

TU Berlin Fakultät IV  
Institut für Energie und Automatisierungstechnik  
Fachgebiet Elektronische Mess- und Diagnosetechnik  
Praktikum Messdatenverarbeitung

# **Praktikum Messdatenverarbeitung**

## **Termin 3**

Dirk Babendererde (321 836)  
Thomas Kapa (325 219)  
Magdalene Busuru (319 433)

14. November 2012

Gruppe: G2 Fr 08-10

Betreuer: Jürgen Funk

## Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitungsaufgaben</b>	<b>1</b>
<b>2</b>	<b>Vorbereitung</b>	<b>1</b>
2.1	Tiefpassfilter 2. Ordnung . . . . .	1
2.2	Tiefpassfilter 8. Ordnung . . . . .	2
<b>3</b>	<b>Praxisteil</b>	<b>3</b>
3.1	Ineare Regression, Steigung und Offset der ADU-Kennlinie . . . . .	3
3.2	Rauschen . . . . .	3
3.3	Amplitdenfrequenzgang des Antialiasing-Filters . . . . .	4
3.4	Matlab-Code . . . . .	4

## 1 Vorbereitungsaufgaben

Eine digitale Messkette besteht aus einem Sensor, der Signalkonditionung (z.B. Linearisierung, Verstärkung ...), dem Antialiasingfilter, einem Abtast- und Halteglied und einem Analog-Digital-Umsetzer. Der Analog-Digital-Umsetzer hat eine minimale Abtastfrequenz, z.B. dadurch, dass man ein Register als Referenz für einen Timer nutzt, dass aber eine begrenzte Anzahl an Bits zur Verfügung stellt. Sensor, Signalkonditionierung und Antialiasingfilter sitzen in der Wandlerbox, Abtast- und Halteglied und ADU sitzen auf dem Microcontroller.

## 2 Vorbereitung

### 2.1 Tiefpassfilter 2. Ordnung

Um die Dämpfung bestimmen zu können muss zunächst das  $U_{LSB}$  errechnet werden:

$$U_{LSB} = \frac{14V}{2^{10} - 1Bit} = \frac{14V}{1023Bit} \quad (1)$$

Die Dämpfung ergibt sich nun daraus, dass ab der Grenzfrequenz die Maximale Eingangsspannung von  $\pm 7V$  auf unter  $U_{LSB}/2$  gedämpft werden muss:

$$A = \frac{14}{7 \cdot 2} \cdot \frac{1}{1023} = \frac{1}{1023} = -60dB \quad (2)$$

Nun simulieren wir einen Butterworth-Tiefpass 2. Ordnung und lesen ab, bei welcher Frequenz die Amplitude unter  $-60dB$  sinkt.

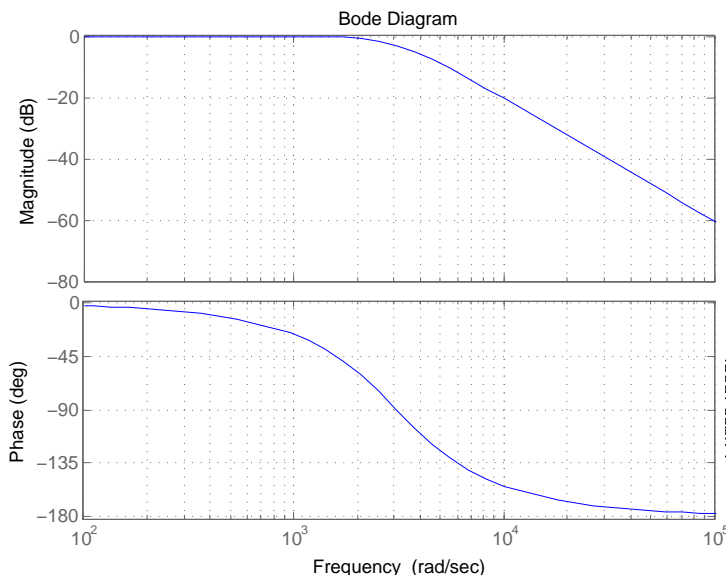


Abbildung 1: Butterworth Filter 2. Ordn.

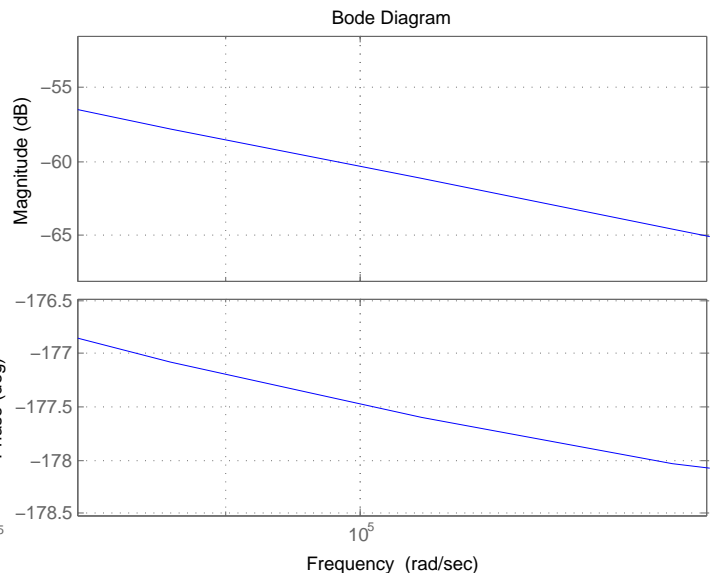


Abbildung 2: In die Grafik des Filters hineingezoomt

Daraus folgt, dass  $f \approx 100kHz$ .

Da  $f_0 < 2 \cdot f$  sein muss, ist die Mindestabtastrate:

$$f_0 \approx 2 \cdot 100kHz = 200kHz \quad (3)$$

### 2.2 Tiefpassfilter 8. Ordnung

Die Vorgehensweise beim Tiefpass 8. Ordnung ist analog.

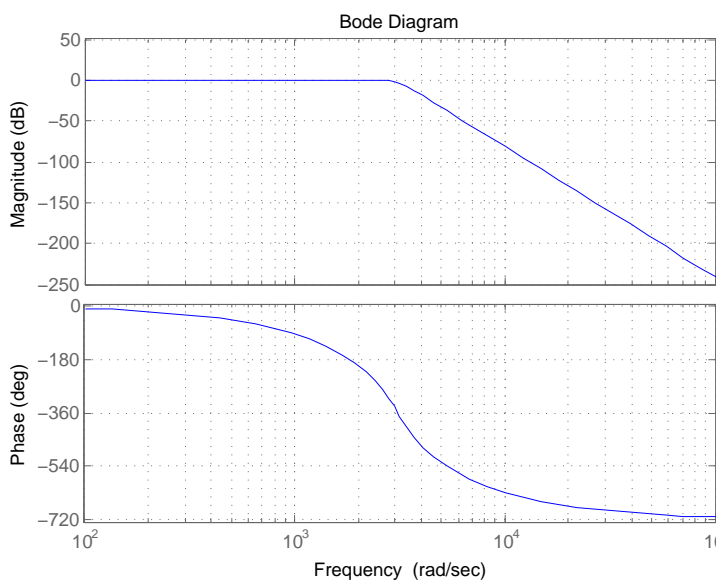


Abbildung 3: Butterworth Filter 2. Ordn.

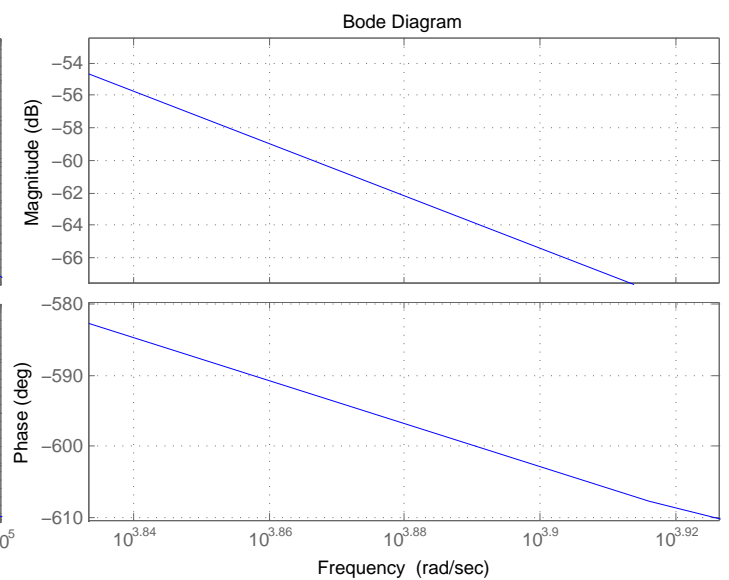


Abbildung 4: In die Grafik des Filters hineingezoomt

$$f_0 \approx 2 \cdot 7,3kHz = 14,6kHz \quad (4)$$

Mit diesem Tiefpass müssten wir mit mindestens  $14,6kHz$  abtasten. Da wir in diesem Versuch aber nur eine Spektralanalyse bis maximal  $f_m = 2kHz$  durchführen möchten, stören uns Aliasingeffekte über  $2kHz$  nicht, weshalb es genügt mit  $6kHz$  ab zu tasten.

### 3 Praxisteil

#### 3.1 Lineare Regression, Steigung und Offset der ADU-Kennlinie

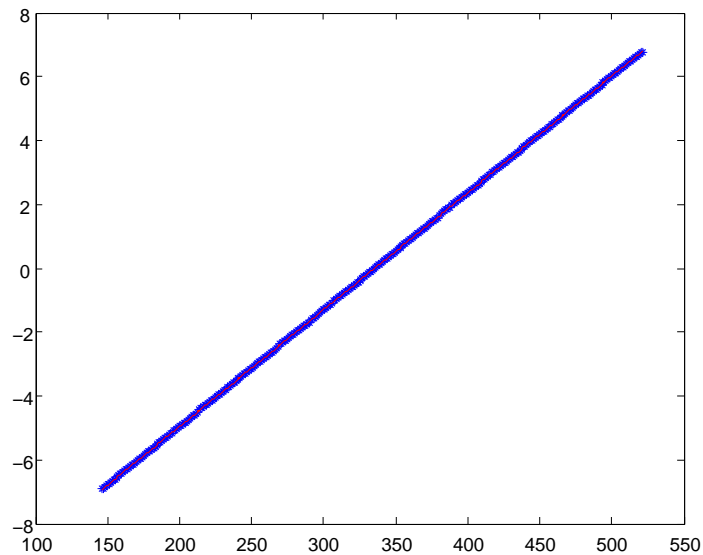


Abbildung 5: ADU-Kennlinie + Regressionsgerade

Der Offset beträgt  $-0.0959V$  und die Steigung ist  $0.0367$ .

#### 3.2 Rauschen

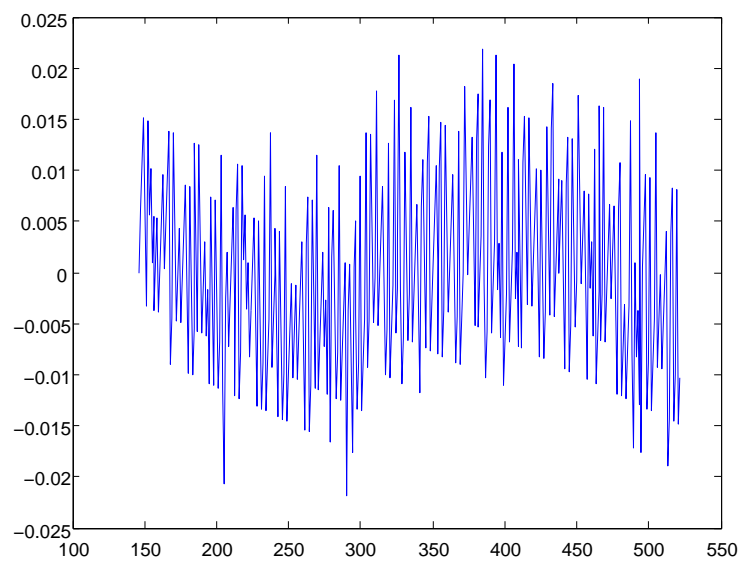


Abbildung 6: Rauschen

Der Effektivwert des Rauschens ist  $0.0091V$ . Das Singal- zu Rauschverhältnis beträgt 121.5515.

### 3.3 Amplitidenfrequenzgang des Antialiasing-Filters

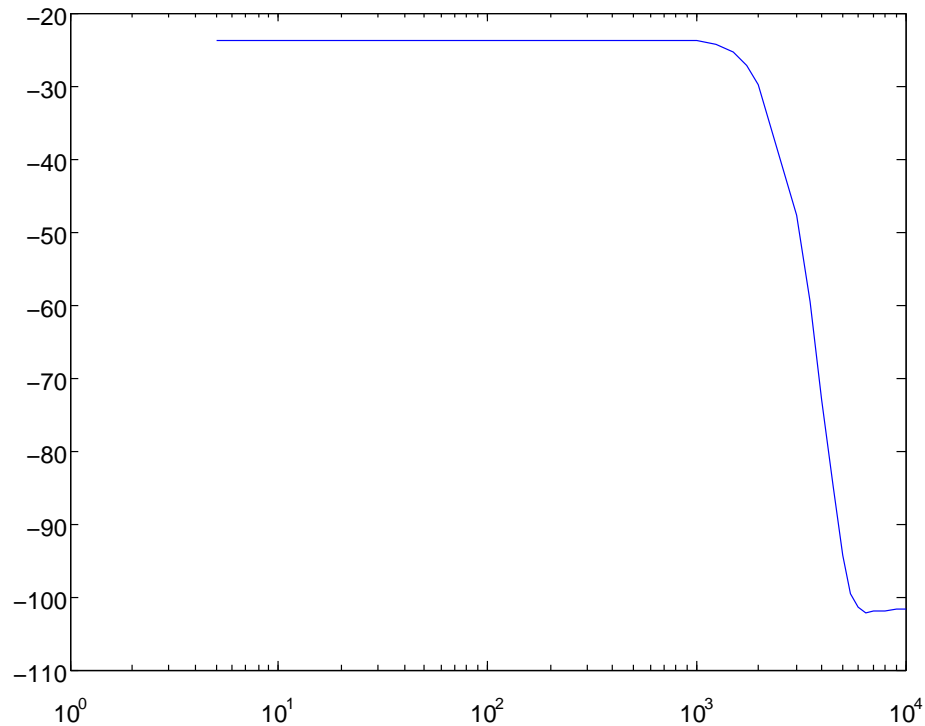


Abbildung 7: Amplitidenfrequenzgang des Antialiasing-Filters

Der Amplitidenfrequenzgang des realen Filters weicht an einigen Stellen von dem des simulierten, idealen ab.

Der Amplitudengang startet nicht bei  $0dB$ , da in der Leitung und im Filter Verluste entstehen.

Außerdem knickt der Reale Filter nicht so abrupt ab sondern hat einen weicheren Verlauf, was auf die nicht idealen Bauteileigenschaften zurückzuführen ist. Aus dem selben Grund ist die Steigung nicht ganz so steil wie in der Simulation.

Schließlich wird die Dämpfung nicht unendliche groß, da der ADU sehr kleine Spannungen nicht mehr auflösen kann.

### 3.4 Matlab-Code

Listing 1: Matlab-script

```
1 % Auslesen des ADU bzw. Einlesen der aufgenommenen Werte
2
3
4 %a=ucAnalogRead('COM14', 15000, 2000, 'NONE', 0)
5 load a;
6
```

```
7
8 % Konvertieren der digitalen ADU–Werte in Spannungen
9 b=Code2Volt(a);
10
11 % Ausschneiden einer Flanke
12 [val,min_ind]=min(b);
13 [val,max_ind]=max(b);
14 c=b(min_ind:max_ind);
15
16 % Plotten der Ausgeschnittenen Flanke
17 t=[min_ind:1:max_ind];
18 figure(1);
19 clf();
20 plot(t,c,'*');
21
22 % Regressionsgerade erstellen
23 n=polyfit(t',c,1);
24 d=polyval(n,t');
25
26 % Plotten der Flanke und der Regressionsgerade
27 figure(2);
28 clf();
29 plot(t',c,'*',t',d,'r-');
30
31 % Die Steigung ermitteln
32 m=(c(250)-c(200))/(250-200);
33
34 % Den Offset ermitteln
35 Mitte_ind=length(c)/2;
36 of=c(Mitte_ind);
37
38
39
40
41
42 % Rauschen: Differenzsignal aus Regression und Signal
43 r=c-d;
44
45 % Plotten des Rauschen
46 figure(3);
47 clf();
48 plot(t',r);
49
50 % Effektivwert des Rauschens
51 r_eff=sqrt(mean(r.^2));
52
53 % Effektivwert des Signals
54 c_eff=sqrt(mean(b.^2));
55
56 % Berechnung des Signal–Rausch–Verhältnisses
57 SNR=20*log(c_eff/r_eff);
58
59 % Amplitudengang des Tiefpassfilters
60 f=[5 1000 1250 1500 1750 2000 3000 3500 4000 4500 5000 5500 6000 6500 7000 8000 9000 10000];
61 H=[4.5827 4.5599 4.4335 4.1968 3.8557 3.3739 1.3839 0.7583 0.3883 0.2162 0.1333 0.1035 0.0941 0.0903 0.0915 0.0915
    0.0930 0.0933]./15;
62 Amp=20*log(H);
63
64 % Plotten des Amplitudengangs
```

```
65 figure(4);
66 clf();
67 b = mkfilter(3100/(2*pi),8,'butterw');
68 bode(b,'-');
69 grid on;
70 semilogx(f,Amp)
```

Listing 2: Matlab-script

```
1 %function [ voltage ] = Code2Volt (Code )
2 %
3 %filename : Bin2Volt .m
4 %author : Ihr_Name
5 %organization : TU Berlin
6 %project : MDV PR
7 %date : Tag_der_Bearbeitung
8 %
9 %description : rechnet die vom ADU ausgegebenen Zahlencodes
10 %in Spannungen ( Einheit : V ) um
11 %input : Code : vom ADU ausgegebene Zahlencodes
12 %output : [ v o l t a g e ] : b e r e c h n e t e Spannungen [V]
13
14
15 function [voltage] = Code2Volt(Code);
16
17 n = 7/511;
18
19 voltage = Code * n;
20 return;
```