Versuchsprotokoll 6

Digitale Filterung II
Realisierung auf Mikrocontrollern

Serdar Gareayaghi (374183)
Ongun Türkcüoglu (371690)
Onur Akdemir (375959)
Marjan Chowdhury (344675)

5. Juli 2018

Inhaltsverzeichnis

1 Einleitung **1**

2 Praktikumsaufgaben **2**

2.1 Im Mikrocontroller implementierten FIR-Filter 2

2.2 Aufnahme mit analoges Filter 3

2.3 Aufnahme mit digitales Filter und Nachabtastung 4

2.4 Spektren der Signale und SNR 5

3 Abgabeaufgaben **7**

3.1 Implementierung der Funktion filterFIR 7

3.2 Aufgabe 2 7

1 Einleitung

In diesem Laborpraktkikum soll die Unterschiede zwischen analoges Filter und im Mikrocontroller implementiertes digitales Filter untersucht. Dafür wird in Matlab ein digitales FIR-Filter entworfen, denen Filterkoeffizienten für die Implementierung des Filters im Mikrocontroller benutzt werden. Um die Filterkoeffizienten in einer C-Header-Datei zu exportieren wird die Matlab-Funktion exportCoeff verwendet. Die Angabe des Filters folgt in dem Code MDV_PR.c mit die Änderung der Funktion filterIdentity() zur filterFIR(). Dazu wird eine C-Funktion filterFIRDecim in der Datei filter.c implementiert, die wie bei der Matlab-Funktion DecimFilt von Digitale Filterung 1 funktioniert. Zusäztlich wird die implementierte Dezimation von Mikrocontroller für die Nachabtastung verwendet.

2 Praktikumsaufgaben

2.1 Im Mikrocontroller implementierten FIR-Filter

In diesem Teil soll eine digitale Tiefpassfilterung, wie bei Digitale Filterung 1 entworfen aber dieses mal soll der Filter in dem Mikrocontroller implementiert, indem der Code in der Datei MDV_PR.c angepasst wird. Die Amplitudenfrequenzgang des im Mikrocontroller implementierten FIR-Filters wird im Bereich von 0 bis 15kHz aufgenommen und ist in der Abbildung 1 dargestellt. Bis ungefähr 1000 Hz ist keine Dämpfung zu sehen aber danach fängt eine Dämpfung von 40 dB bis 13kHz. Mit diesem integrierten digitalen Filter wird zwischen 13-14kHz nicht gut gedämpft, da in diesem Bereich die Dämpfung immer weniger wird und nach 14kHz gibt es eigentlich gar keine Dämpfung.

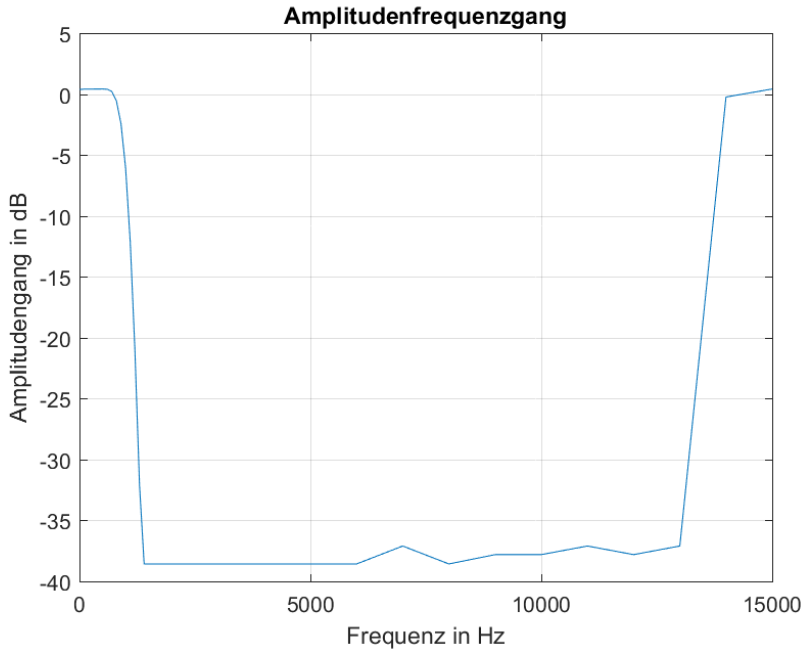

Abbildung 1: Amplitudengang des FIR-Filters

Im Vergleich zu einem analogen Filter hat der digitale Filter an der Frequenz 15kHz keine Dämpfung. Das liegt daran, dass der digitale Filter auch eine Abtastung durchführt, und damit auch mit Nyquist-Kriterium konform gehen muss. Dieser digitale Filter dämpft alle Frequenzanteilen kleiner also 1500 Hz, wobei die Frequenzanteile am 15kHz wieder nicht gedämpft sind.

2.2 Aufnahme mit analoges Filter

Mit dem Matlab-Skript generateTestSignal wird ein verrauschtes Sinussignal mit einer Frequenz von 440 Hz an dem Funktionsgenerator erzeugt. Das Signal wird ein mal mit einer Frequenz von 3 kHz und ein mal mit einer Frequenz von 15kHz abgetastet. Bei der Aufnahme wird die digitale Filterung nicht benutzt sondern das analoges Tiefpassfilter in der Wandler-Box verwendet. Die beide Aufnahmen werden für die gleiche Zeitdauer in der 2. Abbildung bzw. in der 3. Abbildung geplottet.

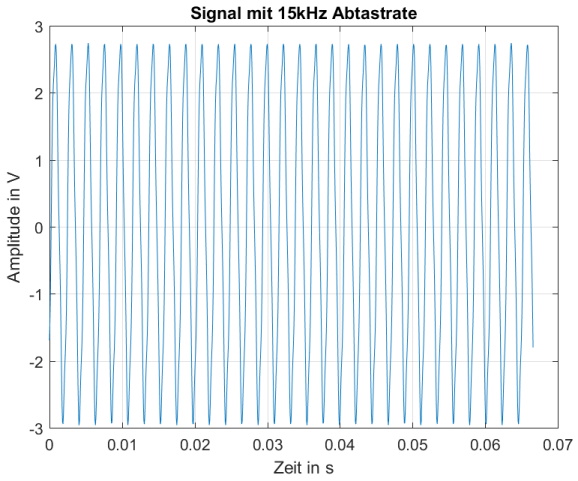

Abbildung 2: Aufnahne mit 15kHz Abtastrate

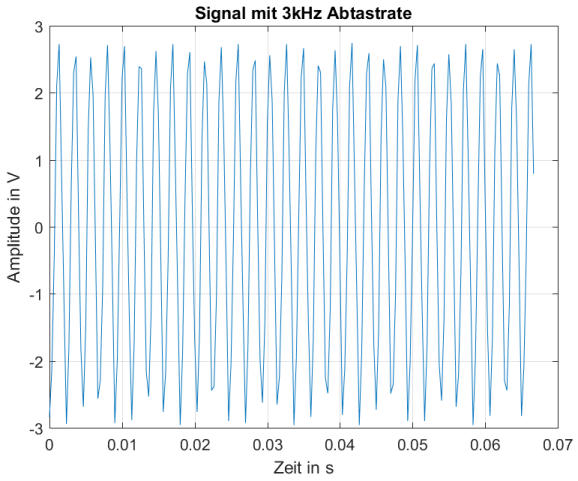

Abbildung 3: Aufnahne mit 3kHz Abtastrate

2.3 Aufnahme mit digitales Filter und Nachabtastung

Das gleiche Signal wie in Aufgabe 2 wird mit in dem Mikrocontroller implementierten digitalen Filter und mit einer Frequenz von 15kHz abgetastet. Bei der Aufnahme wird eine Nachabtastung auf 3kHz durchgeführt, indem den Parameter Decimation beim Aufruf der Funktion ucAnalogread auf 'ON' gesetzt wird. Die Parametern SampleRate und NumSamples werden so eingestellt, dass sie die Werte der Abtastrate bzw. der Sample-Anzahl nach der Dezimation haben.

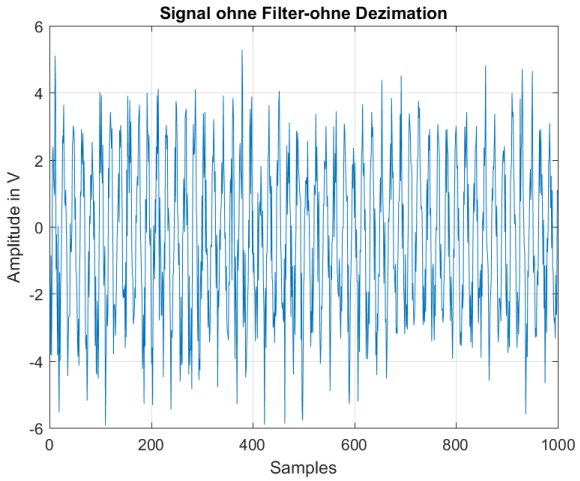

Abbildung 4: Ohne Filter, Ohne Nachabtastung

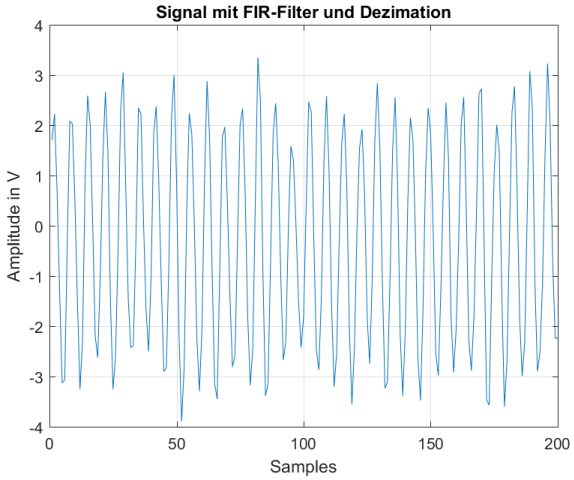

Abbildung 5: Mit digitales Filter und Nachabtasung

2.4 Spektren der Signale und SNR

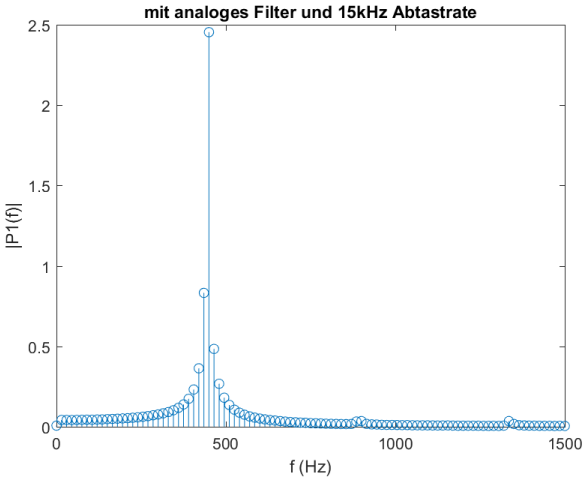

Abbildung 6: Spektrum des Signals mit analoges Filter und 15kHz Abtastrate

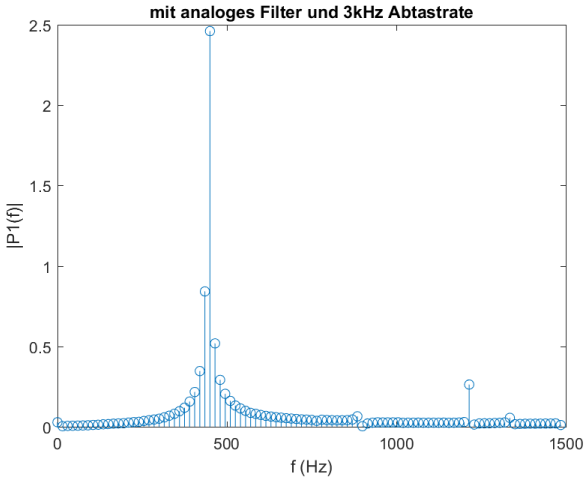

Abbildung 7: Spektrum des Signals mit analoges Filter und 3kHz Abtastrate

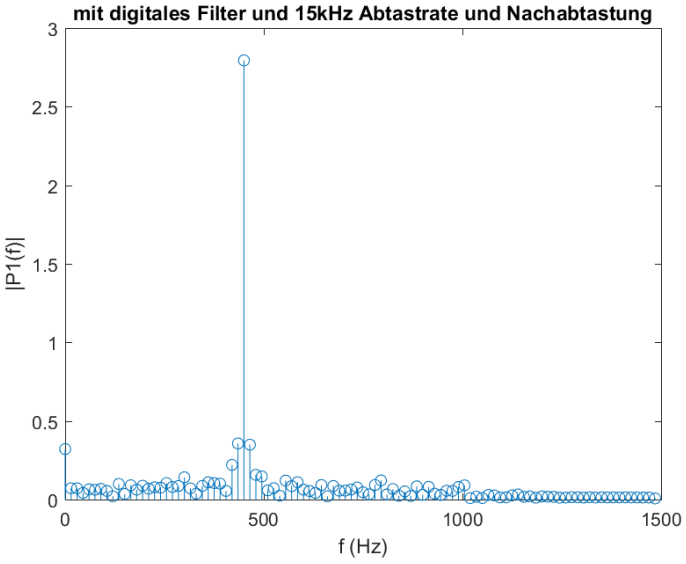

Abbildung 8: Spektrum des Signals mit digitales Filter und Nachabtastung

Das Signal mit analoges Filter und 15 kHz Abtastrate in Zeitbereich ist in der 2. Abbildung und das Signal mit analoges Filter und 3 kHz Abtastrate in Zeitbereich ist in der 3. Abbildung dargestellt. Das Signal mit digitales Filter und 15kHz Abtastrate mit einer Nachabtastung auf 3kHz ist in der Abbildung 5. Ein Vergleich zwischen die analog gefilterte Signale zeigt, dass das Signal mit einer Abtastrate von 3kHz nicht den idealen Sinusverlauf entspricht. Das Signal mit der 15kHz Abtastrate hat allgemein den idealen Sinusverlauf trotz des Rauschens also der Grund für den nicht-idealen Verlauf für das Signal mit der 3kHz Abtastrate nicht das Rauschen sondern die verminderte Abtastrate bzw. die wenigere Anzahl der Messwerte pro eine Periode der Sinuskurve. Für das Signal mit digitales Filter ist das auch derselben Fall in dem Zeitbereich wie bei dem Signal mit der 3kHz Abtastrate.

Die Amplitudenspektren in die Abbildungen 6 und 7 sehen sehr ähnlich zueinander aus aber das Signal mit 3kHz Abtastrate und analoge Filterung hat einen Sprung ungefähr auf einer Frequenz von 1200 Hz wegen Aliasing-Fehler. Das Amplitudenspektrum des Signals mit digitale Filterung und der Nachabtastung hat im Vergleich zu andere Spektren weniges Rauschen, dass man auch in der SNR sehen kann. Die SNR-Werte für die drei entsprechende Signale sind in der 1 zu sehen. Es gibt keine größe Unterschied zwischen dem Signale A1 und A2 aber A1 hat immer noch eine größere SNR wegen der größere Abtastrate bzw. eine größere Anzahl der Messwerte führt zu einem besseren Signal mit etwas wenigen Rauschen. Das Signal mit digitaler Filter und Dezimation hat aber eine wesentliche Unterschied bei dem SNR-Wert.

	A1	A2	A3
Abtastrate [kHz]	15	3	15
digitaler Filter	nein	nein	ja
Dezimation	nein	nein	ja
Signal - RMS [V]	1.7353	1.74	1.9763
Rauschen - RMS [V]	0.8628	0.8849	0.6637
SNR [dB]	6.0694	5.8730	9.4776

Tabelle 1: Effektivwert und SNR-Berechnungen für die Messdaten

Die unterschiedliche Signal-Rausch-Verhältnisse stimmen mit der Theorie überein. Der SNR-Wert für die Signalaufnahme mit 15kHz, 6.06 dB, verglichen mit dem SNR-Wert für die Signalaufnahme mit 3kHz, 5.87 dB, ist größer. Diese Tatsache ist durch die 5-fache Abtastfrequenz entstanden, denn durch die erhöhte Abtastfrequenz werden die Quantisierungsfehler über größere Frequenzen verteilt, welcher das Rauschen unterdrückt. Der Unterschied ist aber relativ klein, also in dem Beispiel verursacht eine 5-fache Abtastfrequenz eine Verbesserung von etwa 0.2 dB.

Der Unterschied zwischen analogem Filter und digitalem Filter mit Dezimation ist deutlich größer. Dies liegt daran, dass die analoge Filter von realen Bauelementen und deren Qualität abhängig sind. Jedes Bauelement in der analogen Filter verursacht thermisches Rauschen (wie Johnson-Rauschen), welcher überall das Signal verrauscht. Die Dezimation (Nachabtastung) hat jedoch keinen Einfluss auf dem SNR, wie wir schon in dem vorherigen Protokoll gezeigt haben.

3 Abgabeaufgaben

3.1 Implementierung der Funktion filterFIR

Die Funktion filterFIR() implementiert in dem Mikrocontroller den FIR-Filter mit den angegebenen Filterkoeffizienten. Das heißt, im vergleich zu dem vorherigen Praktikum, wird die FIR-Filterung in dem Mikrocontroller stattfinden. Die drei Parameter für die Funktion sind jeweils der Eingangswertpuffer (filtInBuf), die Filterordnung und der Eingangswert.

Die Implementierung des FIR-Filters ist im Grunde genommen eine Summe des Produkts von dem Eingangswert und dem Filterkoeffizient. Aus diesem Grund ist der 32-bit Integer äccumëine Summation der Multipilikation der i-ten Filterkoeffizient mit dem Eingangswert.

Wenn der Summation fertig ist, dann wird mittels fifoPopWord() der letzte Wert aus dem Puffer gelöscht.

Die Implementierung des FIR-Filters findet mittels Fixed-Point-Arithmetik statt, denn es ist im Mikrocontroller aufwendiger, mit dem eigentlichen Integer Daten zu verarbeiten. Jedoch ist die Umsetzung von Q0.15 (Fixed-Point mit 15 Nachkommastellen) nach Integer eindeutig: Der Fixed-Point wird um 15 Nachkommastellen nach rechts verschoben (right-shift operator) bzw. mit 2^15 multipliziert.

3.2 Aufgabe 2

◦ Die Aufgabe 2a wurde in dem Praktikumsteil 2.1 'Im Mikrocontroller implementierten FIR-Filter' bearbeitet.

◦ Die Aufgabe 2b wurde in dem Praktikumsteil 2.4 'Spektren der Signale und SNR' bearbeitet.