



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

Praktikum Messdatenverarbeitung  
Betreuer: José-Luis Bote-Garcia  
SS2018

# **Versuchsprotokoll 3**

## **Spektralanalyse I - DFT**

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

14. Juni 2018

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Praktikumsaufgaben</b>	<b>1</b>
2.1	Aufgabe 1 . . . . .	1
2.2	Aufgabe 2 . . . . .	1
2.3	Aufgabe 3 . . . . .	4
2.4	Aufgabe 4 . . . . .	8
<b>3</b>	<b>Abgabeaufgaben</b>	<b>9</b>

# 1 Einleitung

In diesem Labor wird der Strom eines Verbrauchers untersucht. Der Strom wird anhand einer Phasenanschnittsdimmer gesteuert mit einem Anschnittswinkel zwischen  $0^\circ$  und  $180^\circ$ . In dem folgenden Teil "Praktikumsaufgaben" wird die Spektralanalyse der im Labor aufgenommene Ströme mit verschiedene Anschnittswinkeln zu sehen.

## 2 Praktikumsaufgaben

### 2.1 Aufgabe 1

In der ersten Aufgabe ist es das Ziel, eine Schaltung aufzubauen, mit der der Strom im Dimmerschaltkreis aufgenommen werden kann. Dafür verwenden wir den Wandler-Box als Antialiasing-Filter (also ein Tiefpassfilter) und den programmierten Mikrocontroller mit der Anschlussplatine. Dabei ist es wichtig, die Sicherheitskabel zu benutzen, wenn die Schaltung mit 230V (Netzspannung) versorgt ist. Mit dem Mikrocontroller und dem Matlab-Skript `ucAnalogRead()` werden dann die Messdaten aufgenommen und für die Verarbeitung mit der Matlab-Funktion `Code2Volt()` vom digitalen Bereich (Binärcodes) zum analogen Bereich (Spannungsbereich) transformiert. Die aufgenommenen Messdaten und die Abbildungen sind in der Aufgabe 2 der Praktikumsaufgaben weiter untersucht.

### 2.2 Aufgabe 2

Für das Versuchsprotokoll 3 wurden 3 Messungen mit unterschiedlichen Phasenanschnitten aufgenommen, nämlich Ströme mit Phasenanschnitten 1.2ms, 3.4ms, 6.2ms (Werte sind mit dem Oszilloskop ermittelt). Die Signale im Zeitbereich sind in der Abbildung 1 dargestellt.

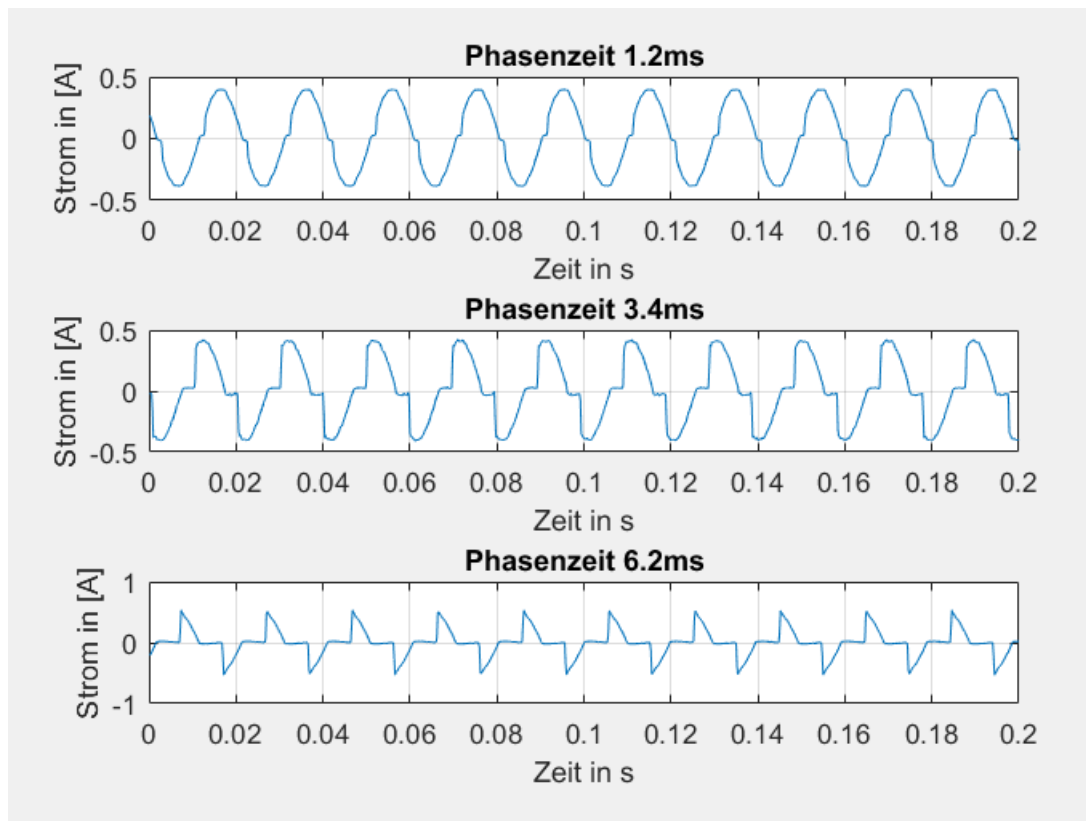


Abbildung 1: Stromsignale mit unterschiedlicher Phasenanschnitten - Zeitbereich

Übrigens, die mit dem Oszilloskop ermittelten Phasenanschnitten sind nicht unbedingt fehlerfrei. Der Oszilloskop hat intern keine Methode, um die Phasenzeit zu bestimmen und deswegen verwendet man Cursors, um Abstände des Nullstroms zu ermitteln. Der Matlab-Skript im Listing 1 berechnet den Phasenanschnitt mit einem übergebenen Toleranzbereich, wo der Strom voraussichtlich null wird. Wenn die Abbildung 1 untersucht wird, ergibt sich als Toleranzbereich  $[-0.05\text{A}, 0.05\text{A}]$ , da der Strom in dem Bereich nahe Null ist.

```

1 function [alpha, phaszeit] =
    stromanschnittswinkel(signal, sigfreq)
2
3 %   Berechnet die Anschnittswinkel aus den aufgenommenen Daten
4 %   signal ist der Vektor, der den Stromverlauf enthält
5 %   alpha gibt den Anschnittswinkel vom signal
6 %   phaszeit gibt die Dauer des Anschnitts in millisekunden
7 %   In unserer Messungen:
8 %   sigfreq = 50 Hz
9 %   anzahlmesswerte = 1000
10 %   abtastfreq = 5000 Hz

```

```
11
12
13
14 werte = Code2Volt(signal,65.5549,-9.8077); %Messungen in
    Spannungswerte rueckwandeln
15 strom = werte'.*(0.9/10); %Mit Stromfaktor multiplizieren
16 anzahlmesswerte = length(signal);
17 abtastfreq = sigfreq*anzahlmesswerte;
18 signaldauer = anzahlmesswerte*(1/abtastfreq);
19 periodenanzahl = signaldauer/(1/sigfreq); %gesamte ganzzahlige
    Periodenanzahl
20
21 counter = 1;
22 nulltime = 0;
23
24 while(counter<(length(strom)/(periodenanzahl*2)))
    %(length(strom)/(periodenanzahl*2)), gibt die halbe
    Periodendauer
25 if(-0.05<strom(counter)) && (strom(counter)<0.05) %
    Toleranzbereich ueberpruefen
26
27 nulltime = nulltime+1;
28 end
29 counter = counter+1;
30 end
31
32 phaszeit = 1000*nulltime * (1/abtastfreq);
33 alpha = ((nulltime*(1/abtastfreq))/((1/sigfreq)/2))*180;
34 end
```

Listing 1: Code für die Bestimmung der Phasenanschnitten

Der Vergleich der aus dem Oszilloskop ermittelten Werten und den Werten aus dem Matlab-Skript sieht man in der Abbildung 2.

Am Oszilloskop gemessene Phasenwinkel von 3 Signalen sind:

Fuer das Signal mit 1.2 ms Phasenanschnittszeit =

21.6000

im Matlab berechnete Phasenanschnittszeit und Winkel

1.5400

27.7200

Fuer das Signal mit 3.4 ms Phasenanschnittszeit

61.2000

im Matlab berechnete Phasenanschnittszeit und Winkel

3.4000

61.2000

Fuer das Signal mit 6.2 ms Phasenanschnittszeit

111.6000

im Matlab berechnete Phasenanschnittszeit und Winkel

5.7800

104.0400

Abbildung 2: Phasenanschnitten, Phasenwinkel und Phasenzeit - Oszilloskop und Matlab-Skript

## 2.3 Aufgabe 3

Für die Aufgabe 3 werden die drei Stromsignale mit Hilfe der Diskreten Fouriertransformation untersucht, indem die Amplituden- und Phasenspektrum bestimmt werden. Außerdem muss man beachten, dass die aufgenommene Anzahl von Signalperioden ganzzahlig sind, damit keine Leckeffekte auftreten. Mit der Abtastfrequenz von 5000Hz und der Periodendauer  $T = 20\text{ms}$  (Netzfrequenz ist 50Hz) werden 1000 Samples aufgenommen, welches eine ganzzahlige Anzahl von Signalperioden entspricht.

Für die Frequenzganganalyse ist der Code im Listing 2 verwendet, welcher für das Praktikum 4 entwickelt wurde.

```

1 function [BetragXf,PhaseXf]=Spektrum
    (xt,wn,startfenster,fs,LogFlag)
2 % function [BetragXf,PhaseXf]=Spektrum(xt,wn,fs,LogFlag,n)
3 % description: Bestimmt und plottet das Betrags- und
4 %               Phasenspektrum eines Signals xt
5 % input:        xt - Signal im Zeitbereich
6 %               wn - Fensterfolge hann(L) , blackman(L)
7 %               fs - Abtastfrequenz
8 %               LogFlag - Flag = 1 y-Achse fuer das
9 %               Betragsspektrum wird logarithmisch
10 %               in dB dargestellt, sonst 0
11 %
12 % output:       BetragXf - Betragsspektrum des Signals xt
13 %               PhaseXf - Phasenspektrum des Signals xt
14 %               Bildschirm - Plotten des Betrags-
15 %               und Phasenspektrums
16
17 vec_window = zeros(1,length(xt));
18 vec_window(startfenster:startfenster+length(wn)-1) = wn;
19 signalfenster = xt'.* vec_window;
20 anzahlmesswerte = length(xt);
21 %f = fs*(0:(anzahlmesswerte/2))/anzahlmesswerte; % Frequenz-Achse
22 f = (0:anzahlmesswerte-1)*((2*fs)/length(anzahlmesswerte));
23
24 anzahlfenster = sum(wn);
25
26 DFT = fft(signalfenster);
27 Amp = abs(DFT)/anzahlfenster;
28
29 if(LogFlag == 1)
30     BetragXf = 20*log10(Amp);
31     PhaseXf = unwrap(angle(DFT));
32 else
33     BetragXf = Amp;
34     PhaseXf = unwrap(angle(DFT));
35 end
36
37 subplot(2,1,1)
38 plot(f,BetragXf);
39 title('Magnitude');
40 xlabel('Frequency in Hz');
41 ylabel('Amplitudengang');

```

```

42 % ax = gca;
43 % ax.XTick = [5 15 30 45 60 75 90 95];
44 grid on;
45
46 subplot(2,1,2)
47 plot(f,PhaseXf);
48 title('Phase');
49 xlabel('Frequency in Hz');
50 ylabel('Phasengang');
51 % ax = gca;
52 % ax.XTick = [5 15 30 45 60 75 90 95];
53 grid on;
54
55 end

```

Listing 2: Code für Frequenzganganalyse

Die Abbildungen 3, 4 und 5 zeigen jeweils die Frequenzganganalyse für die abgelesenen Phasenzeiten 1.2ms, 3.4ms und 6.2ms.

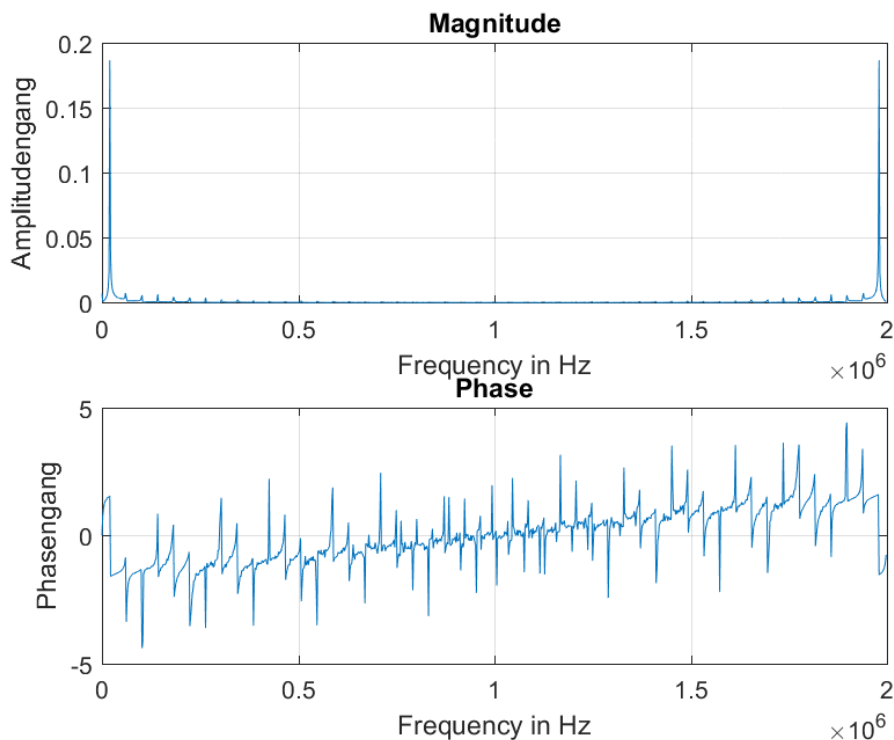


Abbildung 3: Frequenzganganalyse für Stromsignal mit Phasenzeit 1.2ms



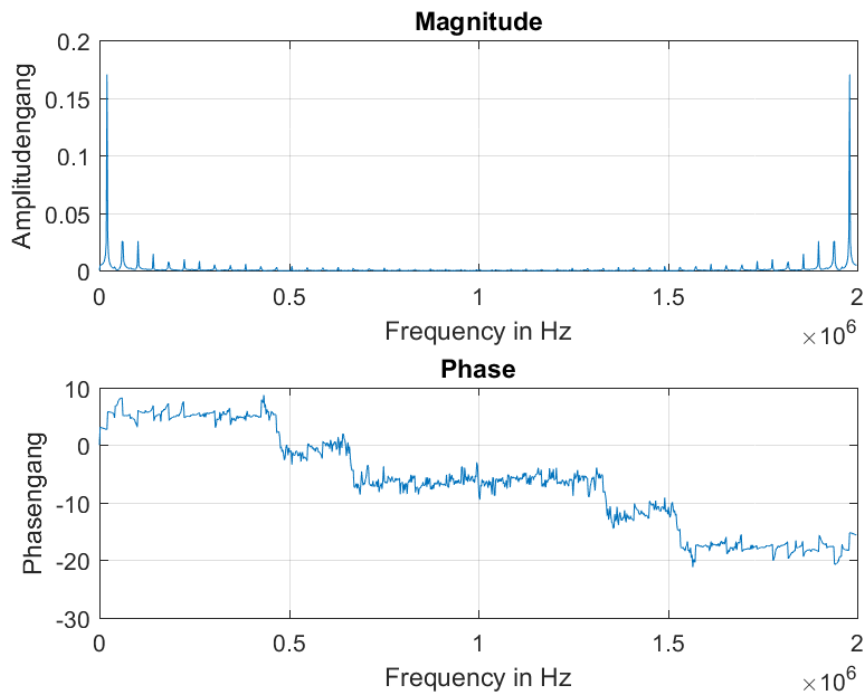


Abbildung 4: Frequenzganganalyse für Stromsignal mit Phasenzeit 3.4ms

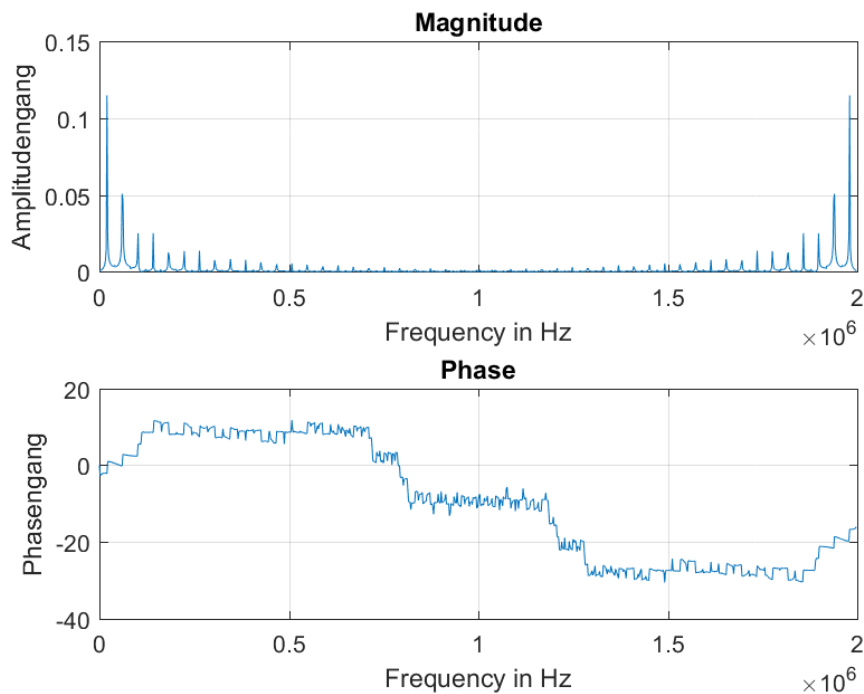


Abbildung 5: Frequenzganganalyse für Stromsignal mit Phasenzeit 6.2ms

Die Abbildungen zeigen jeweils einen Amplitudengang und einen Phasengang. Die Diskrete Fouriertransform ermöglicht bzw. erleichtert die Untersuchung der Frequenzanteile eines Signals. Einen Dimmerschaltkreis funktioniert damit, dass die Schaltung in einem bestimmten Zeitraum den Stromfluss vermindert, bzw. null setzt. Das verursacht aber tatsächlich Oberschwingungen (also Frequenzanteile, die von der Grundfrequenz abweichen), welches man mit der Fouriertransformation beobachten kann.

Also, in der Abbildung 3 ist die Phasenzeit als 1.2ms vorgegeben und dementsprechend sind Oberschwingungen nur begrenzt vorhanden. In der Abbildung 4 bzw. 5 sind Oberschwingungen deutlicher zu beobachten, welche aufgrund der Verzerrung wegen dem Dimmerschaltkreis auftreten. Es ist jedoch an der Stelle wichtig zu erläutern, dass aufgrund des gausschen Rauschens (White-Noise) in jeder Schaltung bzw. Messung Oberschwingungen entstehen, welche nicht unbedingt wegen einer Schaltungseigenschaft entstanden sind.

## 2.4 Aufgabe 4

Der Effektivwert im Zeitbereich ergibt sich im digitalen Werten als RMS (root mean square) aller Werten im Signal:

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \dots + x_n^2)} \quad n - \text{Anzahl von Samples} \quad (1)$$

Wobei der Effektivwert im Frequenzbereich sich mit dem Parsevalschen Theorem berechnen lässt:

$$\sum_{n=1}^N x^2[n] = \frac{1}{N} \sum_{m=1}^N |X[m]|^2 \quad n - \text{Anzahl von Samples} \quad (2)$$

Die beiden Formeln liefert identische Ergebnisse.

Die Matlab-Implementierung beider Formeln sind im Listing 3 und 4 dargestellt.

```
1 function RMSxt = EffektivwertZeitbereich(xt)
2
3 RMSxt = sqrt((1/length(xt))*sum(power(xt,2)));
4
5 end
```

Listing 3: Code für Effektivwert im Zeitbereich

```

1 function RMSIf = EffektivwertFourier(xt)
2
3 DFTxt = fft(xt);
4 RMSIf = (1/length(DFTxt)) *sqrt(sum(power(abs(DFTxt),2)));
5
6 end

```

Listing 4: Code für Effektivwert im Frequenzbereich

	TrueRMS	EffektivwertZeitbereich	EffektivwertFourier
Stromsignal PZ=1.2ms	0.275332	0.2758	0.2758
Stromsignal PZ=3.4ms	0.253212	0.2634	0.2634
Stromsignal PZ=6.2ms	0.185541	0.2139	0.2139

Tabelle 1: Vergleich der TrueRMS, EffektivwertZeitbereich und EffektivwertFourier

Die Tabelle 1 zeigt den Vergleich der drei Methoden, um den Effektivwert des Stromsignals zu berechnen. Wie es schon vermutet gewesen ist, liefern die Effektivwertberechnungen im Zeitbereich und Frequenzbereich identische Ergebnisse. Jedoch abweichen diese Ergebnisse von der Ergebnis der TrueRMS-Funktionalität des Multimeters. Diese Tatsache kann dadurch erklärt werden, dass während der Aufnahme der Messdaten Fehler bzw. Abweichungen auftreten kann, welche infolge die Ergebnisse beeinflussen. Ein wichtiger Punkt ist, dass die Effektivwerte mit steigender Phasenzeit immer weniger werden, wobei der Effekt zwischen dem Stromsignal mit Phasenzeit 3.4ms und dem Stromsignal mit Phasenzeit 6.2ms am größten ist. Der Grund dafür ist, dass der Dimmerschaltkreis die Leistung in der Schaltung verkleinert, und den Strom in bestimmten Zeiträume null setzt. Deswegen ist der Effektivwert eines verzerrten Stromes am Ende weniger.

### 3 Abgabeaufgaben

Die Abgabeaufgaben sind in dem Protokoll ziemlich ähnlich (teilweise identisch) zu den Praktikumsaufgaben. Aus diesem Grund sind die Antworten für die Abgabeaufgaben gleich im Praktikumsteil gegeben.

## Literatur

- [1] Prof. Dr.-Ing. Clemens Gühmann  
*Skript zur Messtechnik*, TU Berlin, WS 2016