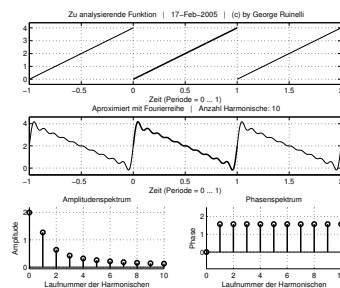


Elektrotechnik Praktikum

Versuch 3.4

Fourieranalyse und -Synthese



Zusammenfassung

Das Fourierspektrum eines periodischen Signals zu berechnen ist mit geeigneten Mitteln leicht realisierbar. Die Messung dieser Koeffizienten ist jedoch nur sehr aufwändig und mit teuren Messgeräten möglich.

Die Signalleistung eines periodischen Signals ist gleich der Summe der Signalleistungen ihrer einzelnen Teil-schwingungen. Sie lässt sich somit einfach berechnen, wenn man die Amplitudenkoeffizienten der einzelnen Harmonischen kennt. Die Phasenkoeffizienten tragen nichts zur Signalleistung bei.

Bei den Signalquellen gibt es massive Unterschiede zwischen den Signalqualitäten (Klirrfaktor). Wenn für einen Versuch ein möglichst klirrfaktor-freies Sinus-Signal benötigt wird, sollte ein Signalgenerator oder ein Low Distortion Generator verwendet werden.

Versuchsdatum: 18. Januar 2005 (Letzte Überarbeitung: 17. Februar 2005)

Versuchsleiter: George Ruinelli <george@ruinelli.ch>

Assistent: Ersan Egeli

Klasse: ET2a

Fach: Elektrotechnik

Dozent: M. Schlup <spm@zhwin.ch>

Der Messbericht ergab die Note 5.5

Er wurde vollständig in \LaTeX unter Linux erstellt (Bilder: OpenOffice).

Der Quellcode sowie dieses Dokument sind unter <http://www.ruinelli.ch> frei erhältlich.

Sie dürfen frei verwendet werden, solange das Copyright erhalten bleibt.

1 Aufgabenstellungen und Zielsetzungen

1.1 Aufgabe 1

Es soll ein Matlab-Programm entwickelt werden, welchem man eine periodische Funktion als Formel sowie die Anzahl Harmonischen übergeben kann. Das Programm soll daraus die Amplituden- sowie die Phasen-Koeffizienten der Fourierreihe berechnen und in einer Grafik ausgeben.

Im gleichen Grafik-Ausdruck soll die Original-Funktion der mit einer Fourierreihe angenäherte Funktion gegenübergestellt werden.

Mit geeigneten Hilfsmitteln soll das Amplitudenspektrum eines Sägezahn-Signal messtechnisch aufgenommen werden. Mit dem Matlab-Programm aus obiger Aufgabe sollen dann die gemessenen Werte verifiziert werden..

1.2 Aufgabe 2

Es soll die Parsevalsche Beziehung überprüft werden, indem von mehreren periodischen Signalen die Wirkleistung über einem Widerstand ermittelt wird.

Die Summe dieser Werte soll dann mit der Wirkleistung der Überlagerung dieser Signale verglichen werden.

1.3 Aufgabe 3

Mit einem Klirrfaktor-Messgerät sollen die Sinus-Signale von verschiedenen Signalquellen untersucht und miteinander verglichen werden.

2 Aufgabe 1

2.1 Berechnung mit Matlab

2.1.1 Matlab-Programm

Siehe Anhang A.1.

2.1.2 Berechnete Koeffizienten

Als zu analysierende Funktion wurde ein Sägezahn mit DC-Offset gewählt.

Die maximale Amplitude betrug 2 V und der DC-Offset 2 V. Die Frequenz war 1 kHz.

In der Tabelle wurden die ersten 10 Harmonischen aufgelistet:

Laufnummer der Harmonischen	Amplitudenspektrum	Phasenspektrum
0	2	0
1	1.27	1.57
2	0.64	1.57
3	0.42	1.57
4	0.32	1.57
5	0.25	1.57
6	0.21	1.57
7	0.18	1.57
8	0.16	1.57
9	0.14	1.57
10	0.13	1.57

Tabelle 1: Berechnete Werte aus Matlab-Programm A.1

2.1.3 Grafik-Ausdruck

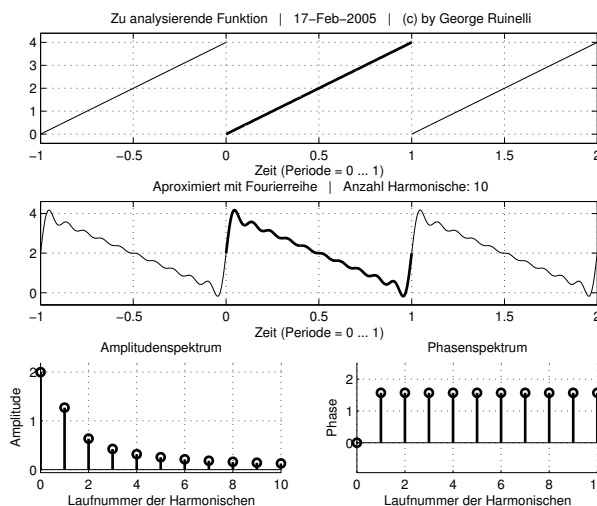


Abbildung 1: Matlab-Ausdruck aus Matlab-Programm A.1

Die integrierte Periode des Signals ist dick ausgezogen.

Damit der eigentliche Signalverlauf besser erkennbar ist, wurden insgesamt 3 Perioden aufgezeichnet.

2.2 Messung

Das Sägezahn-Signal wurde mit TinaLab II generiert.

Für die Messung wurde der Spektrumanalysator aus TinaLab II verwendet.

2.2.1 Messschaltung

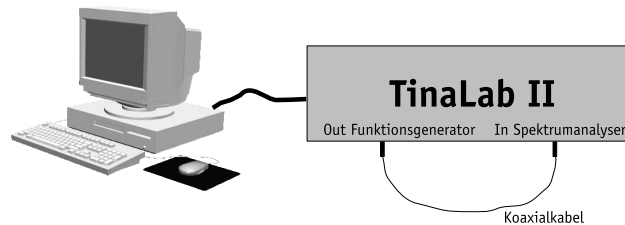


Abbildung 2: Messschaltung

2.2.2 Matlab-Programm

Siehe Anhang A.2.

2.2.3 Gemessene Koeffizienten

Laufnummer der Harmonischen	Amplitudenspektrum in V		Phasenspektrum in ° berechnet
	berechnet	gemessen	
0	2	2.1 ± 0.3	0
1	1.27	1.3 ± 0.2	1.57
2	0.63	0.59 ± 0.09	1.57
3	0.42	0.36 ± 0.06	1.57
4	0.32	0.29 ± 0.05	1.57
5	0.25	0.24 ± 0.04	1.57
6	0.21	0.19 ± 0.03	1.57
7	0.18	0.15 ± 0.03	1.57
8	0.16	0.12 ± 0.02	1.57
9	0.14	0.10 ± 0.02	1.57
10	0.13	0.09 ± 0.02	1.57

Tabelle 2: Berechnete sowie gemessene Werte aus Matlab-Programm A.2

2.2.4 Fehlerbetrachtung und Auswertung

Für die Toleranz von TinaLab II bei der Signalgenerierung, sowie bei der Messung wird eine Genauigkeit von etwa 14% angenommen. Ebenfalls nicht vernachlässigt werden darf die Ablesegenauigkeit beim Amplitudenspektrum aus dem Diagramm (Bildschirm, ebenfalls etwa 1%). Somit ergibt sich eine Toleranz von 15%.

Die Messwerte bis zur 7. Harmonischen sind noch knapp in der Toleranz, die Werte der höheren Harmonischen liegen aber mit bis zu 20% Abweichung ganz klar ausserhalb.

Für das Phasenspektrum lieferte Spektrumanalysator aus TinaLab II nur unbrauchbare Werte, sie wurden deshalb gar nicht erst in die Tabelle aufgenommen.

Für aussagekräftigere Messungen müsste hier ein deutlich besseres Messgerät aus einer höheren Preisklasse verwendet werden.

Um eine einigermaßen aussagekräftige Synthese durchführen zu können, wurden die *gemessenen* Amplituden und die *berechneten* Phasenwerte verwendet!

Die Resultate dieser Messung müssen deshalb vorsichtig interpretiert werden.

2.2.5 Grafik-Ausdruck

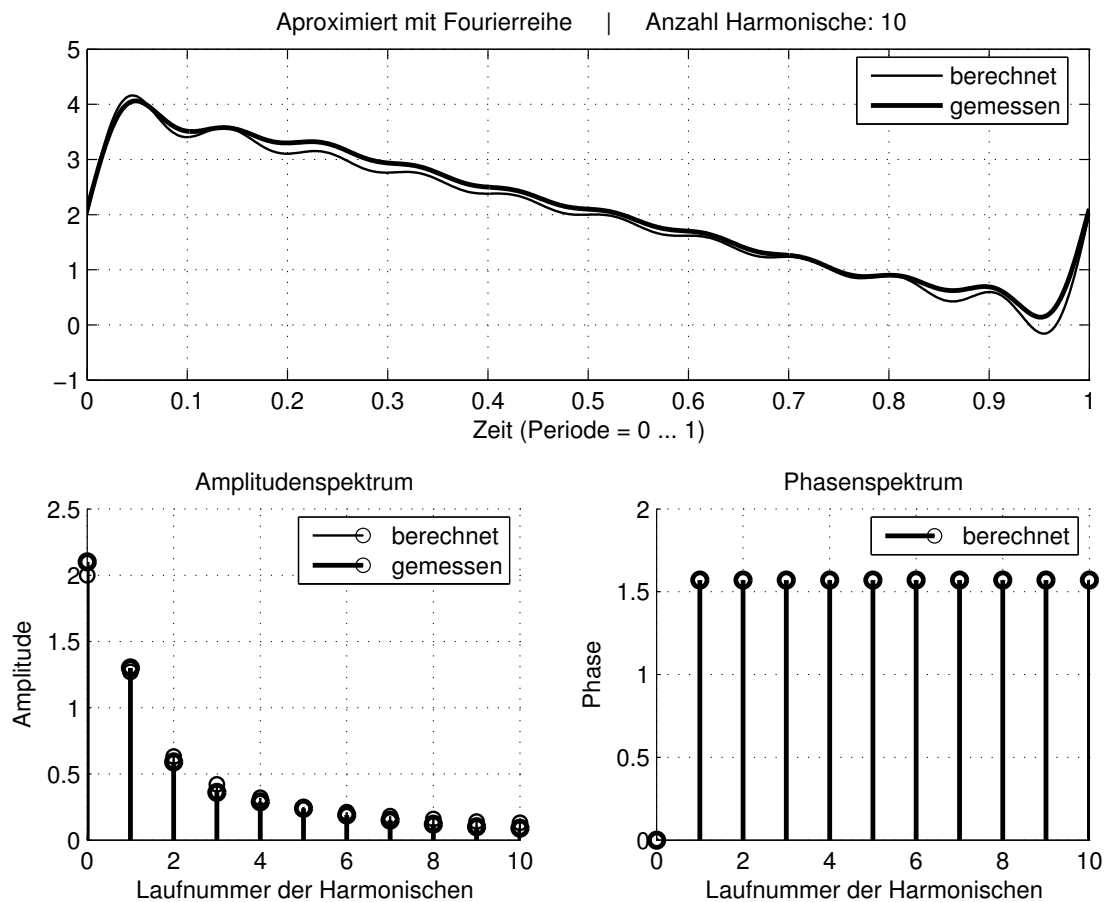


Abbildung 3: Matlab-Ausdruck aus Matlab-Programm A.2

3 Aufgabe 2

Die Parsevalsche Beziehung besagt, dass die Leistung eines periodischen (nicht-harmonischen) Signals der Summe der Signalleistungen der einzelnen Teilschwingungen entspricht.

Da keine potentialfreien Signalgeneratoren zur Verfügung standen, konnte die Messung nur theoretisch durchgeführt werden. Dies wurde mit einer Simulation in Tina gemacht.

3.1 Versuchsaufbau in Tina

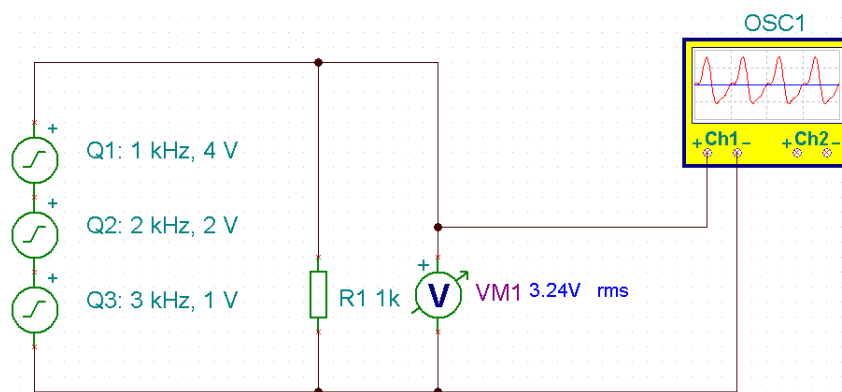


Abbildung 4: Versuchsaufbau in Tina

3.2 Ermittelte Werte

Aktive Quellen	Spannung über R1	Berechnete Leistung über R1
Q1: 1 kHz, 4 V	2,83 V	8,01 mW
Q2: 2 kHz, 2 V	1,41 V	1,99 mW
Q3: 3 kHz, 1 V	0,707 V	0,50 mW
Q1 + Q2 + Q3	3,24 V	10,5 mW

Tabelle 3: Berechnete Werte aus Tina

Die Parsevalsche Beziehung konnte bestätigt werden durch simulierte Messungen (Tina berechnet die Werte im Hintergrund) der Leistungen an einem Widerstand von mehreren Sinus-Signalen, sowie der Überlagerungen dieser Signale.

Dabei wurde die Phasenlagen der Signale variiert. Es konnten keine Unterschiede der Signalleistungen festgestellt werden. Die Signalleistungen sind also unabhängig von der Phasenlage der einzelnen Teilschwingungen.

Es ist leicht ersichtlich, dass die Summe der Einzelleistungen gleich der Leistung des überlagerten Signales ist:

$$P_{1R1} + P_{2R1} + P_{3R1} = P_{R1} \tag{1}$$

4 Aufgabe 3

Als Klirrfaktor-Messgerät konnte ein Distortion Meter (HM 8027) verwendet werden.

Dieses hat eine Messtoleranz von 0,01%.

Als Testsignal wurde jeweils ein Sinus mit der Frequenz 1 kHz und 1 V Amplitude verwendet.

In der Tabelle 4 sind die untersuchten Quellen sowie die gemessenen Werte aufgelistet:

Signalquelle	Bezeichnung	gemessener Klirrfaktor	in den Spezifikationen angegebener Klirrfaktor
Signalgenerator	HM 8032	0,02%	< 0,2%
Funktionsgenerator	HM 8030-5	0,36%	< 0,5%
HAMEG Low Distortion Generator	HM 8037	0,00%	< 0,01%
TinaLab II	-	0,06%	<i>keine Angaben</i>

Tabelle 4: *gemessene Werte*

Das HM 8032 ist speziell für hohe Signalqualitäten ausgelegt, gleiches gilt für das HM 8037, dessen Klirrfaktor mit der gegebenen Messeinrichtung nicht mehr messbar war.

Erstaunlich ist, dass selbst der Funktionsgenerator von TinaLab II sehr gut abschneidet, obwohl es in die Kategorie der Low Cost-Produkt gehört.

5 Verwendete Geräte

- Signalgenerator HM 8032
- Funktionsgenerator HM 8030-5
- Low Distortion Generator HM 8037
- TinaLab II
- Distortion Meter HM 8027

6 Anhang

- Matlab-Programm für Aufgabe 2.1
- Matlab-Programm für Aufgabe 2.2

Datum: _____

Unterschrift: _____

A Anhang

A.1 Matlab-Programm für Aufgabe 2.1

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Titel: Fourieranalyse und -Synthese
3  % Autor: George Ruinelli
4  % Datum: 17.01.05
5  % Funktion: Es kann eine periodische Funktion sowie die Anzahl Harmonischen
6  % angegeben werden. Die Funktion wird dann in Fourier-Koeffizienten
7  % zerlegt und daraus wieder eine Fourierreihe zusammengebaut.
8  % Die integrierte Periode ist blau, zur Hilfe wird noch eine Periode vorher
9  % sowie nachher dargestellt.
10 %
11 % KNOWN BUGS:
12 % Dieses Programm funktioniert nur in der Vollversion von Matlab, wegen der
13 % Funktions-Deklarationen, welche in der Studenten-Version ANDERS sind.
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15
16 clear all,clc,format compact,format short
17
18 '      Programm gestartet, bitte warten...'
19
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21 % EINGABEN: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22 A = 4; % Max. Amplitude des Signals
23 n = 10; % Anzahl Harmonische
24
25 %% LEGENDE:
26 %% T = Periode des Signals (geht von start bis start+T)
27 %% start = Untere Grenze der Integration (Integr. geht von start bis start+T)
28 %% F=@(t)( FUNKTION(t) );
29
30 % Zu analysierende Funktion (nicht verwendete auskommentieren):
31 % -----
32 T=1; start=0; F=@(t)( A/T*(t)); % Sägezahn ohne DC-Offset
33 %T=1; start=0; F=@(t)(round(t)*A ); % Rechteck mit DC-Offset
34 %T=2*pi; start=0; F=@(t)(1*sin(t)); % einfache Sinusschwingung
35 %T=2; start=0; F=@(t)(-abs(t-T/2)+1); % Dreieck, wie aus SiSy-Praktikum 4
36
37 % AB HIER NICHTS MEHR ÄNDERN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 t = linspace(start,start+T,1000); % Zeitbereich für Plots
40 omega=2*pi/T;
41
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
43 '      a- und b-Koeffizienten werden berechnen...'
44 for i=1:n
45     F_a=@(r) (F(r) .* cos(i*omega*r) ); % Hilfsfunktion für quadl
46     F_b=@(r) (F(r) .* sin(i*omega*r) ); % Hilfsfunktion für quadl
47     a(i) = 2/T*quadl(F_a,start,start+T);
48     b(i) = 2/T*quadl(F_b,start,start+T);
49     i % Ausgabe der gerade berechneten Harmonischen
50 end
51
52 a0 = 2/T*quad(F,start,start+T); % DC-Wert des Signals
53
54 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
55 '      a- und b-Koeffizienten werden umgewandelt in Amplituden- und Phasen-Koeffizienten...'
56 for i=1:n
57     A(i) = sqrt(a(i)^2 + b(i)^2);
58     phi(i) = (atan2(-b(i), a(i)));
59     if A(i)<0.01*A(1)    phi(i)=0;    end % Falls A(k) < (1% von A1) wird Phi(k)
60 end % auf 0 gesetzt, als Darstellungs-Optimierung.
61
62 A0 = a0/2; % DC-Wert des Signals
63
64 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65 % Ausgabe der Koeffizienten:
66 '      Die Koeffizienten sind: (Spalten: Laufnummer | a | b | A | Phi )'
67 M=[0,a0,0,A0,0];
68 for i=1:n
69     M= [M; [i, a(i), b(i), A(i), phi(i)]];
70 end
71
72 M % Koeffizientenmatrix ausgeben
73
74 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
75 '      Fourier-Reihe wird zusammengebaut aus A- und Phi-Koeffizienten...'
76 fourier_reihe=A0;
77 for i=1:n
78     fourier_reihe= fourier_reihe + A(i) * cos(i*omega * t - phi(i) );
79 end
80
81 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
82 % Hilfwerte für die Plots:
83 x_array=linspace(0,n,n+1);
84 A = [A0, A];
85 phi = [0, phi];
86
87 % Achsenbereiche berechnen:
88 delta_ampl_F = 0.1 * (max(F(t)) - min(F(t)));
89 achsen_F = [min(t)-T    max(t)+T    min(F(t))-delta_ampl_F    max(F(t))+delta_ampl_F];

```

```

90
91 delta_ampl_Four = 0.1 * (max(fourier_reihe) - min(fourier_reihe));
92 achsen_Four = [min(t)-T max(t)+T min(fourier_reihe)-delta_ampl_Four max(fourier_reihe)+delta_ampl_Four];
93
94 delta_Ampl_spec = 0.1 * (max(A) - min(A));
95 achsen_Ampl = [0 n min(A)-delta_Ampl_spec max(A)+delta_Ampl_spec];
96
97 delta_Phasen_spec = 0.5 * (max(A) - min(A));
98 achsen_Phasen = [0 n min(phi)-delta_Phasen_spec max(phi)+delta_Phasen_spec];
99
100 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
101 % Plots:
102 figure(1);
103 subplot(3,2, [1 2]),
104 plot(t-T,F(t),'-k'); % 1 Periode vorher
105 hold on
106 plot(t+T,F(t),'-k'); % 1 Periode nachher
107 plot(t,F(t),'-k','LineWidth',2); % eigentliche Periode
108 hold off
109 axis(achsen_F);
110 xlabel(['Zeit (Periode = 0 ... ',num2str(T),')'])
111 title(['Zu analysierende Funktion | ', date, ' | (c) by George Ruinelli'])
112 grid
113
114 subplot(3, 2, [3 4]),
115 plot(t-T, fourier_reihe,'-k'); % 1 Periode vorher
116 hold on
117 plot(t+T, fourier_reihe,'-k'); % 1 Periode nachher
118 plot(t, fourier_reihe,'-k','LineWidth',2); % eigentliche Periode
119 hold off
120 axis(achsen_Four);
121 xlabel(['Zeit (Periode = 0 ... ',num2str(T),')'])
122 title(['Aproximiert mit Fourierreihe | Anzahl Harmonische: ', num2str(n) ])
123 grid
124
125 subplot(3, 2, [5]), stem(x_array, abs(A),'k','LineWidth',2);
126 axis(achsen_Ampl);
127 xlabel('Laufnummer der Harmonischen')
128 ylabel('Amplitude')
129 title(['Amplitudenspektrum' ])
130 grid
131
132 subplot(3, 2, [6]), stem(x_array, phi,'k','LineWidth',2);
133 axis(achsen_Phasen);
134 xlabel('Laufnummer der Harmonischen')
135 ylabel('Phase')
136 title(['Phasenspektrum' ])
137 grid
138
139 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
140 % Ausgabe in TEX-Datei für Dokumentation:
141 M=[];
142 for i=1:n+1
143     M= [M; [(i-1), round(A(i)*100)/100, round(phi(i)*100)/100]];
144 end
145 matrix2tex('tabelle_aufgabela.tex',M);
146
147 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
148 ' Fertig'

```

Listing 1: Matlab-Programm für Aufgabe 2.1

A.2 Matlab-Programm für Aufgabe 2.2

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Titel: Fouriersynthese
3  % Autor: George Ruinelli
4  % Datum: 18.01.05
5  % Funktion: Es können die Amplituden- und Phasenkoeffizienten angegeben
6  % werden. Daraus wird dann mit einer Fouriersynthese wieder eine Fourierreihe
7  % zusammengesetzt.
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 clear all,clc,format compact
11
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 % EINGABEN: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 T = 1; % Periode des Signals
15 % Berechnete Werte:
16 % Arrays mit A0, A1, A2, A3,... bzw. Phi0, Phi1, Phi2, Phi3,...:
17 A_b=[2.00 1.27 0.63 0.42 0.32 0.25 0.21 0.18 0.16 0.14 0.13]; % Amplituden
18 phi_b = [0 1.57 1.57 1.57 1.57 1.57 1.57 1.57 1.57 1.57 1.57]; % Phasen
19
20 % Gemessene Werte:
21 A_m=[2.1 1.3 0.59 0.36 0.29 0.24 0.19 0.15 0.12 0.10 0.09 ]; % Amplituden
22
23 % AB HIER NICHTS MEHR ÄNDERN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 t = linspace(0,T,1000); % Zeitbereich für Plots
26 omega=2*pi/T;
27
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 % Fourier-Reihen werden zusammengesetzt aus A- und Phi-Koeffizienten
30 n=length(A_b);
31 fourier_reihe_b=A_b(1);
32 for i=2:n
33     fourier_reihe_b= fourier_reihe_b + A_b(i) * cos((i-1)*omega * t - phi_b(i)) ;
34 end
35
36 n=length(A_m);
37 fourier_reihe_m=A_m(1);
38 for i=2:n
39     fourier_reihe_m= fourier_reihe_m + A_m(i) * cos((i-1)*omega * t - phi_b(i)) ;
40 end
41
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
43 % Ausgabe in TEX-Datei für Dokumentation:
44 M=[];
45 for i=1:n
46     M= [M; [(i-1), round(A_b(i)*100)/100, round(A_m(i)*100)/100, round(phi_b(i)*100)/100]];
47 end
48 matrix2tex('tabelle_aufgabelb.tex',M);
49
50 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51 % Hilfswerte für die Plots:
52 x_array=linspace(0,n-1,n);
53
54 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
55 % Plots:
56 figure(2);
57 subplot(2, 2, [1 2]),
58 plot(t, fourier_reihe_b, '-k', 'LineWidth', 1);
59 hold on
60 plot(t, fourier_reihe_m, '-k', 'LineWidth', 2);
61 hold off
62
63 xlabel(['Zeit (Periode = 0 ... ', num2str(T), ')'])
64 title(['Aproximiert mit Fourierreihe | Anzahl Harmonische: ', num2str(n-1) ])
65 legend('berechnet', 'gemessen');
66 grid
67
68 subplot(2, 2, [3]),
69 stem(x_array, abs(A_b), '-k', 'LineWidth', 1);
70 hold on
71 stem(x_array, abs(A_m), '-k', 'LineWidth', 2);
72 hold off
73 xlabel('Laufnummer der Harmonischen')
74 ylabel('Amplitude')
75 title(['Amplitudenspektrum '])
76 legend('berechnet', 'gemessen');
77 grid
78
79 subplot(2, 2, [4]),
80 stem(x_array, phi_b, '-k', 'LineWidth', 2);
81 xlabel('Laufnummer der Harmonischen')
82 ylabel('Phase')
83 title(['Phasenspektrum '])
84 legend('berechnet');
85 grid
86
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88 ' Fertig'

```

Listing 2: Matlab-Programm für Aufgabe 2.2