



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

Praktikum Messdatenverarbeitung

Termin 4

Dirk Babendererde (321 836)

Thomas Kapa (325 219)

Magdalene Busuru (319 433)

14. November 2012

Gruppe: G2 Fr 08-10

Betreuer: Jürgen Funk

Inhaltsverzeichnis

1	Vorbereitungsaufgaben	1
1.1	Phasenanschnittsfunktion	1
1.2	Matlab-Code	2

1 Vorbereitungsaufgaben

1.1 Phasenanschnittsfunktion

Testen Sie die Phasenanschnittsfunktion mit verschiedenen (simulierten) Stromsignalen bei verschiedenen Anschnittswinkeln zwischen $\alpha = 0^\circ$ und $\alpha = 90^\circ$. Vergleichen Sie die Effektivwerte im Zeit- und Frequenzbereich.

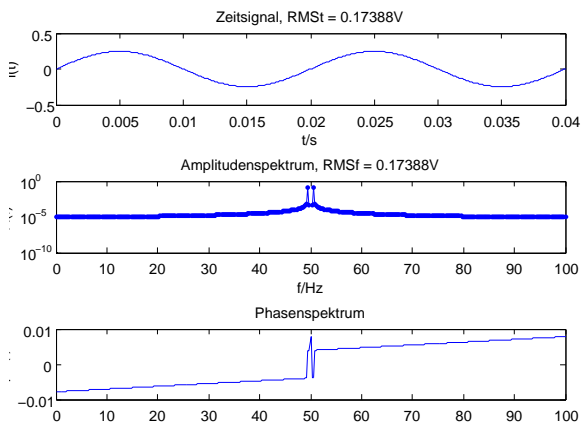


Abbildung 1: Simulierte sinus-Funktion mit $\alpha = 0 \cdot 90^\circ$

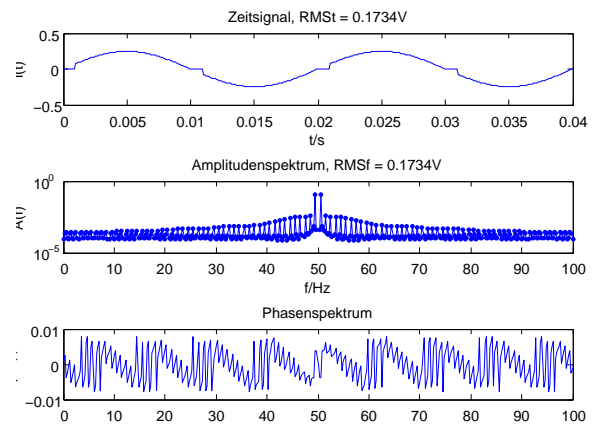


Abbildung 2: Simuliert Sinus-Funktion mit $\alpha = 0,1 \cdot 90^\circ$

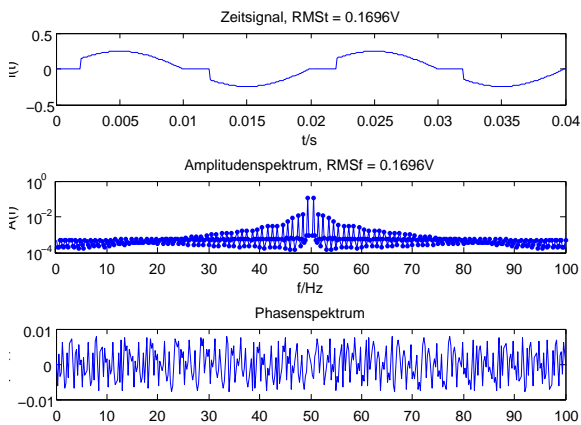


Abbildung 3: Simulierte sinus-Funktion mit $\alpha = 0,2 \cdot 90^\circ$

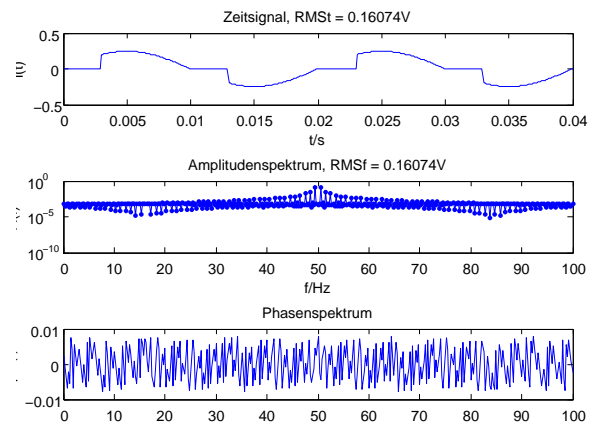


Abbildung 4: Simuliert Sinus-Funktion mit $\alpha = 0,3 \cdot 90^\circ$

1 VORBEREITUNGSAUFGABEN

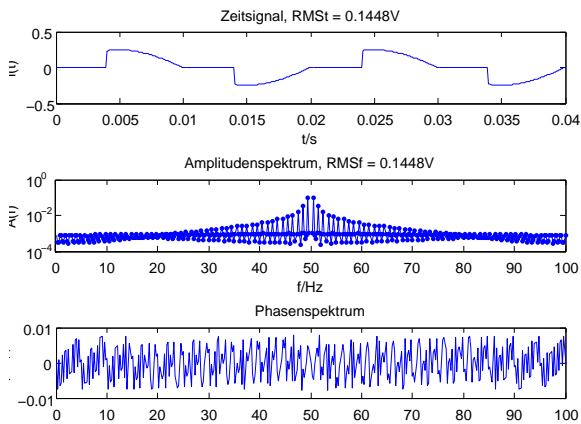


Abbildung 5: Simulierte sinus-Funktion mit $\alpha = 0,4 \cdot 90^\circ$

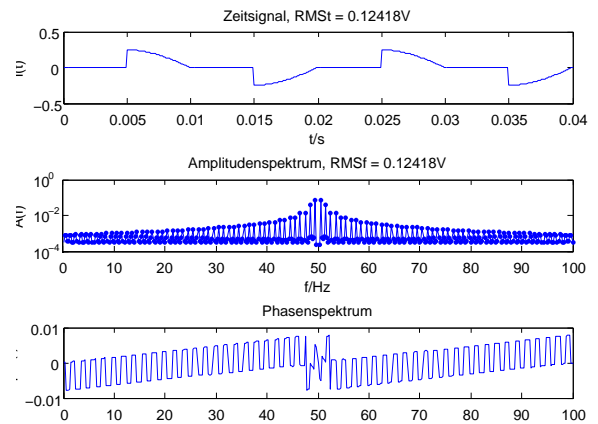


Abbildung 6: Simuliert Sinus-Funktion mit $\alpha = 0,5 \cdot 90^\circ$

Es ist zu beobachten, dass der Effektivwert mit zunehmendem Alpha sinkt und die Lampe somit gedimmt wird.
Die RMS Werte im Zeit- und Frequenzbereich sind jeweils die selben.

1.2 Matlab-Code

Listing 1: Matlab-script

```
1 clear
2
3 t1 = 0:0.0001:1;
4
5 U = 325%230*sqrt(2);
6 R = 1320;
7 f = 50;
8 T = 1/f;
9 alpha = 0 *pi;
10
11
12 t1 = [0:0.0001:2*T];
13
14 strom = stromPhasSchnitt(U/R,f,alpha,t1);
15
16 figure(1)
17 plot(t1,strom)
18
19
20
21
22 %Dauer des Signals:
23 T_ges = 2*T;
24 %Abtastfrequenz:
25 f_T = 2*f;
26
27
28
29 % RMS
30 RMSt = EffektivwertZeitbereich(strom)
```

```

31 RMSf = EffektivwertFourier(strom)
32
33
34
35
36
37
38 figure(1)
39 y = strom;
40 %Berechnung des Spektrums
41 y_DFT = fft(y);
42 y_DFT = fftshift(y_DFT);
43 N = length(y);
44 %Betragsspektrum
45 y_DFT_abs = abs(y_DFT)/N;
46 [val, min] = min(y_DFT_abs);
47 y_DFT_abs(min) = 0;
48 %Phasenspektrum
49 y_DFT_phase = unwrap(angle(y_DFT)/N);
50 %Zeitachse
51 t = T_ges*(0:(N-1))/(N-1);
52 %Frequenzachse
53 f_DFT = f_T*(0:(N-1))/(N-1);
54 %neues Fenster zur Darstellung
55 figure(1);
56 %Darstellung des Zeitsignals
57 subplot(3,1,1);
58 plot(t,y);
59 % ylim([-3 3]);
60 % xlim([0 0.05]);
61 title(['Zeitsignal', RMSf, 'V']);
62 xlabel('t/s');
63 ylabel('i(t)');
64 %Darstellung des Amplitudenspektrums
65 subplot(3,1,2);
66 semilogy(f_DFT, y_DFT_abs, '-');
67 title(['Amplitudenspektrum', RMSf, 'V']);
68 % xlim([99.5 100.5]);
69 % ylim([10^(-3) max(y)]);
70 xlabel('f/Hz');
71 ylabel('A(f)');
72 %Darstellung des Phasenspektrums
73 subplot(3,1,3);
74 plot(f_DFT, y_DFT_phase);
75 %xlim([99.5 100.5]);
76 title('Phasenspektrum');
77 xlabel('f/Hz');
78 ylabel('phi(f)');

```

Listing 2: Matlab-script

```

1 function i=stromPhasSchnitt(id,f,alpha,t)
2 % function i=stromPhasSchnitt(id, f, alpha, t)
3 %
4
5 % filename: stromPhasSchnitt.m
6 % author: Dirk Babendererde (321 836), Thomas Kapa (325 219), Magdalene
7 % Busuru (319 433)
8 % organisation: TU Berlin

```

```

8 % project: MDV PR
9 % date: 10.05.2012
10 %
11 % description: simuliert einen mit dem Winkel alpha
12 % angeschnittenen Strom
13 % input: id      Amplitude des Stroms
14 %         f      frequenz des Stroms
15 %         alpha   Ansnittswinkel [rad]
16 %         t      Vektor mit den Zeitpunkten, zu denen der Strom
17 %                simuliert wird
18 % output: [i]   zeitlicher Verlauf des Stromes
19 %
20
21 % alpha in [0,1]
22 a = alpha/pi;
23
24 % Periode
25 T = 1/f;
26
27 % erst einmal ein Sinus ausgeben
28 i = id * sin(2*pi*f*t);
29
30
31 % der Sinus wird auf numm gesetzt an den Stellen wo die Phase
32 % angeschnitten ist
33 k = 1;
34 while (k < length(t))
35     if (mod(t(k),T/2) < a*T/2)
36         i(k) = 0;
37
38     end
39     k = k + 1;
40 end

```

Listing 3: Matlab-script

```

1 function RMSxt = EffektivwertZeitbereich(xt)
2 % function RMSxt = EffektivwertZeitbereich(xt)
3 %
4
5 % filename:EffektivwertZeitbereich.m
6 % author: Dirk Babendererde (321 836), Thomas Kapa (325 219), Magdalene
7 % Busuru (319 433)
8 % organisation: TU Berlin
9 % project: MDV PR
10 % date: 10.05.2012
11 %
12
13 % description: Bestimmt den Effektivwert im Zeitbereich
14 % eines Signals xt
15 % input: xt: Signalim Zeitbereich
16 %
17 % output: RMSxt: Effektivwert des Signals

```

```
17  
18 RMSxt = sqrt(mean(xt.^2));
```

Listing 4: Matlab-script

```
1 function RMSIf = EffektivwertFourier(xt)  
2 % function RMSIf = EffektivwertFourie(xt)  
3 %  
  
4 % filename:EffektivwertFourier.m  
5 % author: Dirk Babendererde (321 836), Thomas Kapa (325 219), Magdalene  
6 % Busuru (319 433)  
7 % organisation: TU Berlin  
8 % project: MDV PR  
9 % date: 10.05.2012  
10 %  
  
11 % description: Bestimmt im Frequenzbereich den Effektivwert  
12 % eines Signals  
13 % input: xt: Signal im Zeitbereich  
14 %  
15 % output: RMSIf: Effektivwert des Signals, berechnet  
16 % aus den Fourierkoeffizienten  
17  
18 RMSIf = sqrt(mean(abs(fftshift(fft(xt))).^2)/length(xt));
```