



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

Praktikum Messdatenverarbeitung
Betreuer: José-Luis Bote-Garcia
SS2018

Versuchsprotokoll 5

Digitale Filterung I FIR-Filter

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

28. Juni 2018

Inhaltsverzeichnis

1	Einleitung	1
2	Praktikumsaufgaben	2
2.1	Signal Aufnahme mit analoges Anti-Aliasing Filter	2
2.2	Nachabtastung des aufgenommenen Signals	4
2.3	Signal mit digitales Anti-Aliasing Filter	6
2.4	Matlab-Funktion DecimFilt	8
3	Abgabeaufgaben	11
3.1	Aufgabe 1	11
3.2	Aufgabe 2	11

1 Einleitung

In diesem Praktikum wird untersucht, wie die Abtastrate eines Signals durch Nachabtastung reduziert werden kann. Die Unterschiede zwischen dem aufgenommenen Signal und den dezimierten davon werden verglichen und die entsprechenden Unterschiede werden begründet. Zusätzlich wird ein digitales Filter bei der Nachabtastung verwendet und seine Wirkung wird erklärt.

2 Praktikumsaufgaben

2.1 Signal Aufnahme mit analoges Anti-Aliasing Filter

Mit dem Matlab-Skript `generateTestSignal()` wird ein verrauschtes Sinussignal mit einer Frequenz von 440 Hz an dem Funktionsgenerator erzeugt. Das Signal wird mit der Frequenz 15kHz abgetastet und dazu wird das Tiefpassfilter in der Wandler-Box als analoges Tiefpassfilter verwendet. Das Amplitudenspektrum ist in der 2. Abbildung zu sehen.

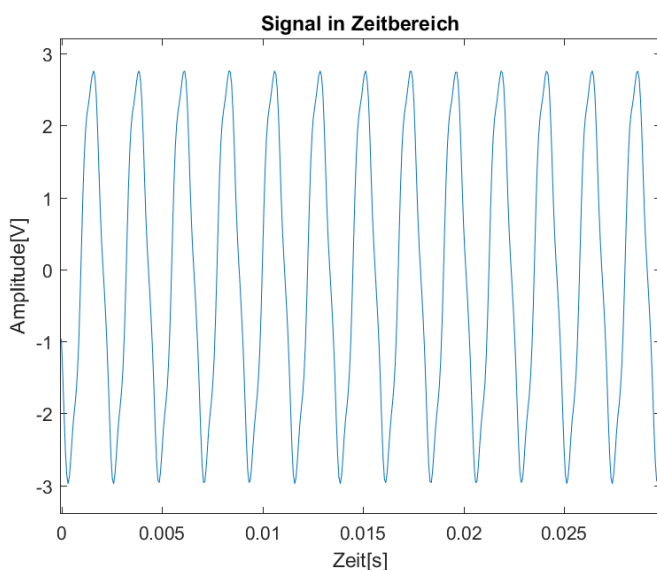


Abbildung 1: Aufgenommenen Signal im Zeitbereich

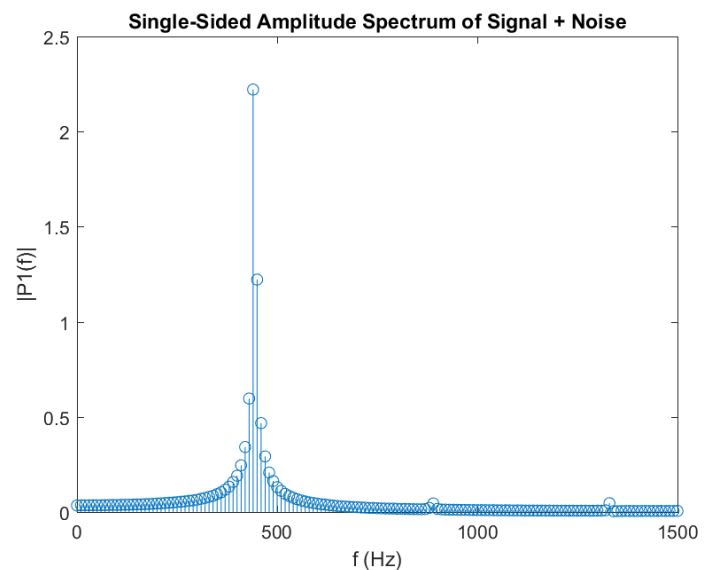


Abbildung 2: Amplitudenspektrum des aufgenommenen Signals

Bei der Effektivwertberechnung des Signals und des Rauschens im Frequenzbereich kann mit dem Parsevalschen Theorem gemacht werden. In den vorherigen Praktikas ist es gezeigt, dass mit dem Parsevalschen Theorem sind Effektivwertberechnungen im Zeitbereich und Frequenzbereich identisch. Jedoch ist die Entscheidung, im Frequenzbereich zu bleiben, nicht willkürlich, sondern liegt daran, dass die Signalfrequenzanteile (also in dem Fall nur 440 Hz) uns bekannt sind. Mit dem Wissen kann abgeleitet werden, dass alle andere Frequenzanteile Rauschen sind. Die Berechnung des Effektivwertes bzw. Signal-Rausch-Verhältnis ist dann nur eine Sache der Isolation des Signals und des Rauschens im Frequenzbereich:

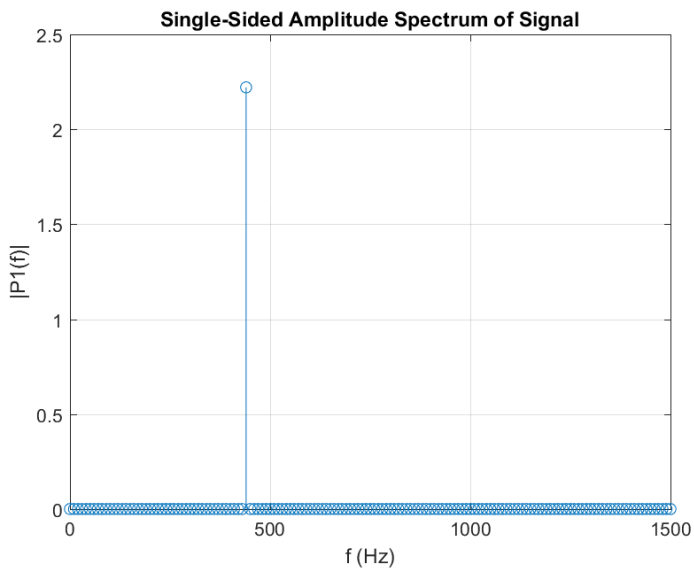


Abbildung 3: isoliertes Signal - Frequenzanteil 440 Hz

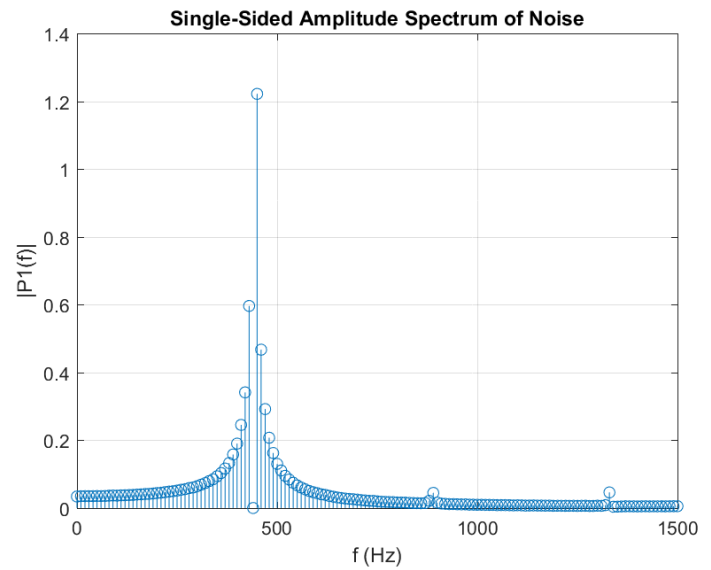


Abbildung 4: isoliertes Rauschen - Frequenzanteile außer 440 Hz

Die Abbildungen 3 und 4 zeigen jeweils das isolierte Signal und das isolierte Messrauschen im Frequenzbereich. Die Effektivwerte beider Messdaten sind als folgendes angegeben:

- Effektivwert des Signals: 1.9599 V
- Effektivwert des Rauschens: 1.1723 V

Das Signal-Rausch-Verhältnis ergibt sich dann als:

$$\text{SNR}_{|db} = 20 \cdot \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{rauschen}}} \right) = 4.46 \text{db}$$

Die Effektiv- und SNR-Werte für die restlichen Aufgaben sind auf der Tabelle 1 gezeigt.

2.2 Nachabtastung des aufgenommenen Signals

In diesem Teil wird eine Nachabtastung des aufgenommenen Signals mit 3kHz Abtastfrequenz durchgeführt. Das heißt, jeder fünfte Wert wird aufgenommen und die Rest wird verworfen. Das Amplitudenspektrum des nachabgetasteten Signals ist in der 5. Abbildung.

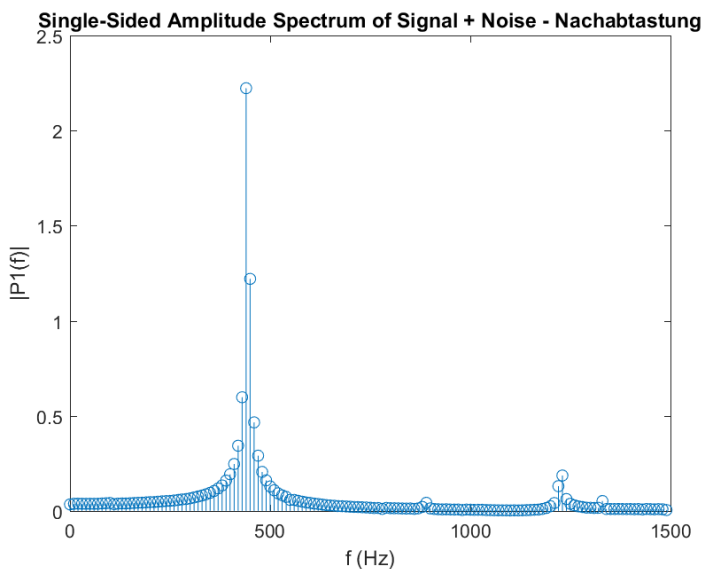


Abbildung 5: Amplitudenspektrum des nachabgetasteten Signals

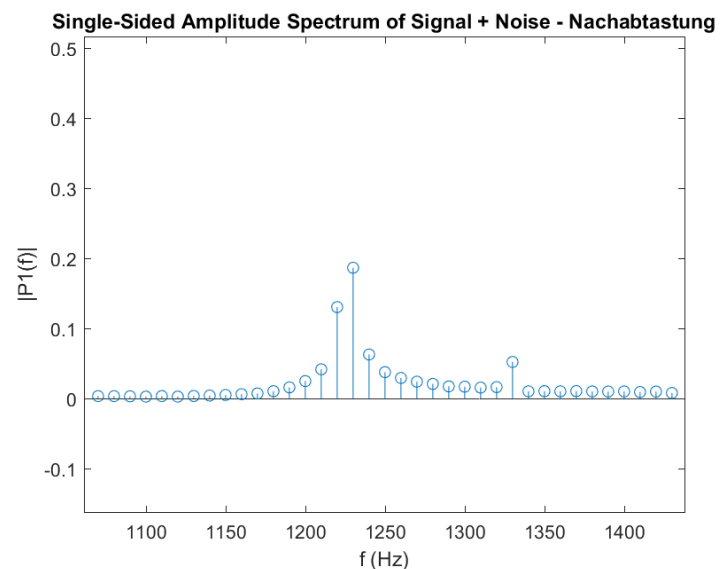


Abbildung 6: Bereich 1100 Hz - 1400 Hz

Allgemein sieht das Amplitudenspektrum des nachabgetasteten Signals fast wie das Signal in der 2. Abbildung aus. Der Unterschied liegt zwischen dem Frequenzbereich 1200-1250 Hz, wo bei dem nachabgetasteten Signal einige Sprünge gibt. Ein näherer Blick auf dem Frequenzbereich 1200-1250Hz ist in der Abbildung 6 geplottet. Genauer hinschauen zeigt, dass diese Sprünge wegen Aliasing-Effekte bei der Nachabtastung entstanden sind. Wenn der Graph des Amplitudenspektrums von der Abbildung 2 bis 2.5 kHz erweitert ist, wird der Upsprung diese Aliasing-Effekte gesehen. Ein Vergleich der Stelle und der Werte der entsprechenden Sprünge zwischen den Abbildungen 6 und 8 zeigt, dass die Sprünge zwischen der Frequenzbereich 1750-1800 Hz des Signals ohne Nachabtastung um dem Punkt 1500 Hz gespiegelt sind und bei dem Signal mit Nachabtastung als Aliasing-Fehler zwischen dem Frequenzbereich 1200-1250Hz entstanden sind.

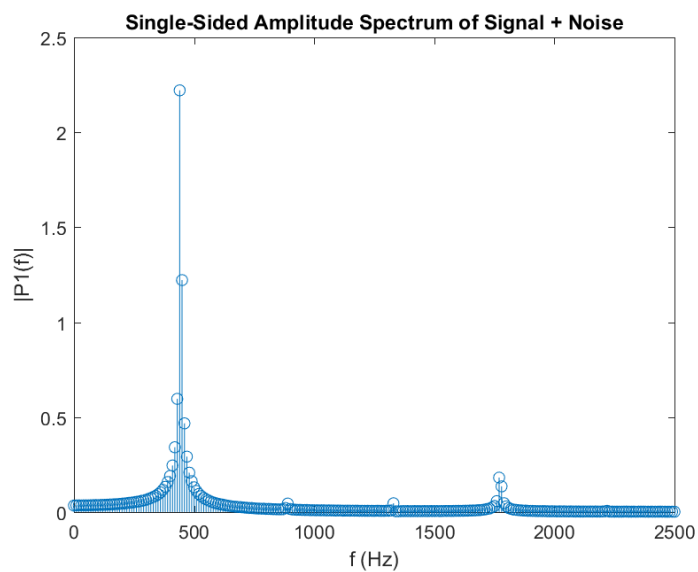


Abbildung 7: Spektrum vom Teil 2.1 bis 2.5kHz geplottet

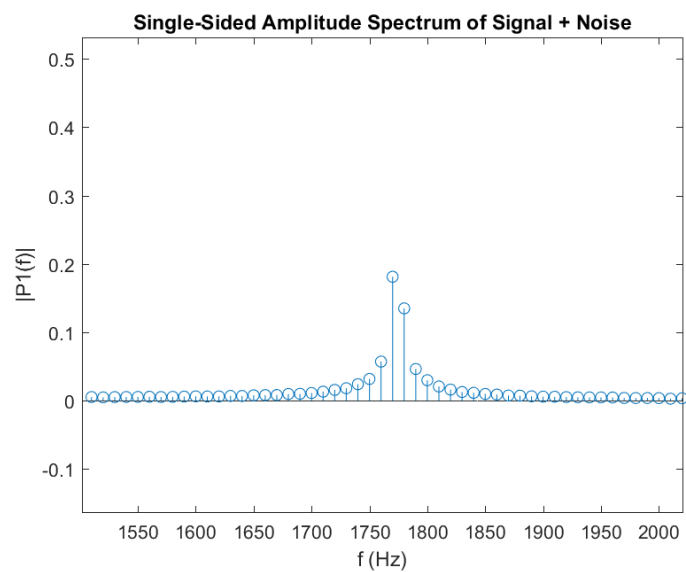


Abbildung 8:

2.3 Signal mit digitales Anti-Aliasing Filter

Ein gewünschter digitaler Filter wird mit dem Matlab-Function FIRentwurf entworfen und mit dem Matlab-Function FIRFilterung wird das Signal gefiltert. Bei der Entwurf des Filters wird die Grenzfrequenz 1070 Hz, die Abtastrate 15000 Hz eingestellt. Das Fenster folgt mit das ideale Fenstertyp-und Ordnung blackman(93). Vor der Nachabtastung sieht das gefilterte Signal im Zeitbereich wie in der Abbildung 9 aus. Am Anfang ist eine Zeitverzögerung zu erkennen, die wegen der FIR-Filterung vorgekommen ist. Das gefilterte Signal wird nachabgetastet und das Amplitudenspektrum wird in der Abbildung 10 dargestellt. Die Sprünge in der Abbildung 6 sind nicht mehr bei dem Spektrum des gefilterten Signals zu sehen. Die entsprechende Sprünge sind anhand der digitalen FIR-Filterung entfernt. Die Funktion FIRentwurf ist in der 1. Listing und FIRFilterung ist in der 2. Listing. Die Vorgehensweise für die digitale Filterung ist in der Listing 3. Die Funktion DecimFilt wird nur für die Nachabtastung benutzt. Die implementierte Filterungsfunktion von DecimFilt wird in dem Praktikumsteil 2.4 untersucht.

```
1 function g_w = FIRentwurf( fg, fs , wn )
2 %fs = Abtastfrequenz
3 %fg = Grenzfrequenz
4 %wn = Fensterfolge
5 n = length(wn);
6 g_w = zeros(1,n);
7 T = 1/fs;
8
9 tau = (n/2);
10
11 for k = 1:n %Filter entwurf laut der Formel 26
12     %in der Vorlesungsfolie
13     %'Kapitel 4: Zeitdiskrete Systeme und digitale
14     %filter'
15     g_w(k) = wn(k)* (sin(2*pi*fg*((k-tau)*T)))/(pi*((k-tau)*T));
16 end
17 sum_y = sum(g_w);
18 g_w = g_w./sum_y; %Normierung
19 end
```

Listing 1: Matlab-Function FIRentwurf

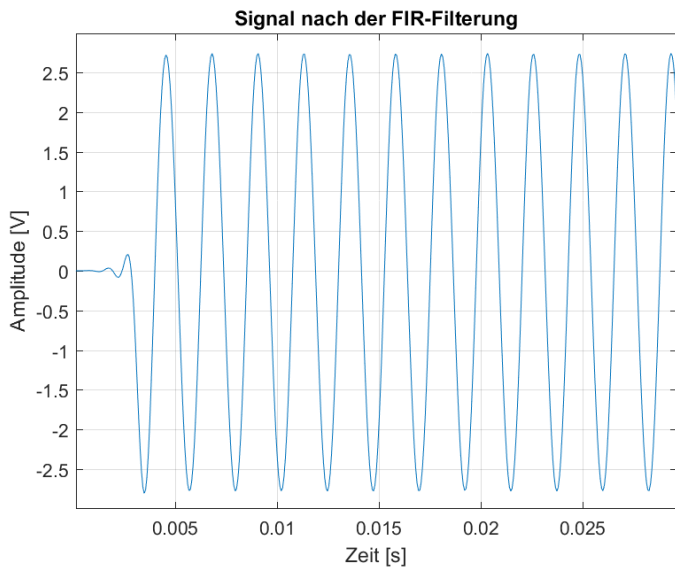


Abbildung 9: gefiltertes Signal im Zeitbereich

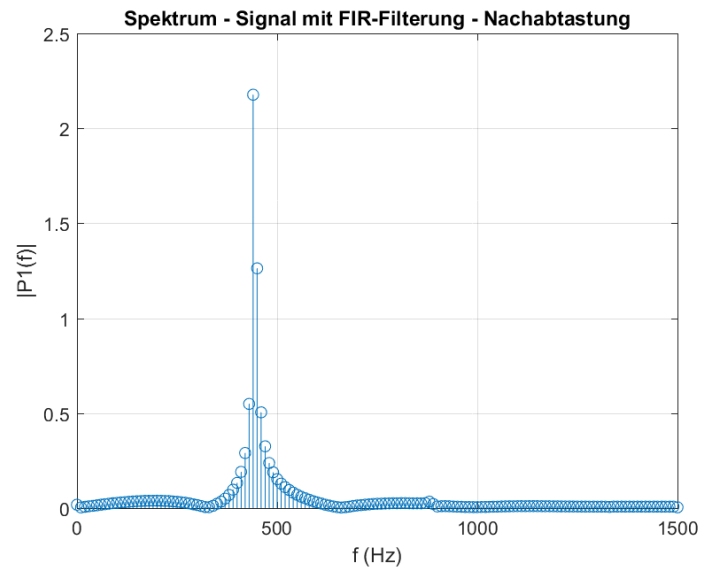


Abbildung 10: Spektrum des gefilterten Signals

```

1 function yk=FIRFilterung (bk , xk)
2
3 output = zeros(1,length(xk));
4 summe = 0;
5
6 for n = 1:length(xk) %Multipliziert folglich
7     %die Koeffizienten der Filter und Signal
8     for k = 1:length(bk)
9         if n > k
10            summe = bk(k)*xk(n-k+1) + summe;
11        end
12    end
13    output(n) = summe; %Jede Element in einem Vektor speichern
14    summe = 0; %Fuer den naechsten Element, die Summe
        zuruecksetzen
15 end
16
17 yk = output;
18 end

```

Listing 2: Matlab-Funktion FIRFilterung

```

1 %% Aufgabe 3
2 %Filterfunktion von DecimFilt ist nicht verwendet

```

```

3 %Decimfilt benutzt nur f?r Nachabtastung
4 fg_filter = 1070;
5 blackman_window = blackman(93);
6 digFilter = FIRentwurf(fg_filter,15000,blackman_window);
7 signal_with_Filter = FIRFilterung(digFilter,data_a1_volt);
8 nachabtast_signalfilt = DecimFilt(1,signal_with_Filter,5);

```

Listing 3: digitale Filterung mit Blackman-Fenster Länge 93

2.4 Matlab-Funktion DecimFilt

Die Funktion DecimFilt bekommt als Eingang der digitale Filter, das Signal und der Nachabtastungsfaktor, dann filtert das Signal anhand der Funktion FIRFilterung und führt eine Nachabtastung des Signals nach dem Nachabtastungsfaktor. Die Funktion ist in der Listing 4.

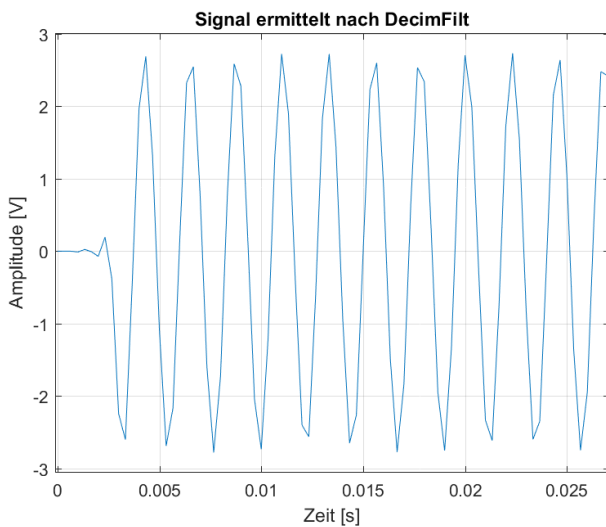


Abbildung 11: Signal im Zeitbereich

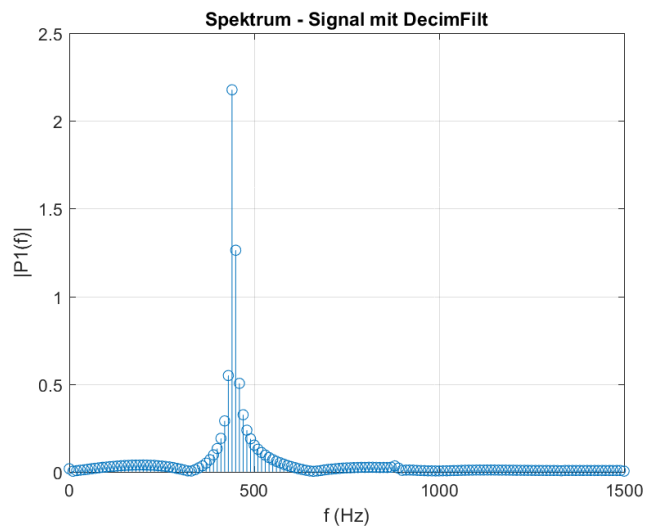


Abbildung 12: Spektrum des Signals

```
1 function yk=DecimFilt (bk , xk , fr )
2
3 %bk, Filter Koeffizienten
4 %xk, Signal
5 %fr, Nachabtastungsfaktor
6
7 n = length(xk);
8 yk_length = n/fr; %die Lange der nachabgetasteten Signals
9
10 yk = zeros(1,yk_length);
11
12 sigfilter = FIRFilterung(bk,xk); %Signal mit FIRFilterung filtern
13
14 for k = 1:yk_length %Nachabtastung, jede fr. Element speichern,
15                     %die Rest verwerfen
16
17 yk(k) = sigfilter(k*fr);
18
19 end
20
21 end
```

Listing 4: Matlab-Funktion DecimFilt

Wie in der Listing 5 zu sehen ist, wird ein gewünschter Filter mit FIRentwurf entworfen und dann als Eingang in DecimFilt eingesetzt. Das mit DecimFilt gefilterte und nachabgetastete Signal im Zeitbereich ist in der Abbildung 11 und dessen Amplitudenspektrum ist in der Abbildung 12 dargestellt. Im Vergleich zum Amplitudenspektrum in dem Praktikumsteil 3 in der 10. Abbildung keine Unterschiede sind zu sehen. Also die Funktion DecimFilt funktioniert fehlerfrei.

```
1 %% Aufgabe 4
2 fg_filter = 1070;
3 blackman_window = blackman(93);
4 digFilter = FIRentwurf(fg_filter,15000,blackman_window);
5 signal_decimfilt = DecimFilt(digFilter,data_a1_volt,5);
```

Listing 5: Verwendung von DecimFilt

	A1	A2	A3	A3
Abtastrate [kHz]	15	15	15	15
digitaler Filter	nein	nein	ja	ja
Dezimation	nein	5	nein	5
Signal - RMS [V]	1.9600	1.9605	1.9187	1.9194
Rauschen - RMS [V]	1.1723	1.1717	1.1430	1.1461
SNR [dB]	4.4641	4.4712	4.4994	4.4785

Tabelle 1: Effektivwert und SNR-Berechnungen für die Messdaten

Der Effektivwert des verrauschten Signals und des dazugehörigen Rauschens ist 1.96V bzw. 1.1723 V. Deswegen kann daraus abgeleitet werden, dass das Rauschen einen wesentlichen Anteil des Signals bildet. Trotzdem ist der eigentlichen Signalanteil (Frequenzanteil 440Hz) stärker als das Rauschen, ausgewiesen mit dem positiven SNR.

Eine Nachabtastung der Messdaten scheint keine Auswirkung auf die Effektivwerte zu haben. Das ist ein klarer Vorteil, denn obwohl nur jeden fünften Wert gespeichert und analysiert ist (im ersten Blick also Informationsverlust), hat dieser Prozess die eigentliche Information beibehalten. Der Vorteil liegt daran, dass die Dezimation effektiv die Messpunkten erniedrigt, und damit den Speicherplatz nicht unnötig erfüllt (weniger wichtig im PCs, mehr ein richtiger Vorteil für Mikrocontroller).

In der Theorie sollte eine digitale Filterung dazu führen, dass das Rauschen unterdrückt wird. Jedoch ist der Unterschied zwischen nicht unbedingt sehr groß, dass damit die Aussage unterstützt werden kann. Wie bei der Nachabtastung ohne Filterung, ändert sich hier tatsächlich auch relativ wenig Information, was auch zu erwarten ist.

3 Abgabearbeiten

3.1 Aufgabe 1

Auch wie in der 2. Protokoll erklärt, soll in einer digitalen Meskette, ein Anti Aliasing Tiefpassfilter vor dem ADU verschaltet sein. Wenn auf diesen analogen Anti-Aliasing-Filter verzichtet wird und nur digitale Filterung benutzt wird, wird vorher eine Digitalisierung des Signals gebraucht und damit da kein Aliasing entsteht, wird wieder ein analoges Anti-Aliasing Tiefpassfilter gebraucht.

Also die Antwort ist nein, um die Spiegelungsfehler (Aliasing) zu verhindern muss ein analoges Anti-Aliasing Tiefpassfilter benutzt werden.

3.2 Aufgabe 2

- Die Aufgaben 2a und 2b wurden in dem Praktikumsteil 2.2 'Nachabtastung des aufgenommenen Signals' bearbeitet.
- Die Aufgabe 2c wurde in dem Praktikumsteil 2.3 'Signal mit digitalem Anti-Aliasing Filter' bearbeitet.
- Die Aufgabe 2d wurde in dem Praktikumsteil 2.4 'Matlab-Funktion DecimFilt' bearbeitet.