



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

Praktikum Messdatenverarbeitung
Betreuer: José-Luis Bote-Garcia
SS2018

Versuchsprotokoll 7

Instationäre Signale Kurzzeit-Fourieranalyse

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

12. Juli 2018

Inhaltsverzeichnis

1	Einleitung	1
2	Praktikumsaufgaben	2
2.1	Aufnahme des Signals	2
2.2	Drehzahl des Motors	2
2.3	Analyse des zeitveränderlichen Stromsignals	4
2.4	Vergleich der Drehzahlen	10
3	Abgabeaufgaben	13
3.1	Chirp-Signal	13
3.2	Aufgabe 2	16
3.3	Aufgabe 3	16

1 Einleitung

In diesem Labor wird ein Gleichspannung angesteuerter Universalmotor verwendet, um eine praktische Anwendung der Kurzzeit-Fourier-Transformation zu betrachten. Im Motorstrom gibt es nicht nur erhöhte Amplitudenwerte bei den einfachen Vielfachen der Drehfrequenz, also nicht nur bei reguläre Oberschwingungen, sondern bei dem 18- und dem 36-fachen der Drehfrequenz. Der Grund dafür ist die Anzahl von Nuten und Lamellen am Komutator. Vom Tachosignal ist eine direkte Ermittlung der Drehzahl möglich aber wie bei dem Motorstrom, gibt das Tachosignal die Vielfache der Drehfrequenz und zwar die 8-fache. Das Tachosignal und der Motorstrom werden während eines Hochlaufs des Motors aufgenommen. Letztendlich wird das Spectrogramm des aufgenommenen Stromsignals erzeugt und das Stromsignal analysiert.

2 Praktikumsaufgaben

2.1 Aufnahme des Signals

Für die Aufnahme des Tachosignals und den Motorstroms wird eine Meskette aufgebaut. Dazu wird in den PC integrierte Messkarte und die Matlab-Funktion datenaufnahme benutzt. Die Signale werden während eines Hochlaufs aufgenommen. Es wird eigentlich mehrere Signale aufgenommen (z.B. das Luftschall mittels ein Mikrofon und das Körperschall) aber in diesem Praktikum wird nur das Tachosignal und Motorstrom gebraucht.

2.2 Drehzahl des Motors

Die Drehzahl des Motors wird im Zeitbereich mit Hilfe der Matlab-Funktion freqganginzeit, siehe Listing 1, und im Frequenzbereich mit Hilfe der Matlab-Funktion rpmtachoinfreq, siehe Listing 2, berechnet.

```
1 function [freqgang, nullgang, zeit] = freqganginzeit(sig, t)
2 %Eingänge; sig = Eingangssignal, t = Zeitvektor des Signals
3 %Ausgänge; nullgang = Vektor, der die index der Nullgänge
   enthält
4 %           freqgang = Vektor, der die Frequenz f für ein Period
   eines
5 %           Sinus-Signals enthält
6 signal = sig;
7 i = 0;
8 for k = 1:1:length(signal)-1; %Index der Nulldurchgänge finden
9                               %und in einem Vektor nullgang
                               speichern
10    if(signal(k) < 0 && signal(k+1) > 0)
11        i = i+1;
12        nullgang(i) = k;
13    end
14 end
15
16 %Plot zur Kontrolle der Nulldurchgänge
17 plot(t, signal);
18 hold on;
19 for k = 1:1:length(nullgang)
20     plot(t(nullgang(k)), signal(nullgang(k)), 'x')
21     hold on;
```

```

22 end
23 xlabel('Zeit_in_s');
24 ylabel('Amplitude');
25
26 for k = 1:1:length(nullgang)/2
27     %mittels Nulldurchgaenge die Periodendauer
28     %jede Sin-Signal berechnen
29     nullzwei = nullgang(2*k); %Ende eines Sin-Signals
30     nullerst = nullgang(2*k -1); %Anfang eines Sin-Signals
31     periodendauer = t(nullzwei) - t(nullerst);%Berechnet die Zeit
        zwischen dem
32     zeit(k)=t(nullerst);
        %Anfang und Ende eines Sin-Signals
33
        %Also die
        Periodendauer
34     freqgang(k) = 1/periodendauer; %Die Periodendauer in Frequenz
        umwandeln
35
        %und in dem Vektor freqgang
        speichern
36 end
37 end

```

Listing 1: Matlab-Funktion freqgangingzeit

```

1 function drehzahl = rpmtachoinfreq(tachosig,fs)
2 %Plottet das Amplitudenspektrum von ein Tachosignal
3 %Berechnet den Drehzahl aus dem Tachosignal
4
5 DFT = fft(tachosig);
6 n = length(tachosig);
7 Amptacho = abs(DFT)/n;
8 f = (0:n-1)*(fs/(n-1));
9 plot(f,Amptacho);
10 title('Amplitudenspektrum_fuer_Motortacho_1');
11 xlabel('Frequency_in_Hz');
12 ylabel('Amplitude');
13
14 [maxtacho indexmaxtacho] = max(Amptacho);
15 drehzahl = round(60*(f(indexmaxtacho)/8));
16 end

```

Listing 2: Matlab-Funktion rpmtachoinfreq

Um die Drehzahl in dem Zeitbereich zu finden wird von der Funktion `freqgang` die Frequenzgang des Motors ermittelt und davon die momentare Drehfrequenz. Weil das Tachosignal mit einem Faktor von 8 erhalten ist, muss die gefundene Drehfrequenz auch durch 8 dividiert. Dann wird die Drehfrequenz mit 60 multipliziert um die Drehzahl in eine Minute zu finden. An der anderen Seite um die Drehzahl in dem Frequenzbereich zu finden wird das Amplitudenspektrum des Tachosignals gebildet und die maximale Wert davon gefunden. Die maximale Amplitude in dem Spektrum des Tachosignals liegt wieder an der 8-Fache der Drehfrequenz. Die Drehzahl wird mit der gefundene momentare Drehfrequenz, wie in dem Zeitbereich, berechnet. Die Ergebnisse sind an der Tabelle 1 dargestellt. Die berechnete Werte im Zeitbereich und im Frequenzbereich sind sehr nah zu einander. Die Drehzahl im Zeitbereich wird eigentlich anhand der Nulldurchgänge des Tachosignals im Zeitbereich berechnet. Die entsprechenden Nulldurchgänge sind nicht immer genau, besonders bei größere Versorgungsspannungen. Deshalb war es erwartet, dass die Berechnungen im Zeitbereich ungenauer würden aber wie in der Tabelle gesehen, es gibt eigentlich eine kleine Unterschied zwischen die Werte. Trotzdem die Ergebnisse, die mittels dem Amplitudenspektrum erhalten sind mehr zuverlässig.

Versorgungsspannungen	10V	20V	30V	40V	50V
Im Zeitbereich	1262	2876	4196	5394	6435
Im Frequenzbereich	1250	2843	4168	5345	6408

Tabelle 1: Drehzahl des Motors pro Minute

2.3 Analyse des zeitveränderlichen Stromsignals

Bei der dritten Praktikumsaufgabe ist die Analyse des zeitveränderlichen Stromsignals beim Hochlauf des Motors verlangt. Mit dem MATLAB-Befehl `spectrogram` kann die Kurzzeit-Fourier-Transformation des Stromsignals durchgeführt werden, und anschließend als Spectrogram dargestellt werden.

Bei der Analyse ist jedoch wichtig, dass geeignete Fensterlängen, Fensterarten und Überlappungen ausgewählt werden, damit die wichtige Informationen am Ende aus der Darstellung abgeleitet werden können. Deswegen ist es wichtig, den Befehl komplett zu verstehen:

spectrogram(x>window,noverlap,nfft,fs)

- x: Input-Signal
- window: Fensterlänge und Fensterart - Ein Vektor mit einer Fensterfunktion, zum Beispiel Hanning, Blackman, Rechteck. Integer ohne Fensterfunktion resultiert in einer Fensterung mit Hanning.
- noverlap: Integer, Anzahl von Samples, die sich zwischen Segmenten überlappen. Ohne Angabe resultiert in Überlappung mit %50.
- nfft: Anzahl von DFT-Punkte, ein Maß für Frequenzauflösung.
- fs: Abtastfrequenz

Also, mit dem Befehl spectrogram wird das Input-Signal mit einer Fensterung bestimmter Länge in gleich großen Segmente unterteilt, die dann anschließend mit FFT transformiert sind. Die freie Auswahl der Parameter stellt ein Gleichgewicht zwischen Frequenzauflösung und Zeitlokalization dar.

Nach Anwendung und Bedarf sind diese Parameter anders zu einstellen und es ist schwer analytisch vorherzusagen, welche Einstellung beweisbar besser ist. Trotzdem gibt es empirisch gefundene Regeln, die einen guten Startpunkt leisten:

- Parameter *window* - *Fensterlänge*: Eine längere Fensterung ermöglicht eine höhere Frequenzauflösung, im Gegensatz dazu eine kleinere Fensterung dient zur besseren Zeitlokalization.
- Parameter *window* - *Fensterart*: Die Auswahl der Art der Fensterung liegt lediglich an der Art des zu analysierenden Signals. Einen Überblick ist in der Abbildung 3 zu sehen (genommen aus DSP Stack Exchange , Original von National Instruments. Eine Hanning-Fensterung reicht für den größten Teil der Anwendungen, und bietet sich als der Startpunkt an. Es ist an der Stelle wichtig zu erwähnen, dass der Hanning-Fenster auch für unbekannte Signale geeignet ist.
- Parameter *noverlap*: Die Auswahl der Überlappung ermöglicht die Anpassung der Frequenzauflösung. Ein vernünftiges Spectrogram soll eine gewisse Überlappung zwischen Segmenten besitzen.
- Parameter *nfft*: Die Anzahl der DFT-Punkten hängt mit der Frequenzauflösung und Details auf der Frequenzachse proportional zusammen.

- Parameter f_s : Die richtige Eingabe der Abtastfrequenz resultiert in einer richtigen Zeitachse.

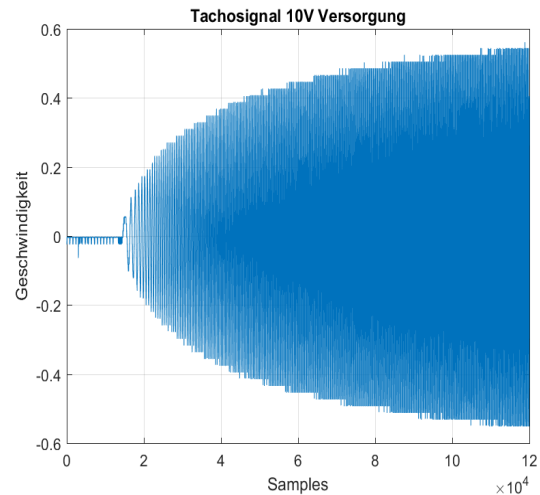
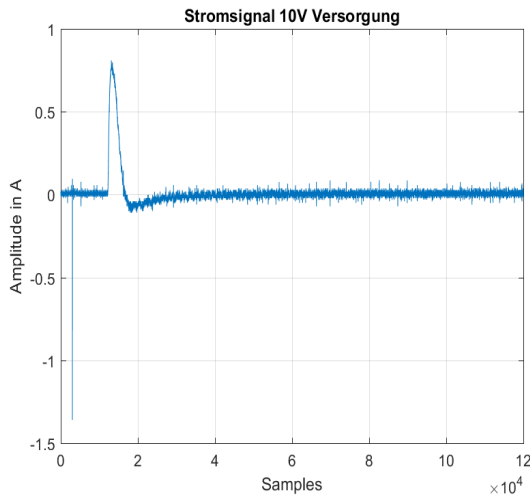


Abbildung 1: Stromsignal - Versorgung 10V Abbildung 2: Tachosignal - Versorgung 10V

Für die Analyse der Stromsignalen mit unterschiedlicher Versorgungsspannungen muss eine deutliche Zeitlokalization sichergestellt werden, denn die Darstellung des Stromsignals zeigt, dass der Verlauf mit einem Hochlauf, danach mit einem Unterschwing und dann mit hochfrequenten Rauschen um Null charakterisiert ist. Diese Tatsache macht Sinn, denn der Hochlauf des Motors charakterisiert den stationeren Ausgang, und der Unterschwing ist das Ausgleichen (siehe Abbildung 1).

Die Parameter sind dann als folgendes ausgewählt: Hanning-Fenster, $N = 10$, Overlap-Samples = 8, $nfft = 256$, $f_s = 40000\text{Hz}$. Damit ist es festgestellt, dass durch die kleinere Fensterlänge Zeitlokalization im Vordergrund ist, aber auch durch höhere NFFT und Überlappung die Frequenzauflösung auch nicht fehlend ist.

Type of Signal	Window
Transients whose duration is shorter than the length of the window	Rectangular
Transients whose duration is longer than the length of the window	Exponential, Hanning
General-purpose applications	Hanning
Spectral analysis (frequency-response measurements)	Hanning (for random excitation), Rectangular (for pseudorandom excitation)
Separation of two tones with frequencies very close to each other but with widely differing amplitudes	Kaiser-Bessel
Separation of two tones with frequencies very close to each other but with almost equal amplitudes	Rectangular
Accurate single-tone amplitude measurements	Flat top
Sine wave or combination of sine waves	Hanning
Sine wave and amplitude accuracy is important	Flat top
Narrowband random signal (vibration data)	Hanning
Broadband random (white noise)	Uniform
Closely spaced sine waves	Uniform, Hamming
Excitation signals (hammer blow)	Force
Response signals	Exponential
Unknown content	Hanning

Abbildung 3: Überblick der Fenster-Funktionen

Im Gegensatz zu dem Motorstromsignal, ist der Verlauf des Tachosignals sinusförmig. Das Signal besteht aus mehrere Sinussignale, die unterschiedliche Frequenzen und Amplituden besitzen. Deswegen muss bei dem Spectrogram die Frequenzauflösung im Vordergrund stehen (siehe Abbildung 2). Die Abbildungen 4 bis 9 zeigen jeweils die Spectrograms der Strom- und Tachosignalen für jeweils Versorgungsspannungen 10V, 30V und 50V.

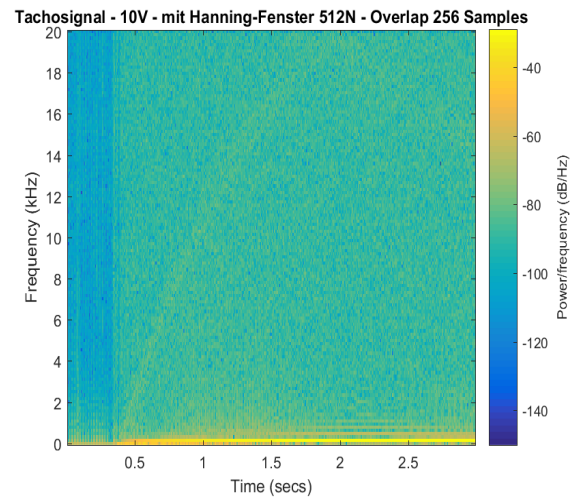
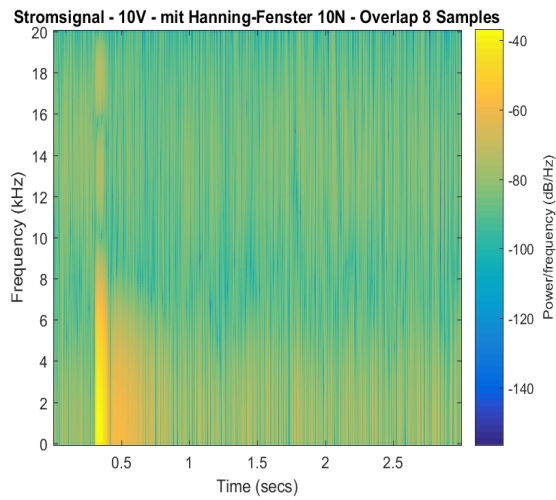


Abbildung 4: Stromsignal - Versorgung 10V -Abbildung 5: Tachosignal - Versorgung 10V -
Spectrogram Spectrogram

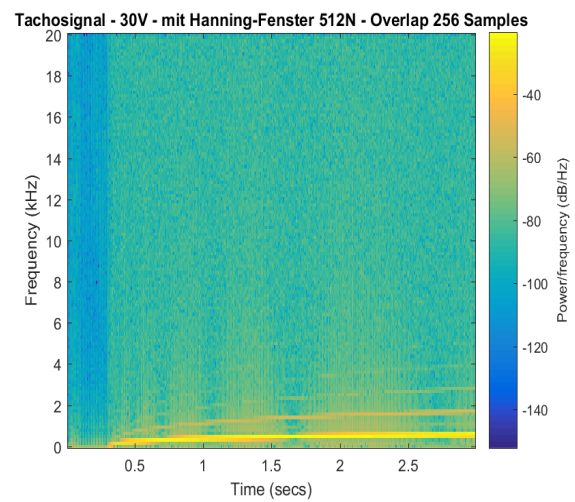
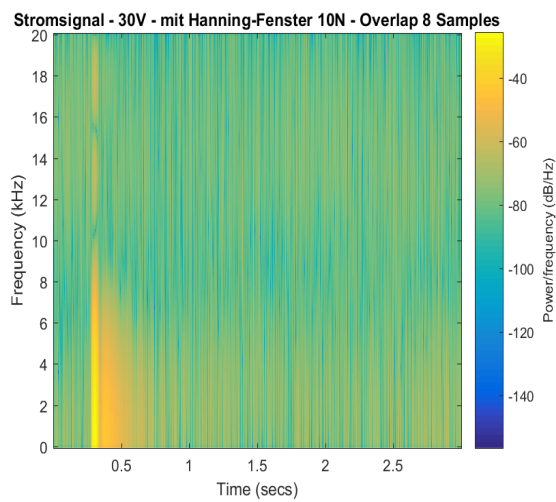


Abbildung 6: Stromsignal - Versorgung 30V -Abbildung 7: Tachosignal - Versorgung 30V -
Spectrogram Spectrogram

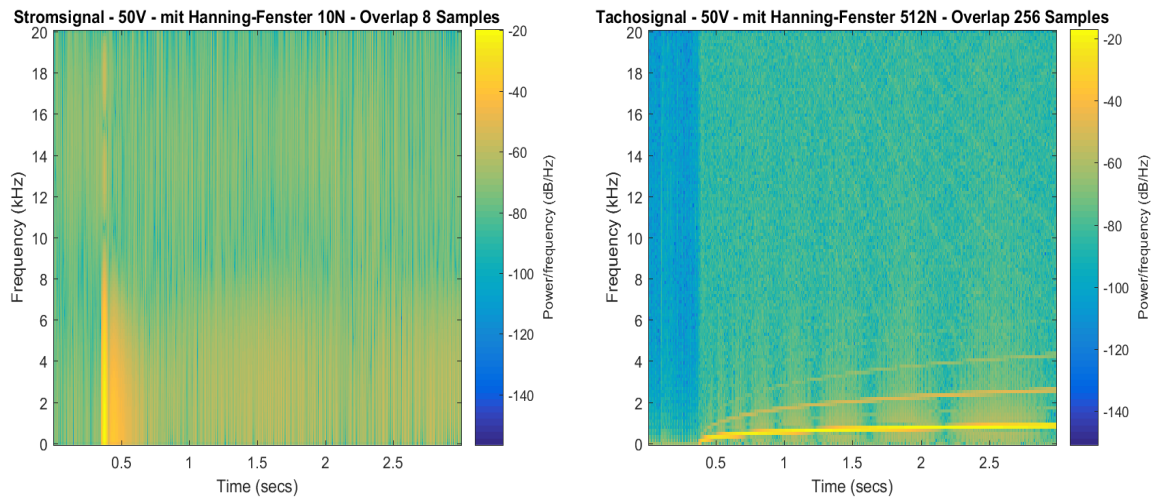


Abbildung 8: Stromsignal - Versorgung 50V -Abbildung 9: Tachosignal - Versorgung 50V -
Spectrogram Spectrogram

Die Abbildungen 5, 7 und 9 zeigen jeweils die Spectrograms für die Tachosignale, die sinusförmig sind. Die Analyse zeigt dann, dass die Frequenz des sinusförmigen Signals durch eine Sättigungskurve charakterisiert werden kann. Die Frequenz nimmt auf, bis der Motor die Endgeschwindigkeit erreicht hat, und dann verhält sich stationär. Selbstverständlich sind die Endfrequenzen bei jedem Tachosignal anders, weil die Versorgungsspannung direkt proportional zu der Geschwindigkeit des Motors steht.

Die Abbildungen 4, 6 und 8 zeigen im Gegenteil bessere Zeitlokalization. Das heißt, man kann sehen, dass es am Anfang (also zwischen circa 0.3 Sekunden bis 0.5 Sekunden) hochfrequente Anteile gibt, jedoch kann man nicht graphisch ermitteln, genau welche Frequenzanteile es gibt (alle Frequenzen von 0 Hz bis etwa 10 kHz sind dabei enthalten).

2.4 Vergleich der Drehzahlen

Die zeitveränderliche Drehzahl wird von dem Spektrogramm mit der Matlab-Funktion SpecToFreq abgelesen. Die Funktion ist in der Listing 3 zu sehen. Mit dieser Funktion wird die maximale Amplitude für jeder aufgeteilter Bereich gefunden und die entsprechende Drehfrequenz für jeder Bereich wird gespeichert.

```

1 function output = SpecToFreq(signal, window, noverlap, f ,fs)
2     [s, f, t] = spectrogram(signal, window, noverlap,f, fs);
3
4     dummy_vec = zeros(1,length(s(:,1))); %zeros for number of
        rows
5     for i=1:length(s(1,:)) % i = 1:number of columns
6         val = s(:,i);      % i-tes Column
7         abs_val = abs(val); % abs of complex number
8         [~, id] = max(abs_val);
9         frequenz = f(id);
10        dummy_vec(i) = round(frequenz);
11    end
12
13    output = dummy_vec;
14 end

```

Listing 3: Matlab-Funktion SpecToFreq

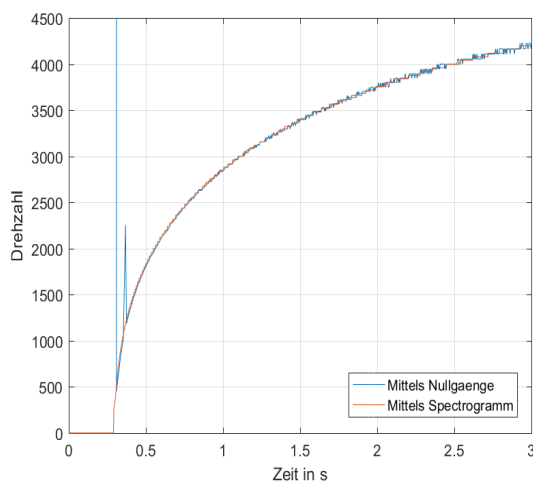


Abbildung 10: Für die Versorgungsspannung 30V

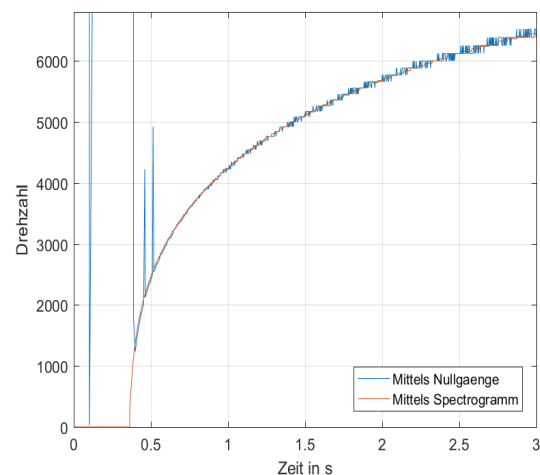


Abbildung 11: Für die Versorgungsspannung 50V

In den Abbildungen 10 und 11 sind die Vergleiche zwischen den Frequenzgängen, die mittels Nulldurchgänge im Zeitbereich und die mittels Spektrogramm ermittelt sind, dargestellt. Die blaue Kurve gehört zum Frequenzgang, der mit der Funktion `freqganganzeit` erstellt ist und die rote Kurve gehört zum Frequenzgang, der mit der Funktion `SpecToFreq` erstellt ist. Auf der blauen Kurve sind einige unbekannte Sprünge am Anfang zu sehen (siehe Abbildung 14). Der Grund dafür könnte an der Funktionsweise der Matlab-Funktion `freqganganzeit` liegen. Die Lage der Nulldurchgänge sind nicht immer ideal gefunden, weil die Messwerte oft nicht ganz auf dem Nullpunkt liegen und sogar manchmal weit entfernt davon. In den Abbildungen 12 und 13 sind die gefundenen Nulldurchgänge geplottet und es ist klar, dass eine ideale Berechnung der Nulldurchgänge bzw. eine ideale Berechnung der momentanen Drehfrequenz unmöglich ist. Deshalb ist der Frequenzgang, der mit `SpecToFreq` erstellt ist, mehr vertrauenswürdig. Trotzdem ein Blick auf die Vergleiche zeigt, dass die beiden Frequenzgänge eigentlich sehr ähnlich sind, sogar die Verläufe fast identisch sind. Also, doch funktioniert die Funktion `freqganganzeit` gut genug für einen zutreffenden Frequenzgang. Beide Kurven fangen bei 0 Hz an und steigen erst nach ca. 0,35 Sekunden. Wenn die Endwerte der Kurven mit den Werten in der Tabelle 1 verglichen sind, ist es festzustellen, dass die maximalen erreichten Drehzahlen gleich sind und die Drehzahl, die mittels Amplitudenspektrum berechnet ist, den richtigen Endwert herausgibt. An den Graphen der Endwert des zeitlichen Frequenzverlaufs bei 30V Versorgungsspannung liegt ungefähr bei 4200 RPM und bei 50V ungefähr bei 6500 RPM.

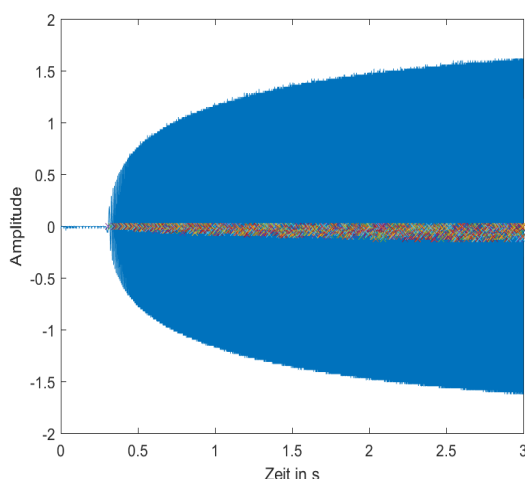


Abbildung 12: Für die Versorgungsspannung 30V

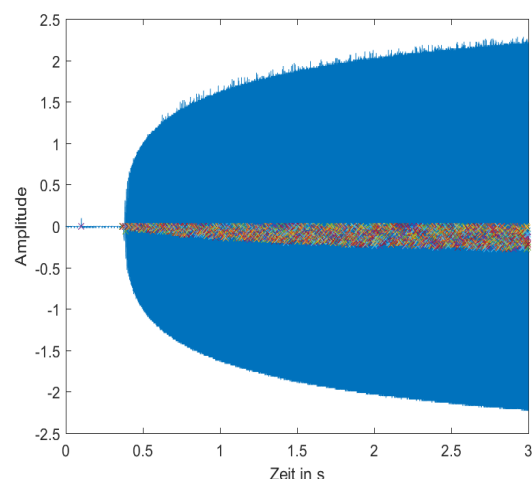


Abbildung 13: Für die Versorgungsspannung 50V

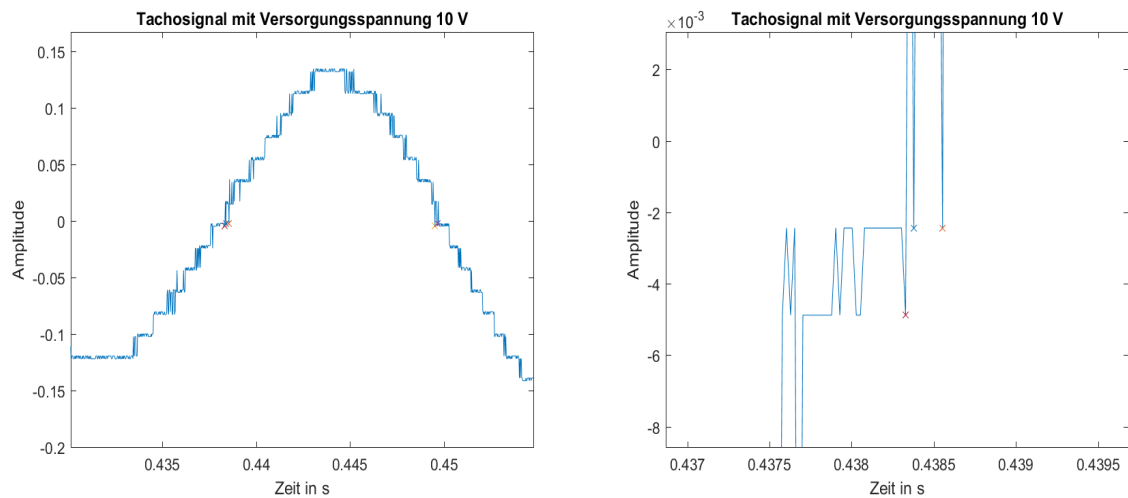


Abbildung 14: Fehler bei der Bestimmung der Nulldurchgänge

An der Abbildung 14 ist es zu sehen, dass mit der Funktionsweise der Matlab-Funktion `freqgangzeit` Fehlern bei einige Nulldurchgänge aufgetreten sind. Der Grund dafür ist das ganz kleine Messrauschen. Wegen diesem Messrauschen wird bei kleinere Drehfrequenzen die Nulldurchgänge mehrmals markiert. Dieser Fehler verursacht die vorhererwähnte Sprünge bei der Drehfrequenzberechnung, die in der Abbildung 10 und 11 gesehen sind. Trotzdem ist der Fehler nicht so wichtig, da die Wirkung sehr klein und vernachlässigbar ist.

3 Abgabearbeiten

3.1 Chirp-Signal

Mit der Matlab-Funktion `chirp` wird ein Sinus-Signal erstellt, dessen Frequenz bei 0 Hz anfängt und linear mit der Zeit ansteigt. Die erstellte Sinus-Signal ist in der Abbildung 15 dargestellt. Die Frequenz des Signals soll pro Sekunde 100 Hz steigen und das Signal dauert 2 Sekunden. Also die Frequenz erreicht am Ende maximal 200 Hz. Der zeitliche Verlauf der Signalfrequenz wird aus dem Spektrogramm abgelesen. Für die Erstellung der besten zeitlichen Verlauf der Frequenz wird der Befehl `spektrogramm` mit folgende Parameter verwendet und ist in der Abbildung 16 dargestellt. Wie erwähnt, nimmt die Frequenz 100 Hz pro Sekunde zu und erreicht 200 Hz am Ende.

- Fensterung mit `hamming(128)`: ermöglicht eine bessere Zeitlokalization und eine ausreichende Frequenzauflösung
- Anzahl der Überlappung 120: die Anzahl der Überlappung ist so gewählt, dass eine gute Anpassung der Frequenzauflösung gibt. Hier spielt die Größe der Fenster eine Rolle, die Anzahl der Überlappung ist fast so groß, wie die Größe des Fensters gewählt.
- Anzahl der DFT-Punkten 256: 128 wurde auch ausreichend sein aber mit 256 DFT-Punkten ergibt eine bessere Frequenzauflösung ohne Schwierigkeiten
- Abtastrate des Signals 1000: Das Sinus-Signal wurde mit der Funktion `chirp` mit einer Abtastrate von 1000 erstellt. Folglich wird die Zeitachse zutreffend abgebildet.

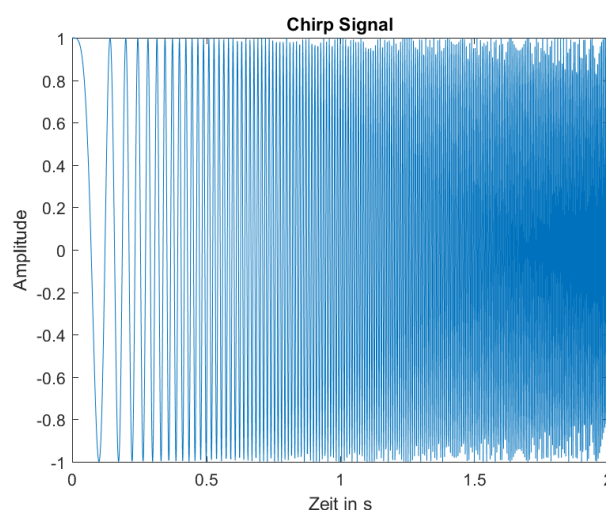


Abbildung 15: mit Chirp-Funktion erstellte Sinus-Signal

Die gelbe Linie in dem Spektrogramme zeigen eigentlich den zeitlichen Frequenzverlauf aber der Frequenzverlauf wird neben jede Spektrogramme auch mit der Matlab-Funktion SpecToFreq geplottet für eine bessere bildliche Verständniss.

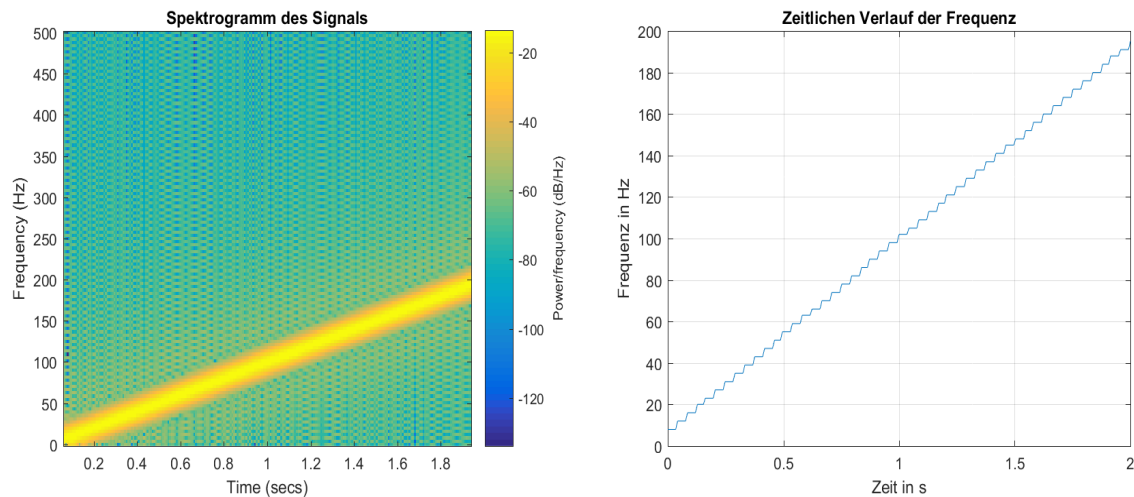


Abbildung 16: Spektrogramm mit hamming(128) und 120 Überlappungen

In der 17. Abbildung ist das Spektrogramm, für eine Fensterung von hamming(128) und 60 Überlappungen, zusammen mit dem zeitlichen Frequenzverlauf dargestellt. Wenn die Anzahl der Überlappungen viel weniger als die Fenstergröße ist, wird wie gesehen die Frequenzauflösung nicht anpassend.

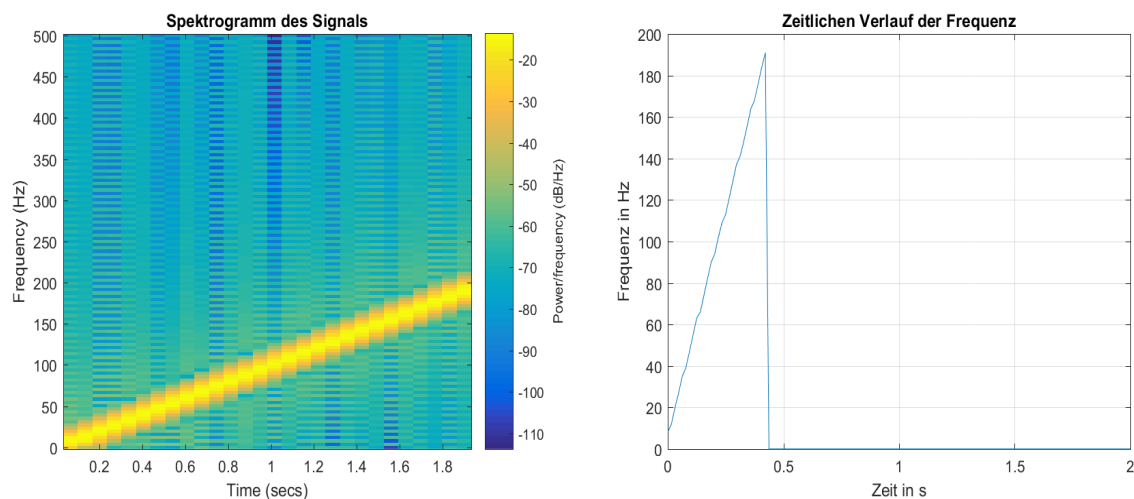


Abbildung 17: Spektrogramm mit hamming(128) und 60 Überlappungen

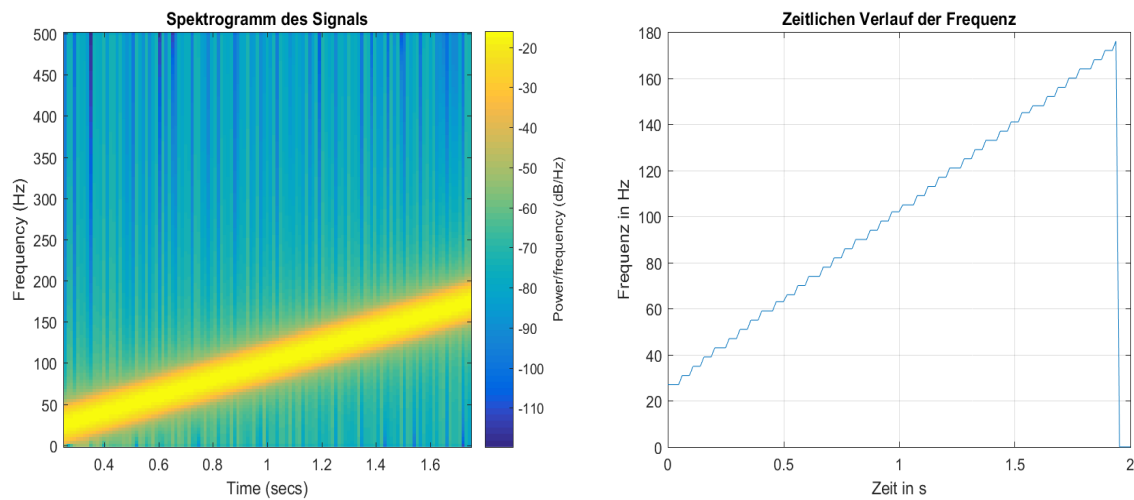


Abbildung 18: Spektrogramm mit hamming(512) und 500 Überlappungen

In der 18. Abbildung ist das Spektrogramm, für eine Fensterung von hamming(512) und 500 Überlappungen, zusammen mit dem zeitlichen Frequenzverlauf geplottet. Da die Anzahl der Überlappungen mit der Fenstergröße passt, sieht die Frequenzverlauf viel besser aus aber jetzt stimmt die Zeitlokalization nicht mehr wegen der große Fensterlänge.

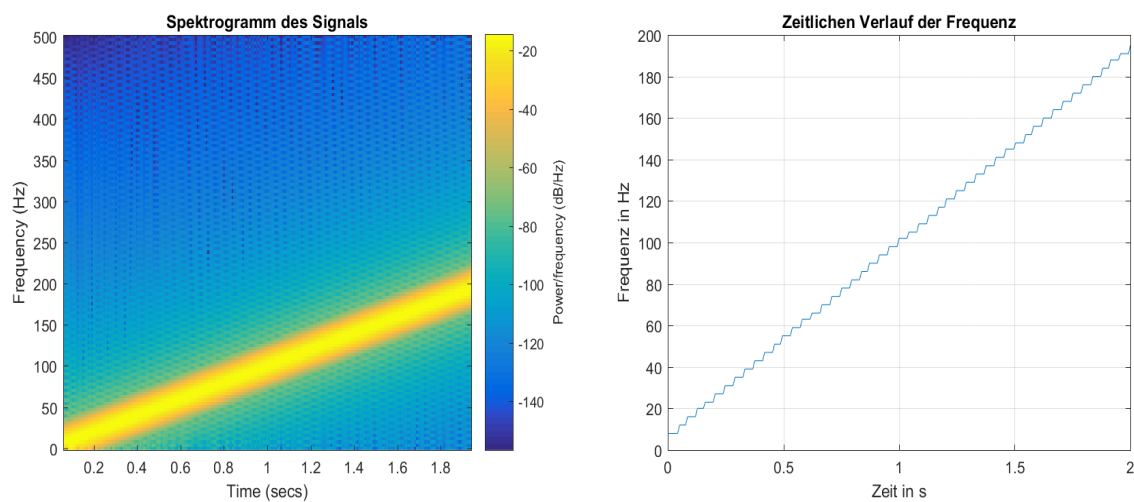


Abbildung 19: Spektrogramm mit blackman(128) und 120 Überlappungen

In der Abbildung 19 ist das Spektrogramm, für eine Fensterung von blackman(128) und 120 Überlappungen, zusammen mit dem zeitlichen Frequenzverlauf geplottet. Hier ist nur die

Art des Fensters anders als die ideale Einstellungen gewählt und zwar ein Blackman-Fenster. Bei dem Frequenzverlauf sieht die Frequenzauflösung und die Zeitlokalisierung sehr gut und passend aus. Bei der Benutzung des Blackman-Fensters wird anscheinend die Frequenzband im Spektrogramm breiter, die schnelle zeitliche Frequenzänderungen werden sichtbar.

3.2 Aufgabe 2

Diese Aufgabe wurde in dem Praktikumsteil 2.2 'Drehzahl des Motors' bearbeitet.

3.3 Aufgabe 3

Diese Aufgabe wurde in dem Praktikumsteil 2.3 'Analyse des zeitveränderlichen Stromsignals' und 2.4 'Vergleich der Drehzahlen' bearbeitet.