

Inhaltsverzeichnis

1 Übung 1 - Abtastung von Signalen	4
1.1 Analyse der ISR	4
1.2 Untersuchung des Samplingprozesses	4
1.2.1 Aufgabenstellung	4
1.2.2 Schaltungsaufbau	4
1.2.3 Ausgangssignal	5
1.2.4 Spannungsauflösung	6
1.2.5 Samplingfrequenz	6
1.3 Untersuchung des Rekonstruktionsfilters	7
1.3.1 Aufgabenstellung	7
1.3.2 Ausgangssignal	7
1.3.3 Vergleich	7
1.4 Untersuchung mehrerer Frequenzen	8
1.4.1 Aufgabenstellung	8
1.4.2 2kHz	8
1.4.3 5kHz	9
1.4.4 8kHz	10
1.4.5 10kHz	11
1.4.6 12kHz	12
1.5 Untersuchung der sinc()-Dämpfung	13
1.5.1 Aufgabenstellung	13
1.5.2 Theorie	13
1.5.3 Messung der Werte	13
1.5.4 Vergleich	13
1.6 Aliassimulation	14
1.6.1 Aufgabenstellung	14
1.6.2 Theorie	14
1.6.3 Messung ohne Antialiasingfilter	15
1.6.4 Messung mit Antialiasingfilter	16
1.6.5 Vergleich	16
2 Übung 2: IIR-TP-Filter 1. Ordnung	17
2.1 Programmierung des IIR-TP	17
2.1.1 Aufgabenstellung	17
2.1.2 Programmcode	17
2.2 Frequenzgang	18
2.2.1 Aufgabenstellung	18
2.2.2 Messung	18
2.3 Simulation des Filters	18
2.3.1 Aufgabenstellung	18
2.3.2 Matlab-Code	18
2.3.3 Frequenzgang	19
2.3.4 Messung der Grenzfrequenz	19
2.3.5 Messung der Flankensteilheit	19
3 Übung 3 - IIR-HP-Filter 2. Ordnung	20
3.1 Simulation des IIR-HP-Filters	20
3.1.1 Aufgabenstellung	20
3.1.2 Entwurf des Filters	20
3.2 Umwandlung in 16Bit unsigned Werte	21
3.2.1 Aufgabenstellung	21
3.2.2 Matlab-Code	21
3.2.3 Umformung der IIR-Formel	21

3.3	Implementierung auf der Megacard	22
3.3.1	Aufgabenstellung	22
3.3.2	Programmcode	22
3.4	Messung des Frequenzganges	22
3.4.1	Aufgabenstellung	22
3.4.2	Bearbeitung	22
4	Übung 4 - FIR-Filter	22

1 Übung 1 - Abtastung von Signalen

1.1 Analyse der ISR

```
ISR(TIMERO_COMP_vect) //fa=10kHz --> T=100µs
{
    ue=ADCH;           //Einlesen des Samples
    ADCSRA|=(1<<ADSC); //Start der neuen Messung

    ua=ue;             //Gibt Eingangssignal wieder aus

    PORTC=ua;          //Ausgabe des Samples
}
```

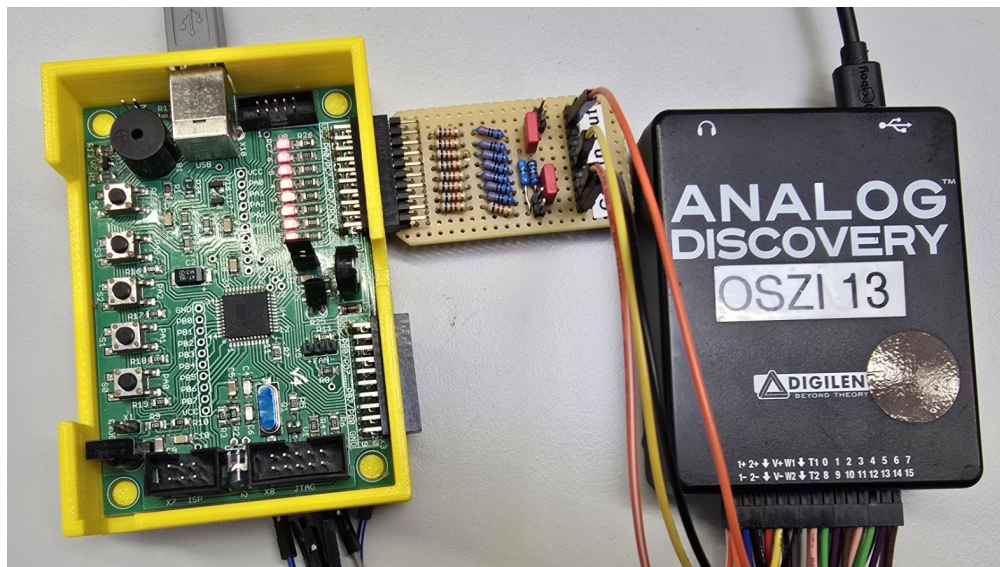
Zuerst wird die aktuelle Spannung des Eingangssignales gemessen, indem man das obere Byte des AD-Cs des ATmega16-Chips ausliest. Obwohl der ADC der MegaCard 10 Bits hat, werden nur die oberen 8 Bits verwendet. Anschließend wird ein neuer Messzyklus des ADCs gestartet, indem man das ADSC-Bit im ADCSRA-Register setzt. Das gemessene Eingangssignal (ue) wird anschließend auf das Ausgangssignal (ua) übertragen, ohne es zu verändern. Schlussendlich wird das Ausgangssignal (ua) am Port C ausgegeben. An diesem ist ein R2R-DAC angeschlossen.

1.2 Untersuchung des Samplingprozesses

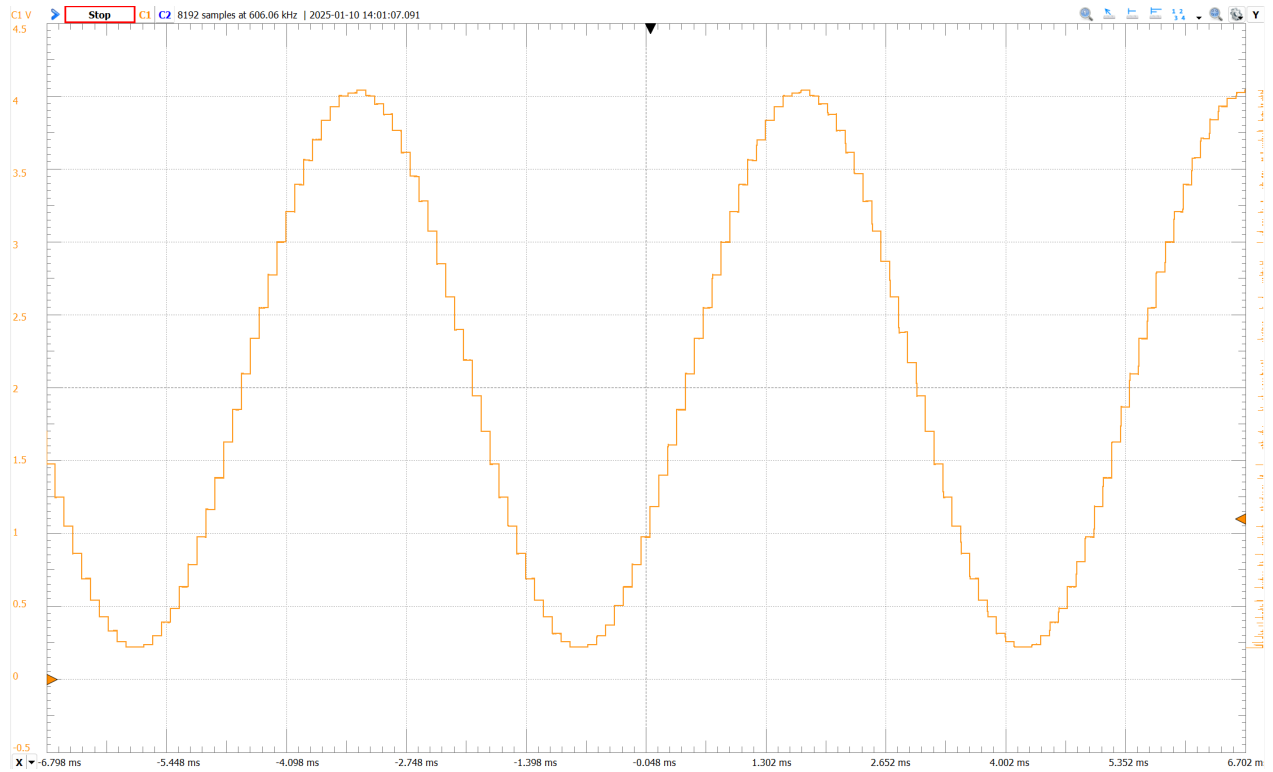
1.2.1 Aufgabenstellung

In dieser Aufgabe soll mit dem Analog Discovery ein Sinussignal mit einer Frequenz von 200Hz, einer Amplitude von 2V und einem DC-Offset von 2V in die MegaCard eingespeist werden. Das entstehende Ausgangssignal soll untersucht werden und die Spannung des niedrigsten Bits (U_{LSB}) und die Frequenz des Samplingprozesses (f_A) sollen erfasst werden.

1.2.2 Schaltungsaufbau



1.2.3 Ausgangssignal



1.2.4 Spannungsauflösung

Theroretischer Wert

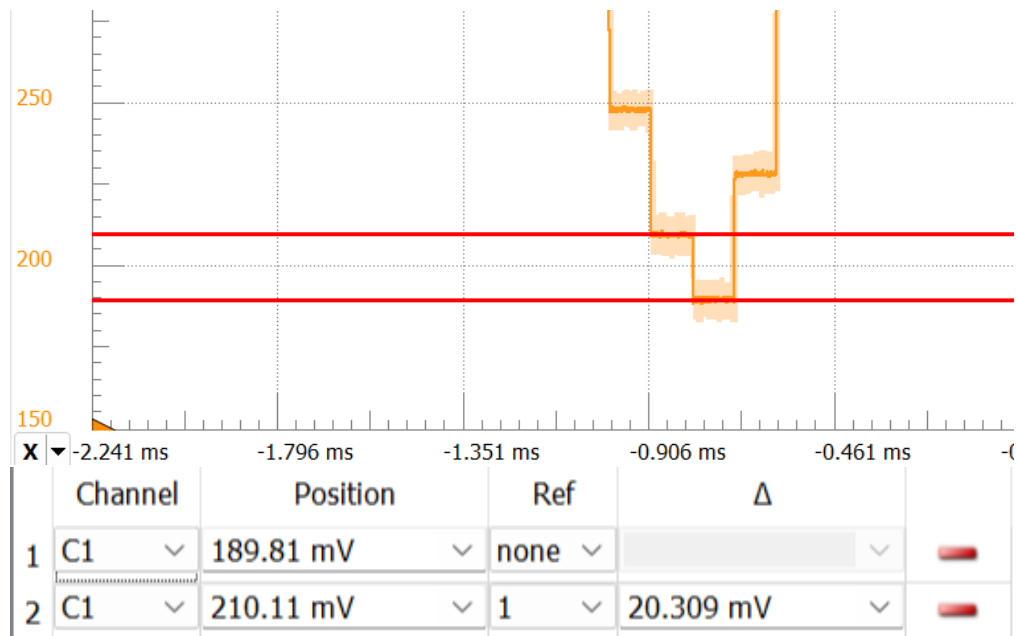
$$U_B = 5V$$

$$n = 8$$

$$U_{LSB} = \frac{U_B}{2^n} = \underline{19.831mV}$$

Messung

Um die Spannungsauflösung des DACs zu bestimmen, muss man am Tiefpunkt des Sinussignales messen, da dort die Spannung nur langsam steigt und somit sichergestellt wird, dass nur ein Bit gemessen wird. Hier wurde ein Wert von **20.309mV** erhalten.



Vergleich

Wenn man den theoretischen Wert mit dem gemessenen vergleicht, unterscheiden sich diese um nur 777.75 μ V. Dies stammt von der Toleranz der Eingangsspannung (5V vom USB) und der Ungenauigkeit der Messung.

1.2.5 Samplingfrequenz

Theroretischer Wert

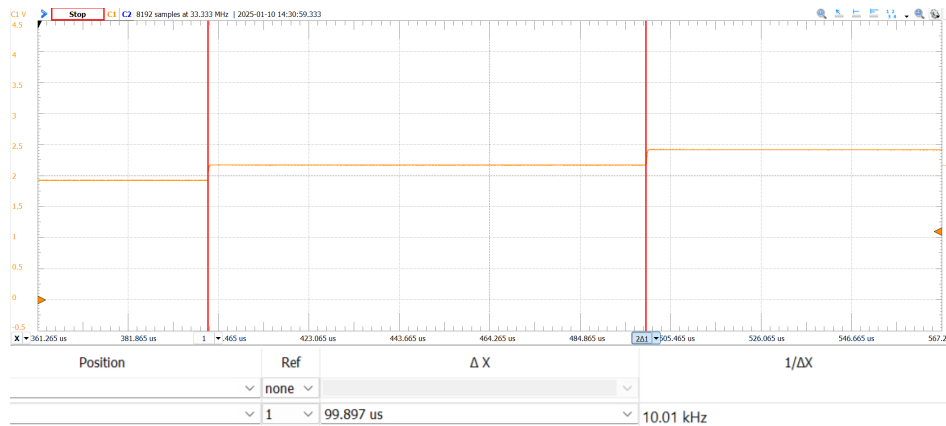
$$F_{CPU} = 12MHz$$

$$OCR0 = 149$$

$$f_a = \frac{F_{CPU}}{8 \cdot (OCR0 + 1)} = \underline{10kHz}$$

Messung

Die Samplingfrequenz muss am Nulldurchgang des Eingangssignales erfasst werden, da sich die Werte des Eingangssignales (Sinus) dort sehr schnell ändern und die benötigten Punkte leichter erkennbar sind.



Vergleich

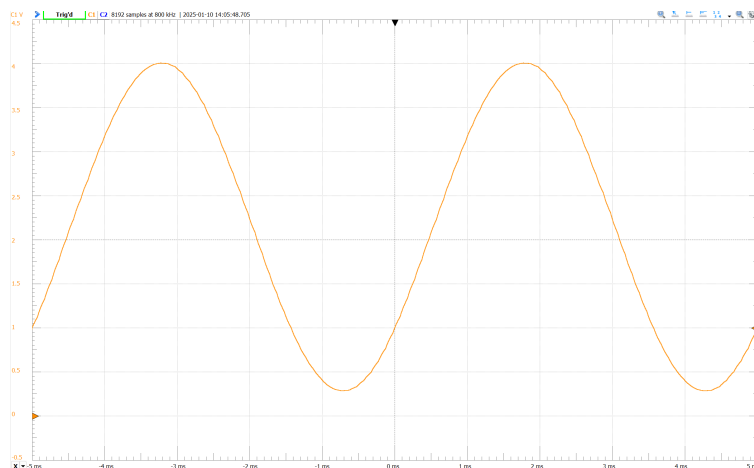
Wenn man den theoretischen Wert mit dem gemessenen vergleicht, unterscheiden sich diese um nur 10Hz. Diese Ungenauigkeit stammt von der Ungenauigkeit des Quarzes der Megacard, welcher den Timer zum Ausführen der Messung steuert, und der Toleranz des Analog Discoverys.

1.3 Untersuchung des Rekonstruktionsfilters

1.3.1 Aufgabenstellung

In dieser Aufgabe soll der Rekonstruktionsfilter des DACs über den Jumper X2 eingeschalten werden. Das entstehende Ausgangssignal soll mit der vorherigen Aufgabe (ohne Rekonstruktionsfilter) verglichen werden.

1.3.2 Ausgangssignal



1.3.3 Vergleich

Beim Vergleichen der beiden Ausgangssignale ist schnell erkennbar, dass das Rekonstruktionsfilter die hohen Frequenzen des Signals wegfiltert. Dies ist erkennbar, da das Signal nicht mehr stufenartig aufgebaut ist. Somit ähnelt es einem reinen Sinus mehr als dem Ausgangssignal ohne den Filter.

1.4 Untersuchung mehrerer Frequenzen

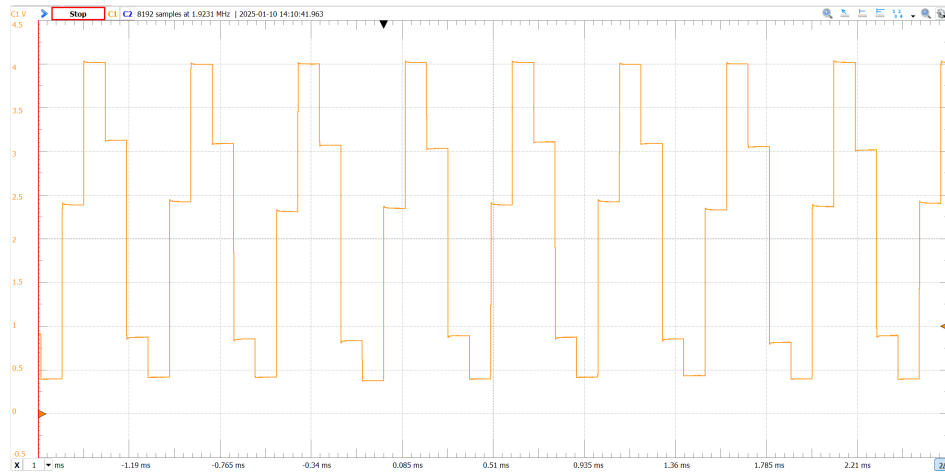
1.4.1 Aufgabenstellung

In dieser Aufgabe sollen mehrere Signalfrequenzen angelegt werden. Dabei soll kein Filter des DACs aktiviert werden. Die zu messenden Frequenzen sind 2kHz, 5kHz, 8kHz, 10kHz und 12kHz. Dabei ist der Zeit- und Frequenzbereich zu untersuchen.

1.4.2 2kHz

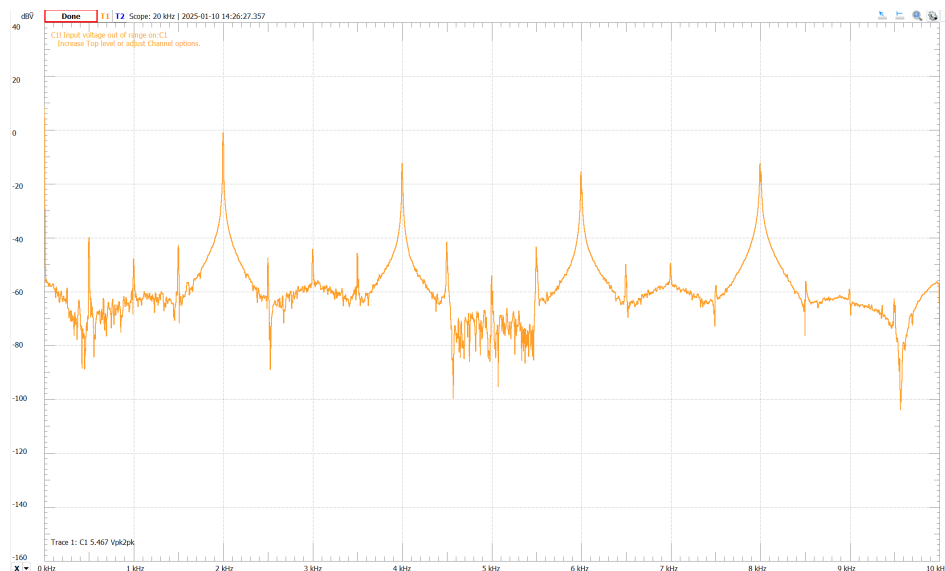
Zeitbereich

Das Signal im Zeitbereich hat zwar wenig Ähnlichkeit mit einem Sinus, aber seine Form ist immer noch erkennbar. Dies stammt von der langsamen Abtastfrequenz im Vergleich zum Eingangssignal.



Frequenzbereich

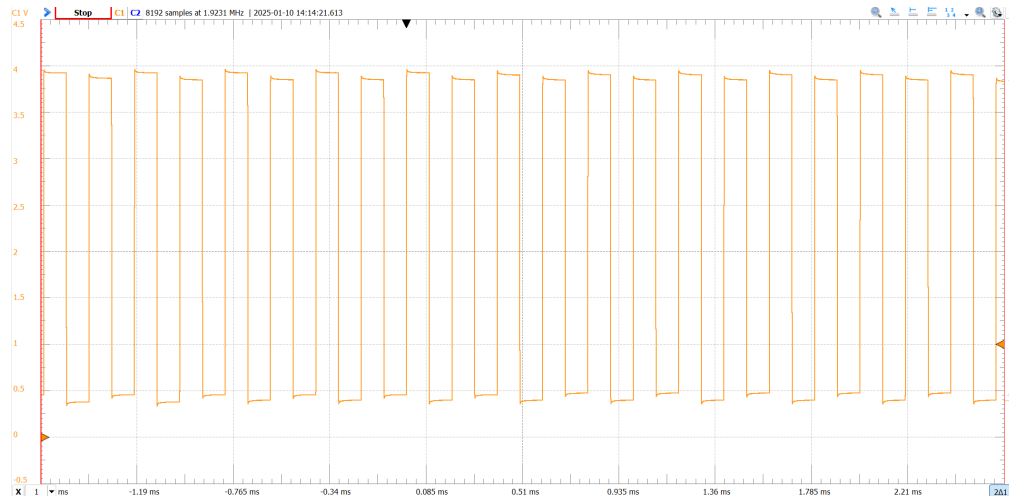
Im Frequenzbereich des Signals kann ein hoher Peak alle 2kHz erkannt werden. Dabei sind die bei 2kHz- und 10kHz-Peaks am höchsten. Der 2kHz-Peak stammt vom Eingangssignal, welches eine Frequenz von 2kHz hat. Der Peak bei 8kHz stammt vom Samplingprozess, da das Signal mit einer Frequenz von 10kHz aufgenommen wird, wird es auf 8kHz ($10\text{kHz} - 2\text{kHz}$) und 12kHz ($10\text{kHz} + 2\text{kHz}$) gespiegelt.



1.4.3 5kHz

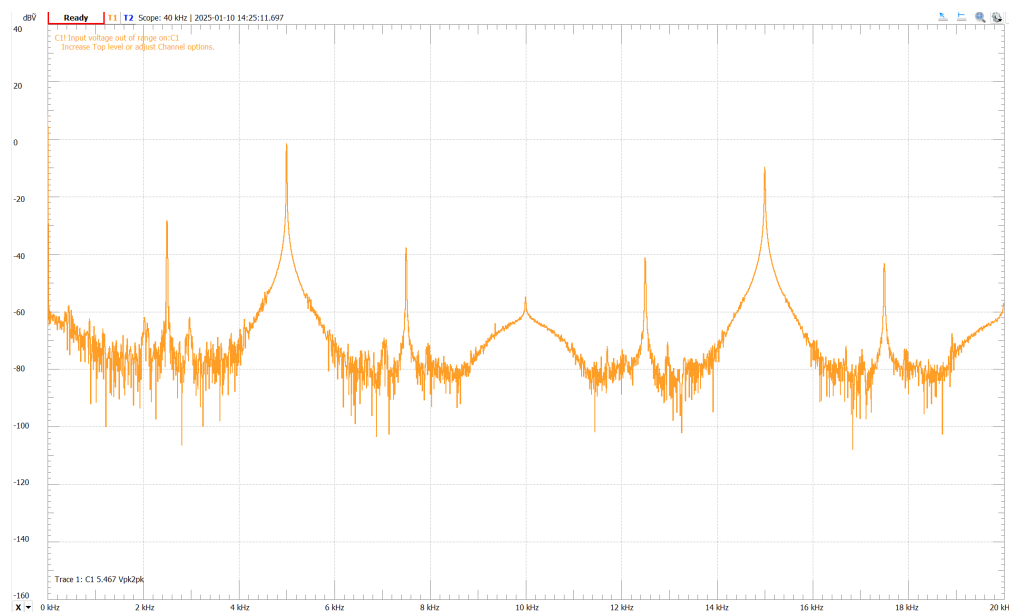
Zeitbereich

Das Signal im Zeitbereich hat keine Ähnlichkeit mit einem Sinus, sondern sieht wie ein Rechtecksignal aus. Das angezeigte Signal verändert zudem seine Amplitude über die Zeit. Dies liegt daran, dass die Nyquist-Frequenz erreicht wurde. Aus diesem Grund werden zu wenig Messpunkte des Eingangssignals aufgenommen und verarbeitet, was zu dem rechteckförmigen Aufbau führt. Die Amplitudenänderung des Ausgangssignals stammt von leichten Frequenzdifferenzen zwischen dem Eingangssignal und der Samplingfrequenz, die dazu führen, dass die Messpunkte immer an anderen Stellen des Sinussignals aufgenommen werden.



Frequenzbereich

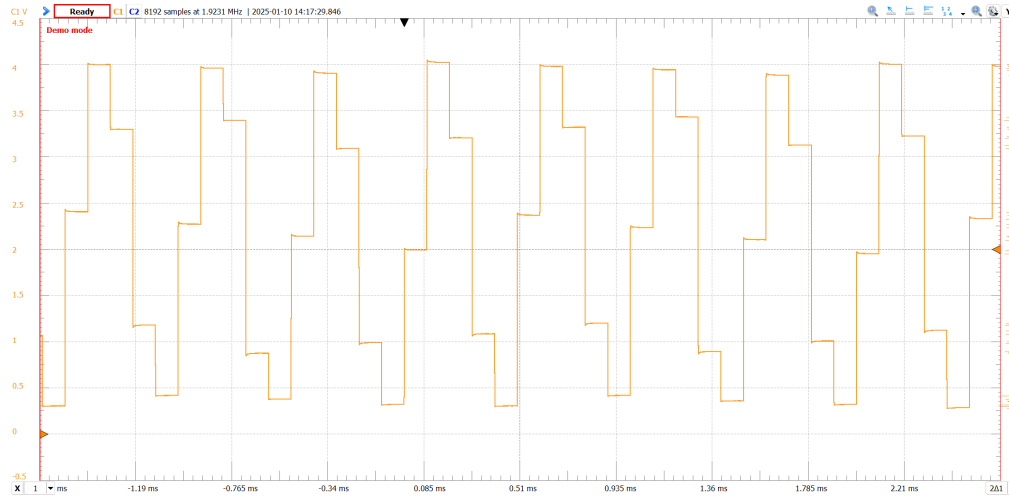
Im Frequenzbereich des Signals kann ein hoher Peak bei 5 kHz und 15 kHz erkannt werden. Das Signal bei 5kHz ist die Summe aus dem Eingangssignal und der unteren Spiegelung bei der Samplingfrequenz (10kHz - 5kHz). Der Peak bei 15kHz stammt von der oberen Spiegelung bei der Samplingfrequenz (10kHz + 5kHz).



1.4.4 8kHz

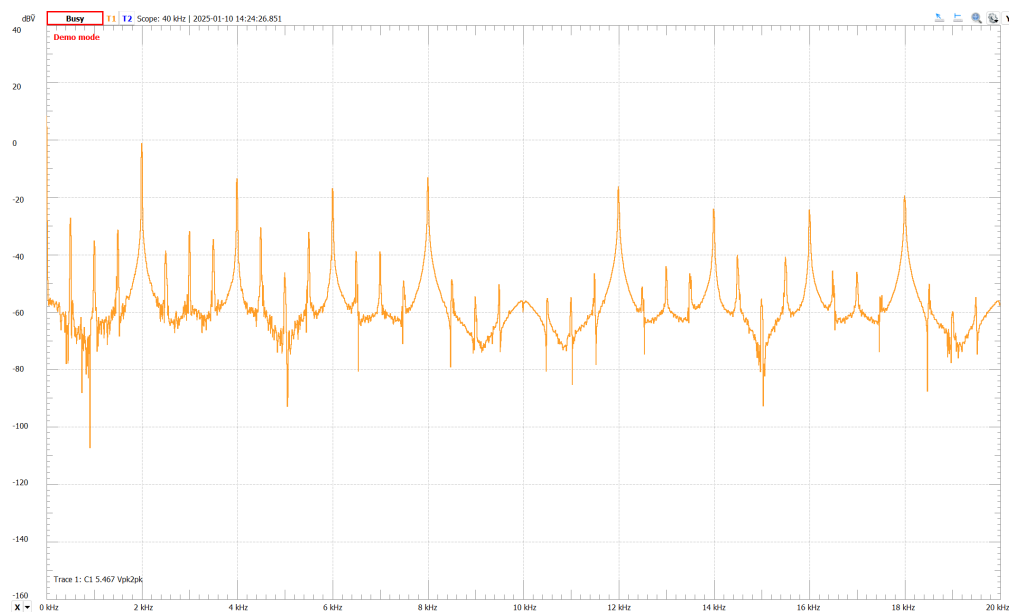
Zeitbereich

Bei der Untersuchung des Signales im Zeitbereich ist schnell erkennbar, dass es dem Signal bei einer Eingangsfrequenz von 2kHz sehr ähnlich ist. Dies liegt daran, dass die Eingangsfrequenz des 2kHz-Signals mit der unteren Spiegelung des 8kHz bei einer Samplingfrequenz von 10kHz ist ($10\text{kHz} - 8\text{kHz} = 2\text{kHz}$). Die untere Spiegelung des 2kHz-Signals (8kHz) wird durch das Eingangssignal ersetzt. Dies ist bei den anderen Punkten des Signales auch der Fall.



Frequenzbereich

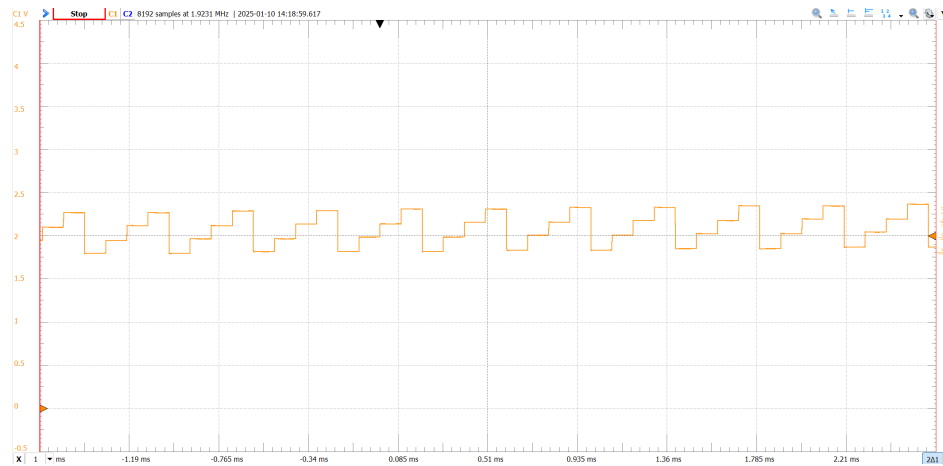
Im Frequenzbereich kann erneut die Ähnlichkeit mit dem des 2kHz-Eingangssignals erkannt werden. Der Grund dafür wurde unter dem Punkt *Zeitbereich* erklärt. Es sind erneut zwei Peaks bei 2kHz- und 10kHz zu erkennen. Der 2kHz-Peak stammt von der Reflektion durch die Samplingfrequenz ($10\text{kHz} - 8\text{kHz}$). Der Peak bei 8kHz stammt vom Eingangssignal.



1.4.5 10kHz

Zeitbereich

Das Signal im Zeitbereich hat keine Ähnlichkeit mit einem Sinus, sondern sieht wie ein Sägezahn aus. Dieses verändert seine Höhe und seine Richtung (Spiegelung entlang der y-Achse). Zudem ist die Amplitude des Ausgangssignals sehr viel kleiner als bei den Messungen mit den anderen Frequenzen und wie das Eingangssignal. Die Form des Ausgangssignals stammt von der hohen Frequenz des Eingangssignals im Vergleich zur Samplingfrequenz. Theoretisch sollte das Signal eine DC-Spannung darstellen, da der Sinus am Eingang immer an den gleichen Punkten abgemessen wird. Durch leichte Toleranzen der Frequenzen wird dies nicht am Ausgangssignal beobachtet.



Frequenzbereich

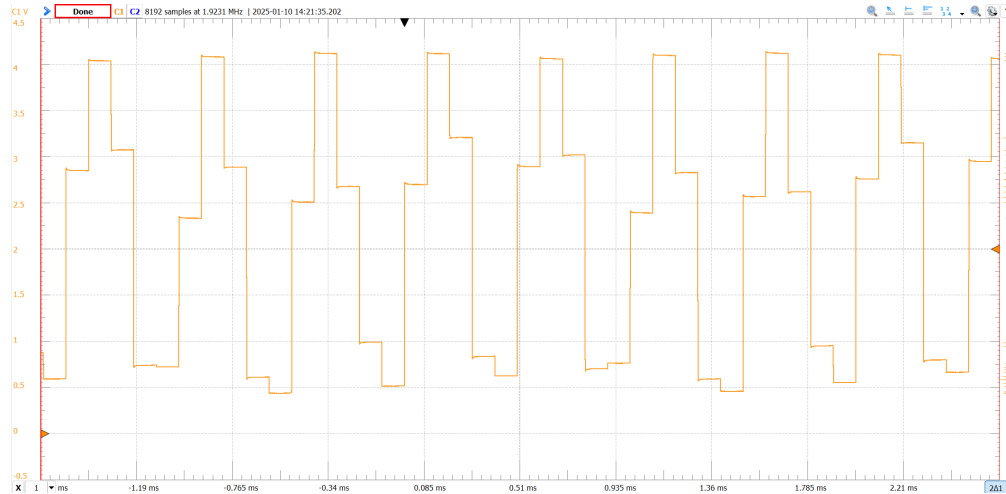
Im Frequenzbereich des Signals können kaum sinnvolle Punkte erkannt werden, welche mit der Theorie übereinstimmen, da sich das Ausgangssignal ständig verändert, ändert sich auch das Diagramm im Frequenzbereich. Jedoch kann bei einer Frequenz von 0 Hz eine Spannung erkannt werden. Diese DC-Spannung stammt (laut Theorie) von der unteren Spiegelung des Eingangssignals bei der Samplingfrequenz ($10 \text{ kHz} - 10 \text{ kHz} = 0 \text{ Hz}$). Theoretisch sollten sich weitere Punkte alle 10 kHz befinden, was allerdings nicht erkennbar ist.



1.4.6 12kHz

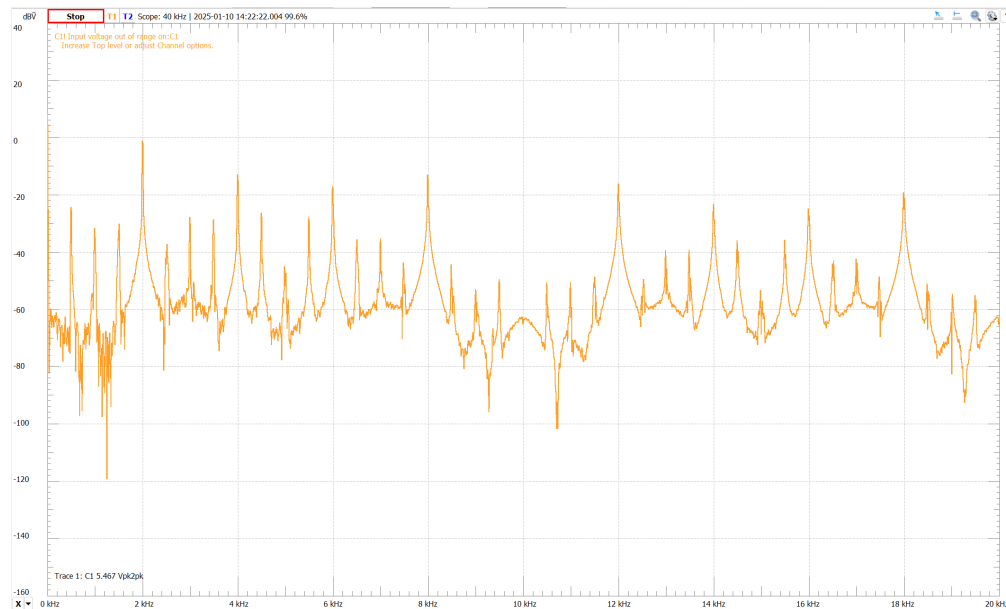
Zeitbereich

Wie bei einer Eingangsfrequenz von 8 kHz ähnelt das Signal sehr dem Signal bei einer Eingangsfrequenz von 2 kHz. Dies liegt daran, dass das Signal durch den Aliasfehler als 2 kHz wahrgenommen wird.



Frequenzbereich

Im Frequenzbereich des Signals kann erneut die Ähnlichkeit zu 2 kHz erkannt werden. Der Peak bei 2 kHz stammt von der unteren Spiegelung bei der Samplingfrequenz ($10 \text{ kHz} - 12 \text{ kHz} = -2 \text{ kHz}$). Dabei werden negative Frequenzen als positive wahrgenommen, da sich bei ihnen der Sinus nur in die gegengesetzte Richtung drehen würde.



1.5 Untersuchung der sinc()-Dämpfung

1.5.1 Aufgabenstellung

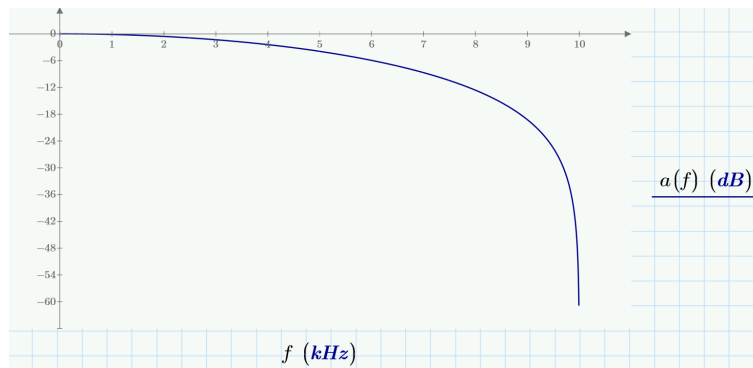
In dieser Aufgabe soll die sinc()-Dämpfung eines 2 kHz-Signals bei 2 kHz und 8 kHz herausgemessen und mit der Theorie verglichen werden.

1.5.2 Theorie

Die Verstärkung (bzw. Dämpfung) der sinc()-Funktion kann über folgende Formel berechnet werden:

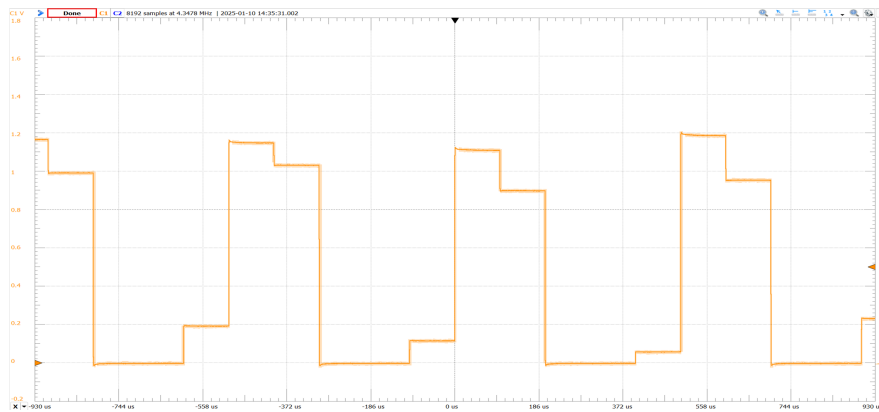
$$a(f) := 20 \cdot \log \left(\frac{\sin \left(\pi \cdot \frac{f}{f_a} \right)}{\pi \cdot \frac{f}{f_a}} \right) \quad (1)$$

dabei ist f die gesuchte Frequenz und f_a die Samplingfrequenz, welche in diesem Fall 10 kHz beträgt. Beim Plotten dieser Funktion entsteht folgender Graph: Aus diesem Graph kann ausgelesen werden, dass die



Dämpfung bei einer Frequenz von 2 kHz -0.579 dB und bei einer Frequenz von 8 kHz -12.62 dB beträgt.

1.5.3 Messung der Werte



Das gemessene Frequenzspektrum ist dem aus der vorherigen Aufgabe sehr ähnlich. Die Dämpfung bei einer Frequenz beträgt laut Messung ca. **-0.894 dB** und bei einer Frequenz von 8 kHz ca. **-11.49 dB**.

1.5.4 Vergleich

Wenn man die Werte der Theorie und der Messung vergleicht, sind diese sehr ähnlich. Die Dämpfung bei 2 kHz unterscheidet sich um 0.315 dB und bei 8 kHz nur um 1.13 dB. Diese Abweichungen stammen von möglichen Toleranzen der Frequenzen, was zur Folge hat, dass sich die sinc()-Funktion verzerrt. Aber auch die Auflösung des Analog Discoveries beeinflusst dies.

1.6 Aliassimulation

1.6.1 Aufgabenstellung

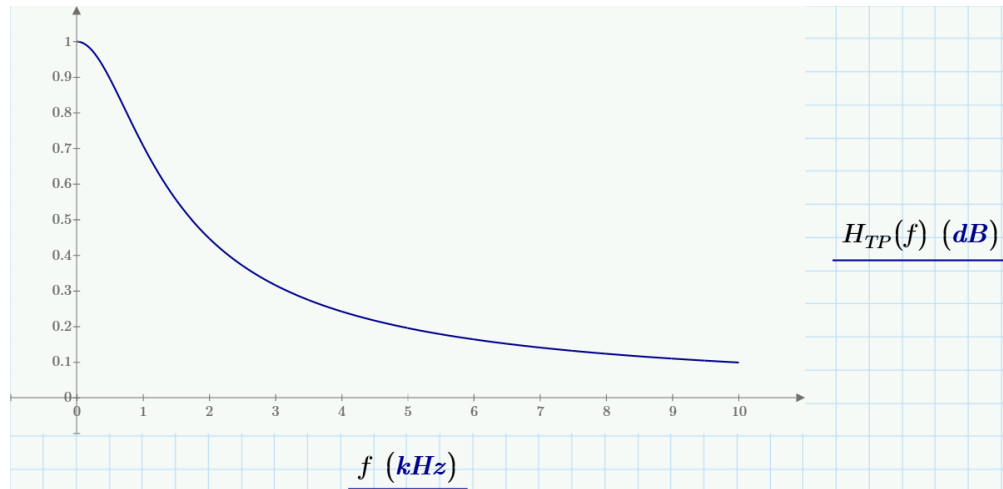
In dieser Aufgabe soll eine Aliasspannung simuliert werden, indem eine Sinusspannung mit 8 kHz eingespeist wird. Um einen besseren Vergleich zu ermöglichen, soll dies zuerst ohne den Antialiasingfilter (X1 aus) und anschließend mit dem Antialiasingfilter (X1 ein) aufgenommen werden. Dabei sollen Änderungen im Nutzbereich ($0 \dots f_a/2$) beschrieben werden und das Ergebnis sowohl im Zeit- als auch im Frequenzbereich mit der Theorie verglichen werden.

1.6.2 Theorie

Die Dämpfung eines Tiefpasses 1. Ordnung kann mit folgender Formel ausgerechnet werden:

$$H_{TP}(f) := \frac{1}{\sqrt{1 + \left(\frac{f}{f_g}\right)^2}} \quad (2)$$

Dabei ist f die Signalfrequenz und f_g die Grenzfrequenz des Tiefpasses. Wenn diese Funktion geplottet wird, entsteht folgender Graph:

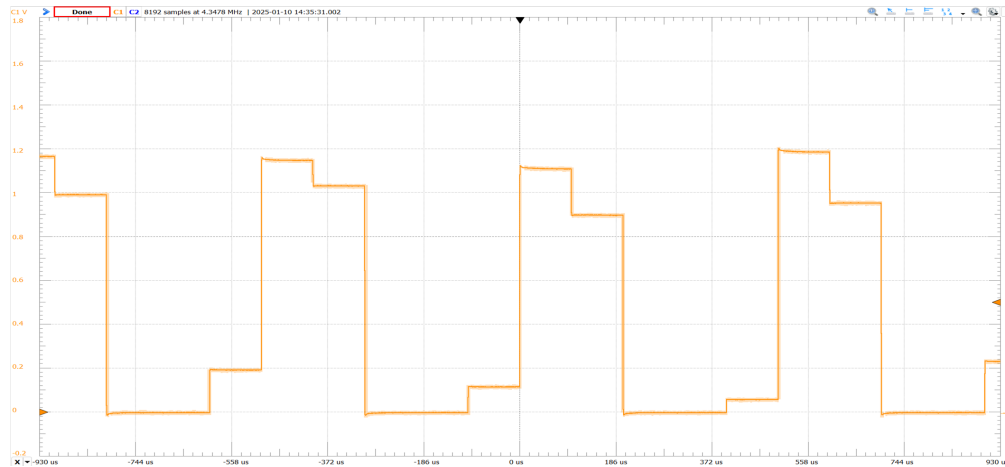


Aus diesem Graph kann herausgelesen werden, dass der Tiefpass ein Signal mit einer Frequenz von 2 kHz mit **0.447 (-7 dB)** und einer Frequenz von 8 kHz mit **0.124 (-18.129 dB)** dämpft.

1.6.3 Messung ohne Antialiasingfilter

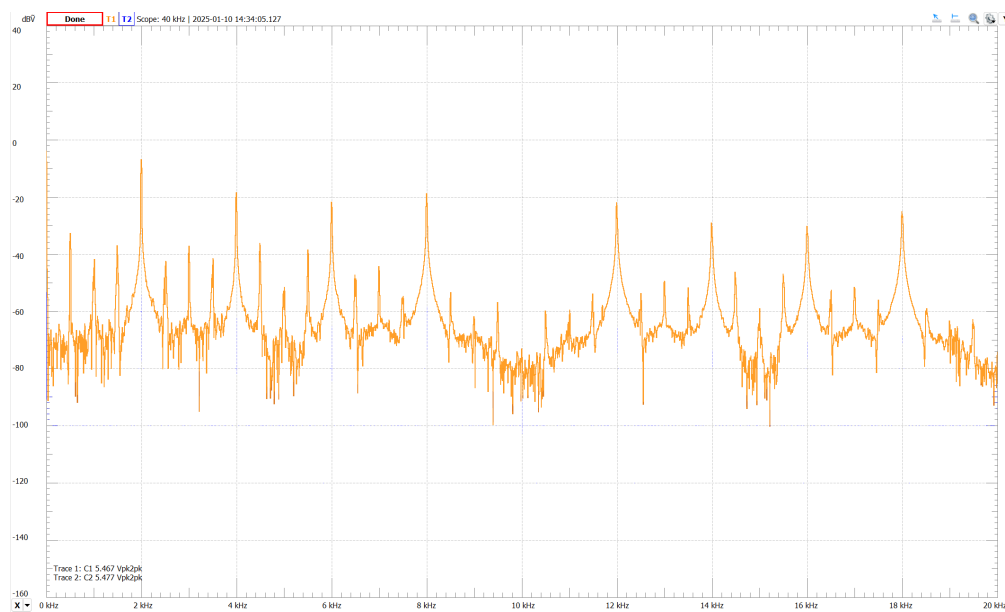
Zeitbereich

Die Form des erhaltenen Signals im Zeitbereich ist ähnlich wie in der vorherigen Aufgabe, bei der auch ein Eingangssignal mit 2 kHz eingespeist wurde. Allerdings ist das Signal kleiner als zuvor. (Dies sollte eigentlich nicht der Fall sein. Es besteht die Annahme, dass eine falsche Eingangsspannung eingestellt wurde.)



Frequenzbereich

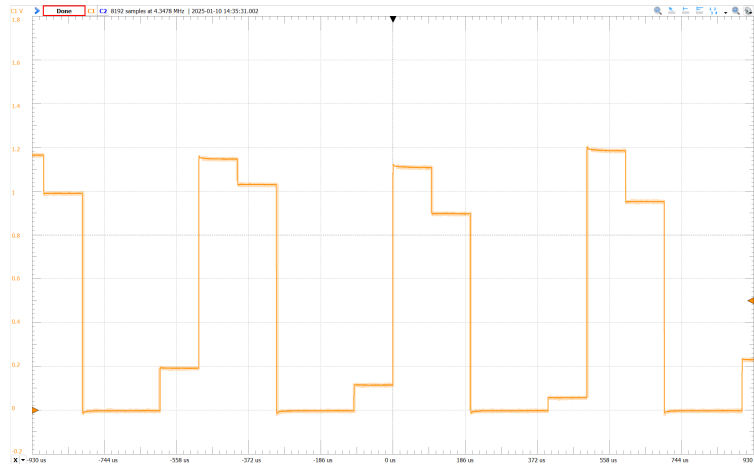
Wie im Zeitbereich ist das Signal auch im Frequenzbereich dem vorherigen 2 kHz Signal sehr ähnlich.



1.6.4 Messung mit Antialiasingfilter

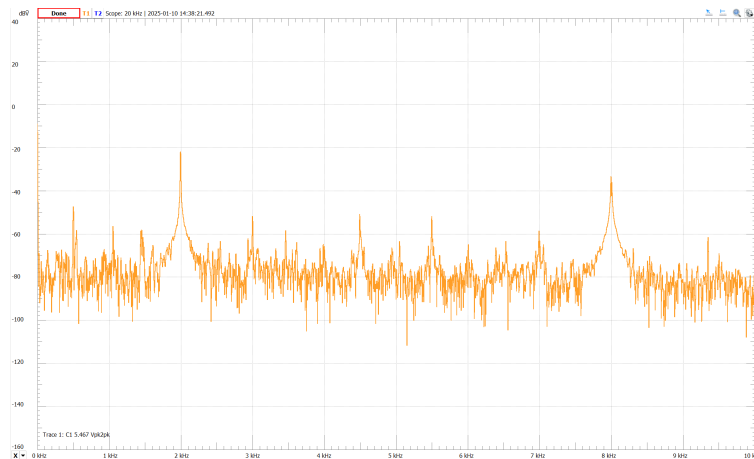
Im Vergleich zum vorherigen Signal ist die Spannung des Signals kleiner. Grund dafür ist, dass der Tiefpass die Spannungen dämpft. Zudem erinnert die Form des Signals mehr an einen Sinus als zuvor.

Zeitbereich



Frequenzbereich

Der Frequenzbereich des gefilterten Signals unterscheidet sich deutlich von vorherigen. Hier ist klar erkennbar, dass viele ungewollte Frequenzen entfernt wurden, da hauptsächlich die zwei Peaks bei 2 kHz und 8 kHz erkennbar sind. Dabei hat der Peak bei 2 kHz einen Wert von ca. -22 dB und bei 8 kHz ca. -32 dB.



1.6.5 Vergleich

Die theoretischen Werte unterscheiden sich stark von den Messergebnissen. Wie schon beschrieben, wird vermutet, dass eine falsche Eingangsspannung eingestellt wurde, da ohne Filter ein seltsames Ergebnis gemessen wurde. Kleine Abweichungen wären durch Bauteilungenauigkeiten oder andere Toleranzen des Systems normal, aber die starken Abweichungen des Ergebnisses weisen auf ein größeres Problem bei der Messung.

2 Übung 2: IIR-TP-Filter 1. Ordnung

2.1 Programmierung des IIR-TP

2.1.1 Aufgabenstellung

In dieser Aufgabe soll der IIR-Tiefpass-Filter 1. Ordnung in der ISR implementiert werden. Dafür sollen folgende Koeffiziente verwendet werden: $b_0 = 0.1$
 $a_1 = 0.9$ Diese müssen in die allgemeine Formel des IIR-Filters eingesetzt werden.

$$y(n) = \sum_{k=0}^M b_k \cdot x(n-k) - \sum_{j=1}^N a_j \cdot y(n-j) \quad (3)$$

Für einen Filter 1. Ordnung lautet die benötigte Formel:

$$y(n) = b_0 \cdot x(n) + y(n-1) \cdot a_1 \quad (4)$$

2.1.2 Programmcode

Der Programmcode wurde mit Floatingpoint-Zahlen ausprogrammiert.

```
ISR(TIMERO_COMP_vect)                //fa=10kHz --> T=100us
{
    ue=ADCH;                          //Einlesen des Samples
    ADCSRA|=(1<<ADSC);               //Start der neuen Messung

    float x_neu = (float)ue;          //ADC-Wert in Float umwandeln

    // u(n) = x(n) * b0 + y(n-1) * a1
    float y_neu = x_neu * b0 + y_alt * a1;

    ua = (unsigned char)y_neu;        //float zu char

    y_alt = y_neu;                    //Speicher für nächste Rechnung

    PORTC=ua;                         //Ausgabe des Samples
}
```

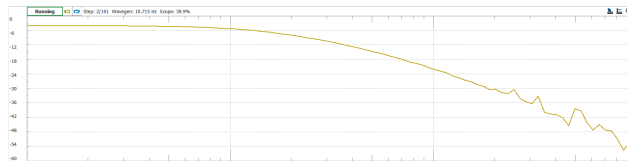
2.2 Frequenzgang

2.2.1 Aufgabenstellung

In dieser Aufgabe soll der Frequenzgang des ausprogrammierten IIR-Tiefpasses überprüft werden. Dafür soll der Netzwerk-Analysator des Analog Discoverys verwendet werden. Dieser soll einen DC-Offset von 2 V und eine Amplitude von 1 V besitzen.

2.2.2 Messung

Am folgenden Frequenzgang ist klar erkennbar, dass es sich um einen Tiefpass 1. Ordnung handelt. Es handelt sich um einen Tiefpass, da die Verstärkung bei niedrigen Frequenzen fast konstant verläuft und dann abfällt. Der Graph läuft von 10Hz bis 10kHz (*x-Achsenbeschriftung wurde im Screenshot abgehackt*)



Da die Verstärkung bei niedrigen Frequenzen ca. -3.5 dB groß ist, hat die Grenzfrequenz eine Verstärkung von -6.5 dB. Laut Graph ist dies bei ca. 150 Hz. Da die Verstärkung bei 200 Hz ca. -7.5 dB und bei 2 kHz ca. -30 dB groß ist, ist die Verstärkung um 22.5 dB gefallen (-20 dB / Dekade). Da dies ungefähr einem Abfall von 20 dB pro Dekade entspricht, handelt es sich um einen Tiefpass 1. Ordnung.

2.3 Simulation des Filters

2.3.1 Aufgabenstellung

In dieser Aufgabe soll das IIR-TP-Filter in Matlab simuliert werden. Dafür soll der `freqz()`-Befehl eingesetzt werden. Aus dem entstandenen Frequenzgang soll die Grenzfrequenz und Flankensteilheit ausgemessen werden.

2.3.2 Matlab-Code

```
%Koeffizienten
b0 = 0.1;
a1 = 0.9;

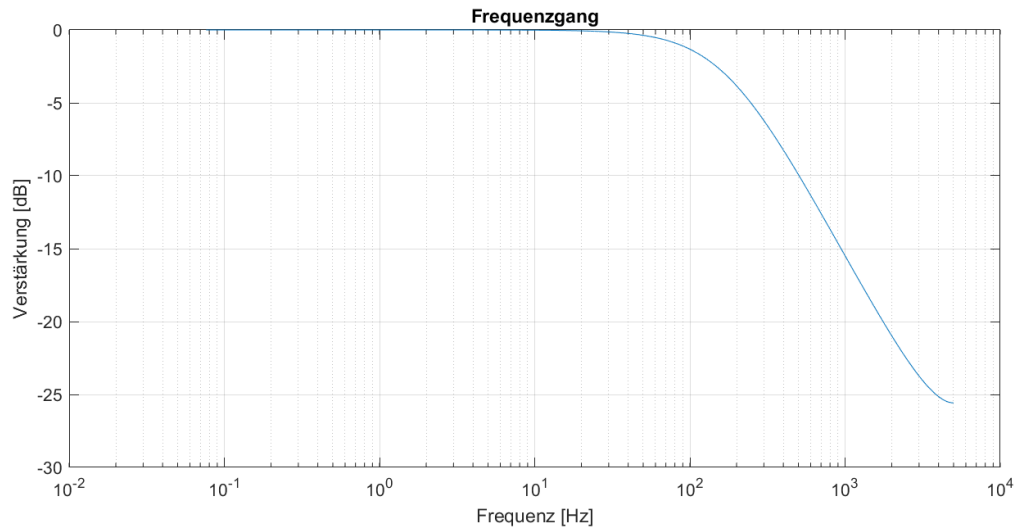
b = [b0];      %Zähler
a = [1, -a1];  %Nenner

%Frequenzgang
[H, f] = freqz(b, a, 2^16, 10000); %2^16 Punkte, Sample 10 kHz

%Plotten
semilogx(f, 20*log10(abs(H)));
xlabel('Frequenz [Hz]');
ylabel('Verstärkung [dB]');
grid on;
title('Frequenzgang');
```

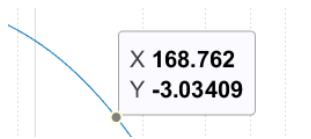
2.3.3 Frequenzgang

Der entstehende Frequenzgang stellt einen Tiefpass dar. Dieser hat bei niedrig eine Verstärkung von 0 dB und filtert hohe Frequenzen weg.



2.3.4 Messung der Grenzfrequenz

Die Grenzfrequenz (Punkt bei -3 dB) liegt laut Simulation bei einer Frequenz von ca. 168.762 Hz.



2.3.5 Messung der Flankensteilheit

Um die Flankensteilheit zu messen, müssen zwei Punkte bekannt sein. Da durch die Messung der Grenzfrequenz schon ein Punkt vorhanden ist, muss nur noch ein Punkt erfasst werden. Dieser liegt eine Dekade nach dem ersten Punkt, also bei 1687.62 Hz. Dort liegt die Verstärkung bei ca. -19.6181 dB. Somit liegt die Differenz bei ca. 16.6171 dB. Ein gewöhnlicher Tiefpass 1. Ordnung sollte einen Abfall von -20 dB / Dekade haben. Die Abweichung des Messergebnisses stammt von der nahen Messung an der Grenzfrequenz, bei welcher der Tiefpass noch nicht linear abfällt.



3 Übung 3 - IIR-HP-Filter 2. Ordnung

3.1 Simulation des IIR-HP-Filters

3.1.1 Aufgabenstellung

In dieser Aufgabe soll Matlab benutzt werden, um einen IIR-HP-Filter der 2. Ordnung zu entwerfen. Dieser soll eine Grenzfrequenz von 1000 Hz haben. Nachdem der Filter entworfen wurde, soll der Frequenzgang dessen untersucht werden.

3.1.2 Entwurf des Filters

```
fc = 1000; %Grenzfrequenz = 1kHz
fs = 10000; %Samplefrequenz = 10kHz

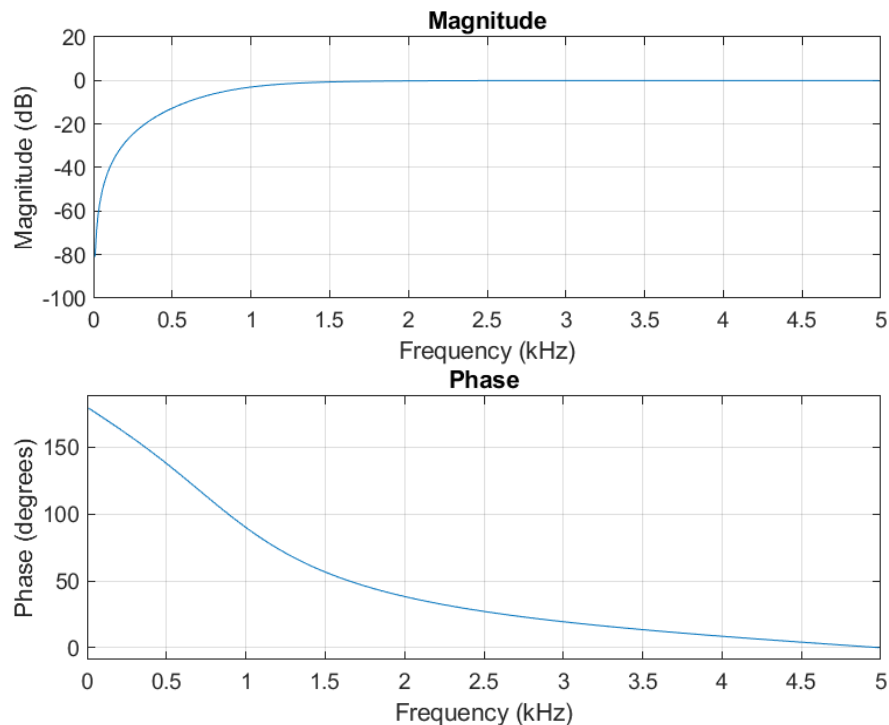
[b,a] = butter(2,fc/(fs/2), "high"); %Faktoren des Filter bestimmen

freqz(b,a,[],fs) %Frequenzgang berechnen

%Faktoren ausgeben
a1 = a(2)
a2 = a(3)

b0 = b(1)
b1 = b(2)
b2 = b(3)

%Frequenzgang plotten
subplot(2,1,1)
ylim([-100 20])
```



3.2 Umwandlung in 16Bit unsigned Werte

3.2.1 Aufgabenstellung

In dieser Aufgabe sollen die berechneten Koeffizienten als 16-Bit unsigned Werte exportiert werden, um sie auf der Megacard zu implementieren.

3.2.2 Matlab-Code

Der folgende Matlab-Code wandelt die dezimalen Koeffizienten in den benötigten Datentyp um. Dabei wurden sie mit 256 multipliziert, damit die Werte auf der Megacard die uint16-Größe nicht überschreiten. Die Formel des IIR-Filters wurde umgewandelt, damit alle eingegebenen Konstanten positiv sind.

```
%Faktoren ausgeben
a1 = round(a(2) * 256)
a2 = round(a(3) * 256)

b0 = round(b(1) * 256)
b1 = round(b(2) * 256)
b2 = round(b(3) * 256)
```

3.2.3 Umformung der IIR-Formel

In der folgenden Umformung wird die Gleichung des IIR-Filters so angepasst, dass keine negativen Koeffizienten mehr vorhanden sind.

Handwritten derivation of the IIR filter equation:

$$b = [0,6389 \quad b_1 \quad b_2]$$

$$a = [a_0 \quad a_1 \quad a_2]$$

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

$$Y(z) = \frac{Y(z)}{1} \cdot \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} = X(z) \cdot \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

$$Y(z) \cdot a_0 = X(z) \cdot b_0 + X(z) \cdot b_1 z^{-1} + X(z) \cdot b_2 z^{-2}$$

$$y(n) \cdot a_0 = x(n) \cdot b_0 + x(n-1) \cdot b_1 + x(n-2) \cdot b_2 - y(n-1) \cdot a_1 - y(n-2) \cdot a_2$$

$$y(n) \cdot \overset{1}{a_0} = x(n) \cdot b_0 - x(n-1) \cdot b_1 + x(n-2) \cdot b_2 + y(n-1) \cdot a_1 - y(n-2) \cdot a_2$$

3.3 Implementierung auf der Megacard

3.3.1 Aufgabenstellung

In dieser Aufgabe sollen die exportierten uint16-Koeffizienten benutzt werden, um den HP-Filter auf der Megacard zu implementieren.

3.3.2 Programmcode

```
ISR(TIMERO_COMP_vect)    //fa=10kHz --> T=100us
{
    ue=ADCH;              //Einlesen des Samples
    ADCSRA|=(1<<ADSC);    //Start der neuen Messung

    ua    = (uint16_t)(164*ue + 164 * x[1] + 293 * y[0] -106 * y[1] - 327 * x[0])>>8; //Wert ausrechnen
    //x-Werte zeitlich verschieben
    x[1] = x[0];
    x[0] = ue;

    //y-Werte zeitlich verschieben
    y[1] = y[0];
    y[0] = ua;

    PORTC=ua;             //Ausgabe des Samples
}
```

3.4 Messung des Frequenzganges

3.4.1 Aufgabenstellung

In dieser Aufgabe soll die Implementierung des HP-Filters auf der Megacard getestet werden, indem der Frequenzgang aufgenommen wird.

3.4.2 Bearbeitung

Leider ergab die Implementierung des Codes keinen sauberen Frequenzgang. Am entstandenen Frequenzgang konnten zudem keine Hochpasscharakteristiken festgestellt werden. Es wurde versucht, den Fehler zu finden und zu beheben, aber dies hatte keinen Erfolg.

4 Übung 4 - FIR-Filter

Aus zeitlichen Gründen wurde diese Übung nicht behandelt.