

# PROTOKOLL ZUR

# EINFÜHRUNG IN RECHNERGESTÜTZTES

# EXPERIMENTIEREN

Jannik Tim Zarnitz (E-Mail: [j\\_zarn02@wwu.de](mailto:j_zarn02@wwu.de))  
Leonhard Segger (E-Mail: [l\\_segg03@wwu.de](mailto:l_segg03@wwu.de))

in der Woche 03.09.2018 bis 06.09.2018  
betreut von  
Dr. Jürgen Berkemeier

20. Februar 2019

# Inhaltsverzeichnis

<b>1</b>	<b>Aufbau einer Sinus- bzw. Bessel-Funktion</b>	<b>3</b>
<b>2</b>	<b>Lissajous-Figuren</b>	<b>7</b>
<b>3</b>	<b>Digitales Oszilloskop mit ExpressVI</b>	<b>10</b>
<b>4</b>	<b>Abtasten von Signalen</b>	<b>12</b>
4.1	Fourier-Theorem . . . . .	12
4.2	Abtasttheorem . . . . .	12
4.3	Leakage-Effekt und Fensterfunktion . . . . .	13
4.4	Digitales Oszilloskop ohne ExpressVI . . . . .	13
4.5	Aliasing . . . . .	15
4.6	Unpassende Abtastrate . . . . .	16
<b>5</b>	<b>Amplitudenmodulierte Signale</b>	<b>19</b>
5.1	Erzeugung eines amplitudenmodulierten Signals . . . . .	19
5.2	Demodulation eines AM-Signals durch Quadrieren und Betragsbildung .	21
5.3	Demodulation eines AM-Signals mittels Trägerfrequenzmultiplikation . .	26
<b>6</b>	<b>Phasen- und frequenzmodulierte Signale</b>	<b>33</b>
6.1	Erzeugung eines phasen- bzw. frequenzmodulierten Signals . . . . .	33
6.2	Demodulation eines phasen- bzw. frequenzmodulierten Signals . . . . .	38
6.3	Erweiterte Demodulation mit Bandpass und zusätzlicher Integration des Signals . . . . .	43

# 1 Aufbau einer Sinus- bzw. Bessel-Funktion

Zunächst soll unter Verwendung der graphischen Programmiersprache „LabVIEW“ eine Sinus-Funktion realisiert werden. Mithilfe der von LabVIEW zur Verfügung gestellten numerischen Operationen, Konstanten und Schleifen (sogenannten „Visual Instruments“ oder einfach VIs) lässt sich ein Programm schreiben, welches eine Sinusschwingung  $u$  der Form

$$u(A, f, t, \varphi) = A \cdot \sin(2\pi f t + \varphi) , \quad (1)$$

umsetzt. Wobei  $A$  die Schwingungsamplitude,  $f$  die Frequenz und  $\varphi$  die Phasenverschiebung bilden. Außerdem ist jedes  $t$  durch

$$t = \frac{i}{N} \quad (2)$$

gegeben.  $N$  ist dabei die Zahl der Stützstellen und zugleich die Anzahl der zu berechnenden Wertepaare. Der Index  $i$  kann Werte im Bereich zwischen 0 und  $N$  annehmen. Der zugehörige Programmcode, das Blockdiagramm in LabVIEW, ist in Abb. 1 zu sehen.

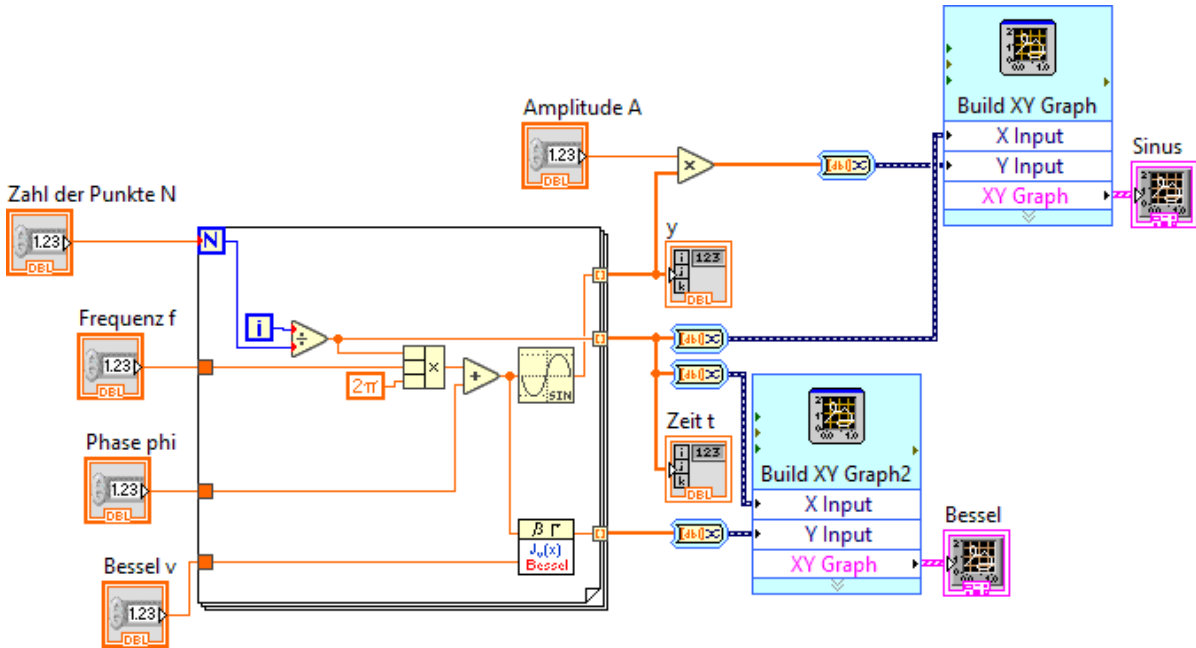


Abbildung 1: Die Abbildung zeigt ein LabVIEW-Blockdiagramm, also den Aufbau des Programms, zur Umsetzung einer Sinus- und Bessel-Funktion. Die eigentliche Berechnung findet innerhalb der For-Schleife statt. Dazu werden feste, aber beliebige Fließkommazahlen für die Amplitude  $A$ , die Zahl der Punkte  $N$ , die Frequenz  $f$ , die Phase  $\varphi$  und die Ordnung  $\nu$  der Bessel-Funktion herangezogen, nachdem sie auf dem Frontpanel eingestellt wurden. Die Ausgabe der Wertepaare erfolgt in Form zweier X-Y-Graphen sowie Arrays.

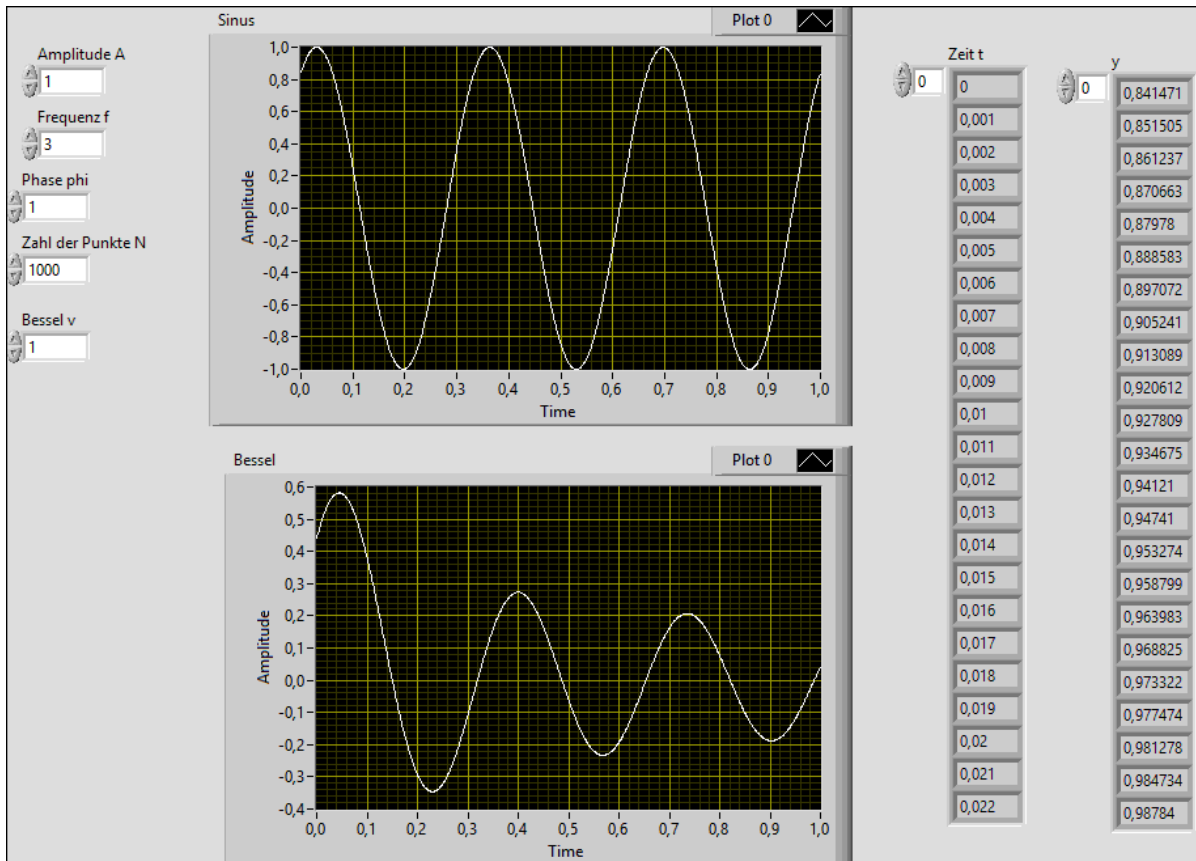


Abbildung 2: In dieser Abbildung ist das Frontpanel als Ausgabe und Benutzeroberfläche dargestellt.

Das dazugehörige ausgegebene Frontpanel ist in Abb. 2 erkennbar. Die Gleitkommazahlen-Werte für  $A$ ,  $f$ ,  $N$  und  $\varphi$  sollen frei wählbar sein, zumal  $u$  insbesondere von diesen abhängt. Das Programm ist daher so eingerichtet, dass auf der linken Seite des Frontpanels die entsprechenden Bedien- bzw. Eingabeelemente erscheinen. Unter gleichem Namen sind die jeweiligen Blockdiagrammobjekte im Programmcode zu finden. Im Zentrum des Programmcodes in Abb. 1 lässt sich eine For-Schleife erkennen. Gemäß Gleichung (1) sowie Gleichung (2) findet in dieser unter Verwendung des Laufindex  $i$  die tatsächliche Berechnung der  $t$ - und  $u$ -Werte statt. Die Blockdiagrammobjekte auf der rechten Seite des Programmcodes in Abb. 1 bilden die Anzeige- bzw. Ausgabeelemente des Frontpanels in Abb. 2. Zum einen werden dabei zwei  $N + 1$  Einträge umfassende, eindimensionale Arrays angelegt, die jeweils alle  $t$ - sowie sämtliche  $u$ -Werte beinhalten und zum anderen wird ein  $X$ - $Y$ -Diagramm erzeugt, in dem alle  $u$ -Werte gegen die entsprechenden  $t$ -Werte aufgetragen sind. Dabei befindet sich  $t$  auf der  $x$ -Achse und  $u$  auf der  $y$ -Achse. Um nun die Informationsübertragung zu bewerkstelligen, ist es notwendig im Programmcode vor den Anschlüssen der „Build XY Graph“-VIs weitere Blockdiagrammobjekte hinzuzufügen, welche das Leitungssignal in ein für das Anzeigeelement ( $X$ - $Y$ -Diagramm) passendes Eingangssignal umwandeln. Zumeist geschieht dies in LabVIEW automatisch und wird dann an der entsprechenden Stelle auf der Leitung mit

einem kleinen roten Punkt markiert. Zusätzlich zur Sinus-Funktion wird eine Bessel-Funktion erster Gattung  $J_\nu$  mit LabVIEW realisiert und ebenfalls in das Programm in Abb. 1 mit aufgenommen. Das entsprechende Blockdiagrammobjekt ist mit „ $J_\nu(x)$  Bessel“ gekennzeichnet. Die Bessel-Funktion besitzt die allgemeine Form

$$J_\nu(x) = \sum_{j=0}^{\infty} \frac{(-1)^j \cdot \left(\frac{x}{2}\right)^{2j+\nu}}{j! \cdot \Gamma(\nu + j + 1)} . \quad (3)$$

Wobei  $\nu$  die Ordnung der Bessel-Funktion bildet,  $\Gamma(\cdot)$  die Gamma-Funktion darstellt und  $x$  auf Basis des Programmcodes in Abb. 1 als

$$x := 2\pi f t + \varphi = \frac{2\pi f \cdot i}{N} + \varphi \quad (4)$$

definiert ist. Der Fließkommazahlen-Wert für  $\nu$  ist, ähnlich wie beim Sinus-Programm, über das entsprechende Bedienelement auf der linken Seite des Frontpanels einstellbar, was in Abb. 2 zu erkennen ist. Genauso wie beim Sinus-Programm findet die Berechnung der  $x$ - und  $J_\nu(x)$ -Werte innerhalb der For-Schleife statt. Die Werteausgabe erfolgt in Form eines  $X$ - $Y$ -Diagramms, dessen Anzeigeelement im unteren Bereich der Abb. 2 ersichtlich ist und in dem  $J_\nu(x)$  als Funktion von  $x$  dargestellt ist. Zudem muss das Eingangssignal so wie zuvor beim Sinus-Programm für das dazugehörige Blockdiagrammobjekt in Abb. 1 passend umgewandelt werden.

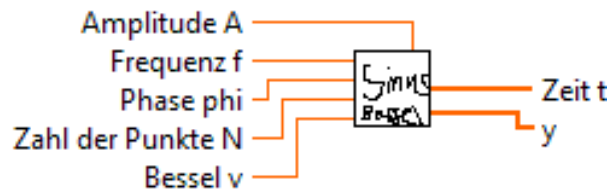


Abbildung 3: Die Abbildung zeigt das gestaltete Programm-Icon sowie die gesetzten Anschlussmöglichkeiten.

LabVIEW bietet die Möglichkeit ein bereits vorhandenes, selbst geschriebenes, funktionsfähiges Programm so zu bearbeiten, dass es für andere LabVIEW-Programme bzw. -Projekte als Blockdiagrammobjekt nutzbar wird. Dazu müssen zunächst die Anschlüsse sowohl die der Eingänge als auch die der Ausgänge „verkabelt“ werden. Allgemein kann es sich dabei um verschiedenste Arten von Ein- und Ausgangssignalen handeln. Bei der programmierten Sinus-Funktion verhält es sich demnach so, dass fünf Eingangs- und zwei Ausgangsanschlüsse generiert werden müssen, welche jeweils nur Gleitkommazahlen annehmen bzw. übergeben. Die Amplitude  $A$ , die Frequenz  $f$ , die Phasenverschiebung  $\varphi$ , die Anzahl der Punkte  $N$  und die Ordnung  $\nu$  der Bessel-Funktion bilden hierbei die Eingänge, die  $t$ -Werte sowie die  $u$ - bzw.  $y$ -Werte erscheinen über die beiden Ausgänge. Des Weiteren ist es in LabVIEW möglich das entsprechende Blockdiagrammobjekt-Icon des Programms zu gestalten. Das komplette Resultat ist in Abb. 3 dargestellt.

*Hinweis:* Aus der Betrachtung des Programmcodes in Abb. 1 geht hervor, dass das Blockdiagrammobjekt, welches das Array-Ausgabeelement des Frontpanels bildet und

alle  $u$ -Werte abgreifen soll, inkorrekt eingefügt ist. Denn anstelle einer Implementierung nach der Multiplikation mit der Amplitude  $A$ , befindet sich das besagte Blockdiagrammobjekt davor, sodass das Array lediglich die Werte des Ausdrucks

$$\sin(2\pi ft + \varphi) \tag{5}$$

angibt, welche zwischen  $-1$  und  $1$  liegen. Da die Ausgabefunktion als Array einen erheblichen Einfluss auf das folgende Vorgehen hat, kann mit dem Sinus-Programm in dieser Form nicht weitergearbeitet werden. Im weiteren Verlauf der Projektbearbeitung ist der beschriebene Fehler allerdings behoben worden. Eine berichtigte, aktualisierte Abbildung des Programms liegt an dieser Stelle jedoch nicht vor.

## 2 Lissajous-Figuren

Im Folgenden sollen je nach Werteinstellungen auf dem Anzeigeelement der Benutzeroberfläche verschiedene Lissajous-Figuren entstehen. Dies ist in Abb. 5 zu sehen. Die Basis für eine solche Lissajous-Figur bilden zwei verschiedene Sinusschwingungen, welche beide dieselbe Gestalt wie die Gleichung (1) besitzen:

$$u_1(A_1, f_1, t, \varphi_1) = A_1 \cdot \sin(2\pi f_1 t + \varphi_1) \quad \text{und} \quad u_2(A_2, f_2, t, \varphi_2) = A_2 \cdot \sin(2\pi f_2 t + \varphi_2), \quad (6)$$

mit den unterschiedlichen Amplituden  $A_1$  und  $A_2$ , den Frequenzen  $f_1$  und  $f_2$  sowie den Phasenverschiebungen  $\varphi_1$  und  $\varphi_2$ . Zudem bedarf es eines zweidimensionalen kartesischen Koordinatensystems mit Rechts- und Hochachse. Eine Lissajous-Figur entsteht nun dadurch, dass man der  $x$ -Achse die Funktion  $u_1$  und der  $y$ -Achse die Funktion  $u_2$  zuweist. Je nachdem wie man die Werte für  $A_1$ ,  $A_2$ ,  $f_1$ ,  $f_2$ ,  $\varphi_1$ ,  $\varphi_2$  und  $N$  wählt, ergeben sich dann unterschiedliche Lissajous-Figuren.

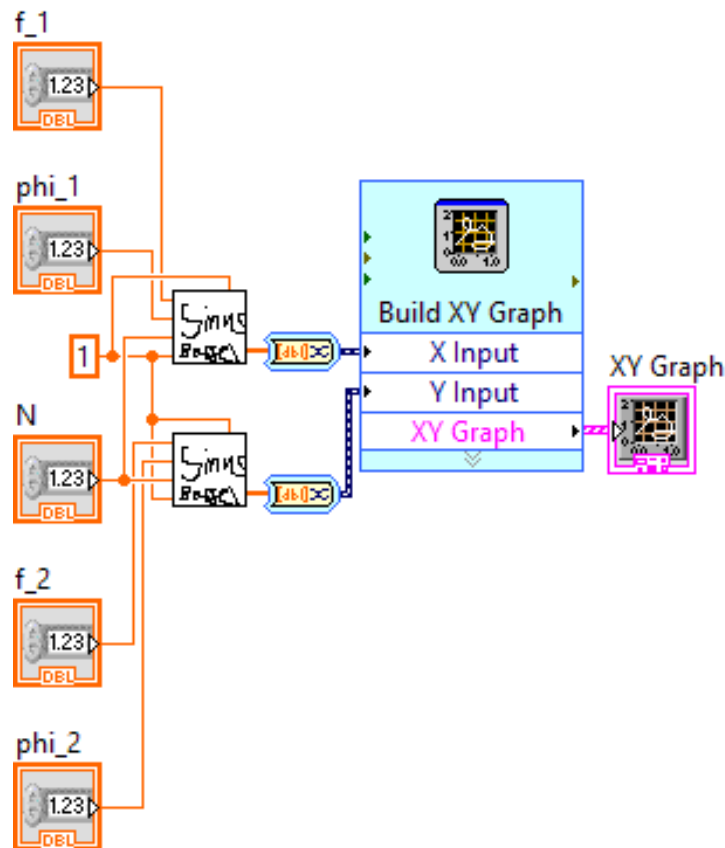


Abbildung 4: Dargestellt ist das Blockdiagramm, also der LabVIEW-Programmcode, mit dem sich Lissajous-Figuren realisieren lassen.

Der dazugehörige Programmcode ist in Abb. 4 zu erkennen. Die in Abschnitt 1 erstellte Sinus-Funktion bildet die Grundlage, denn mit dieser lassen sich die beiden separaten

Schwingungen  $u_1$  und  $u_2$  aus Gleichung (6) realisieren. Damit ist es möglich, das Sinus-Programm als Blockdiagrammobjekt in den Programmcode einzufügen: Jeweils einmal für  $u_1$  und  $u_2$ , was in Abb. 4 deutlich wird. An die offenen Eingangsanschlüsse müssen entsprechende, sogenannte „Controls“ für die Frequenzen  $f_1$  und  $f_2$ , für die Phasenverschiebungen  $\varphi_1$  und  $\varphi_2$  sowie für die Zahl der Stützstellen  $N$  angeschlossen werden. Die dazugehörigen Bedien- bzw. Eingabeelemente befinden sich links auf dem Frontpanel des Programms, so wie es in Abb. 5 dargestellt ist. Für die Amplituden soll einfachheits halber  $A_1 = A_2 = 1$  gelten.

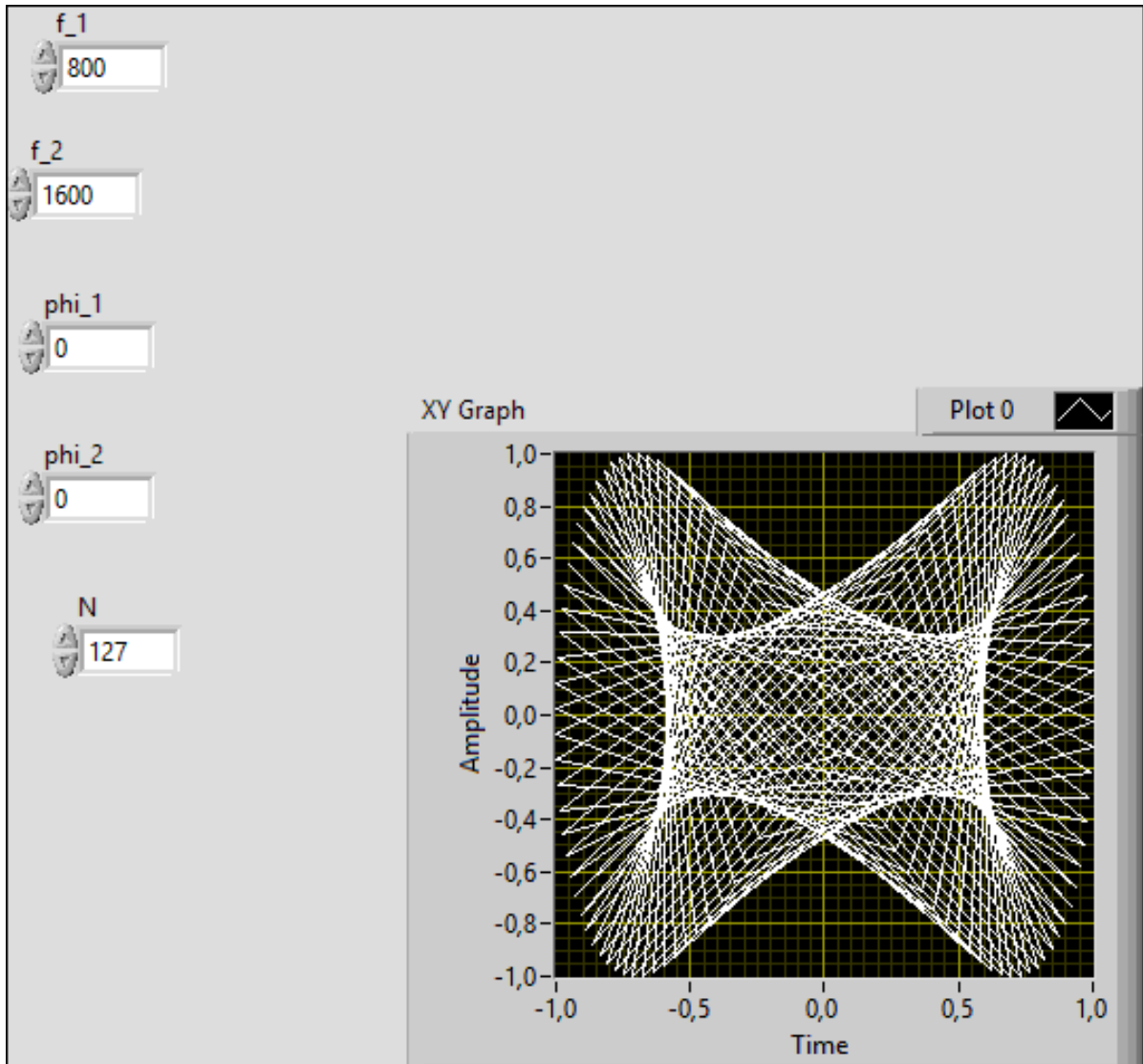


Abbildung 5: Die Abbildung zeigt das Frontpanel bzw. die Benutzeroberfläche, in dem sich im Anzeigeelement eine Lissajous-Figur befindet. Auf der linken Seite können die entsprechenden  $f_1$ -,  $f_2$ -,  $\varphi_1$ -,  $\varphi_2$ - und  $N$ -Werte abgelesen werden.



An den Ausgangsanschlüssen wird ein  $X$ - $Y$ -Diagramm angebracht, wobei die  $u_1$ -Werte auf der  $x$ -Achse und die  $u_2$ -Werte auf der  $y$ -Achse liegen. Wie beim Programm in Abschnitt 1, ist dem Ausgabeelement an der entsprechenden Stelle auf der Leitung ein passendes Blockdiagrammobjekt zur Signalumwandlung vorgeschaltet. Je nach Wahl der  $f_1$ -,  $f_2$ -,  $\varphi_1$ -,  $\varphi_2$ - und  $N$ -Werte können sich somit die unterschiedlichsten Lissajous-Figuren formen.

### 3 Digitales Oszilloskop mit ExpressVI

Es wird ein Funktionsgenerator verwendet. Dessen Signal wird über einen Analog-Digital-Wandler durch den Computer erfasst. Zunächst wird das Signal in LabView mit dem entsprechenden ExpressVI verarbeitet, welches dem Nutzer große Teile des Konfigurationsvorgangs abnimmt. Das zugehörige Programm ist in Abb. 6 und das Frontpanel in Abb. 7 dargestellt.

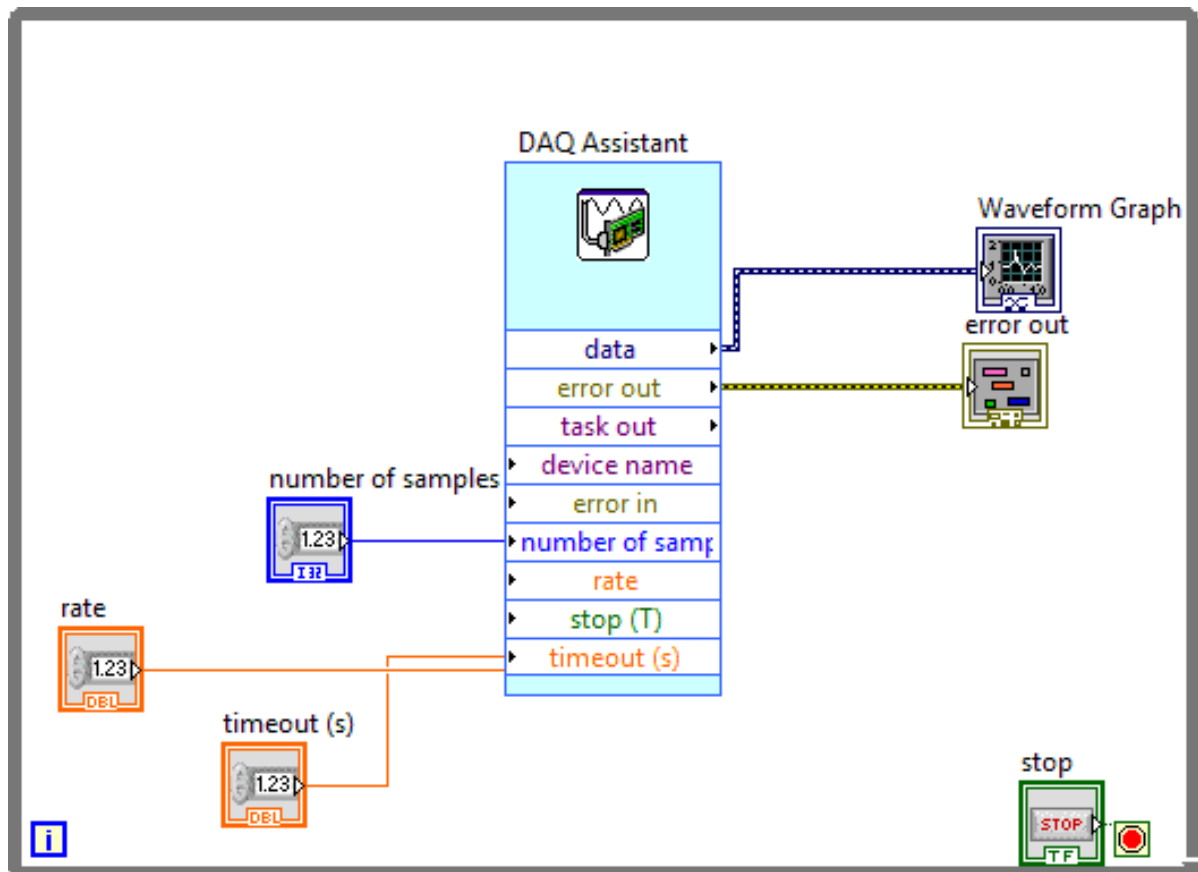


Abbildung 6: Einfaches Oszilloskop mithilfe des ExpressVIs zur Erfassung von Daten von Messgeräten. Das Programm erlaubt die graphische Darstellung einer Wellenform.

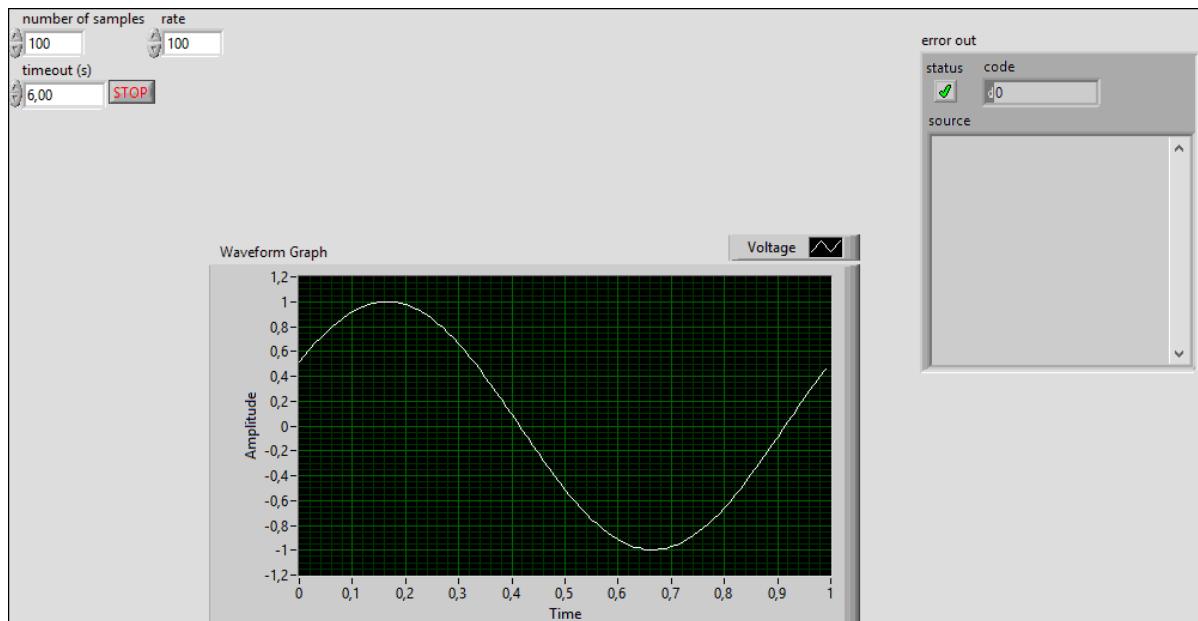


Abbildung 7: Frontpanel des einfachen Oszilloskops mithilfe des ExpressVIs zur Erfassung von Daten von Messgeräten. Dabei wurde durch den Funktionsgenerator ein sinusförmiges Signal ausgegeben.

## 4 Abtasten von Signalen

### 4.1 Fourier-Theorem

Gemäß des Fourier-Theorems kann jede periodische Funktion als Fourier-Reihe bzw. Fourier-Integral ausgedrückt werden. Dies ist nützlich bei der Zerlegung eines möglicherweise verrauschten Signals in seine Bestandteile. Das grundsätzliche Problem hierbei ist, dass Messprozesse immer zeitlich begrenzt sind, weshalb das Signal nicht bis in die positive und negative Unendlichkeit periodisch sein kann. Dies verursacht den sogenannten „Leakage-Effekt“, auf den in Abschnitt 4.3 näher eingegangen wird.

### 4.2 Abtasttheorem

Um ein Signal abzutasten, werden im Analog-Digital-Wandler mithilfe einer Sample-and-Hold-Schaltung zeitlich diskrete Messungen durchgeführt. Dies lässt sich als Multiplikation des Signals mit einem Delta-Kamm (mit Abständen der Delta-Zacken gleich den Messabständen) ausdrücken. Dabei treten Summen- und Differenzfrequenzen von Abtastfrequenz und deren Oberfrequenzen mit den Frequenzen im Signal auf. Im Frequenzraum ergibt sich hierdurch eine periodische Fortsetzung des Spektrums des ursprünglichen Signals, wobei die Periode der Abtastfrequenz entspricht.

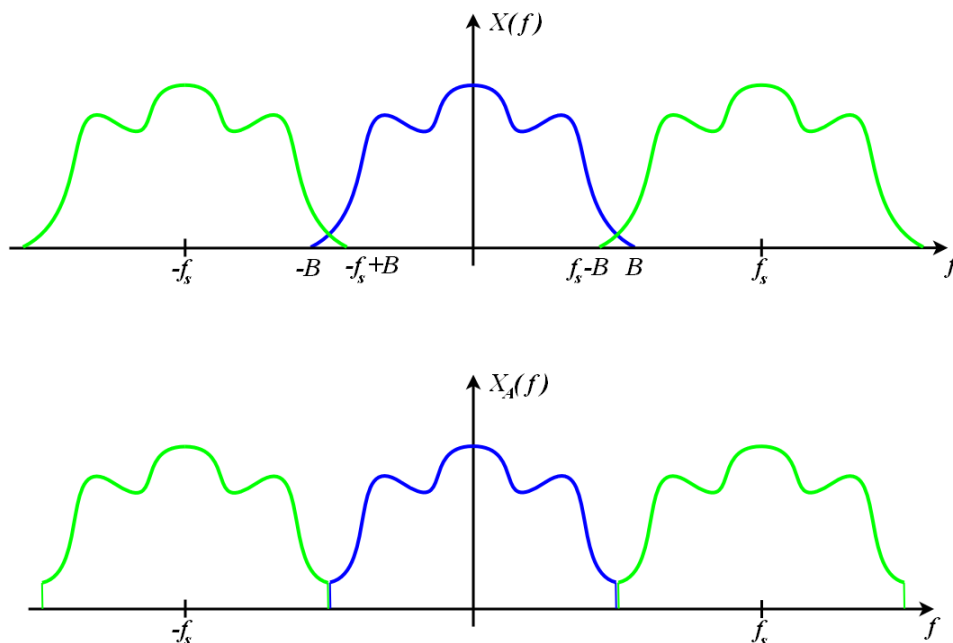


Abbildung 8: Veranschaulichung des Effekts des Aliasing. Dargestellt ist ein Beispielsignal, das mit der Abtastfrequenz  $f_s$  abgetastet wurde. Da diese geringer ist als das Doppelte der höchsten Frequenz im Signal  $B$ , überlappen sich die Perioden im Frequenzraum. Das ursprüngliche Signal kann durch einen Tiefpass nur unter Verlust der höheren Frequenzen rekonstruiert werden. [2]

Dies ist in Abb. 8 veranschaulicht. Wenn die Abtastfrequenz hinreichend groß ist, kann man nun mithilfe eines Tiefpasses das Signal herausfiltern. Dazu muss sie allerdings größer als das Doppelte der höchsten im Signal auftretenden Frequenz sein, da sich ansonsten das Spektrum des Signals mit den Differenzfrequenzen der nächsten Periode überlagern. Wenn letzteres geschieht, bezeichnet man dies als Aliasing.

### 4.3 Leakage-Effekt und Fensterfunktion

Wenn die Signalfrequenz kein Vielfaches des Produkts aus Abtastfrequenz und Zahl an Messpunkten pro Messung ist, tritt aufgrund der Endlichkeit des Messprozesses der sogenannte „Leakage-Effekt“ auf. Dieser verursacht eine Verbreiterung des Peaks der Signalfrequenz im Frequenzbild. Dies ist in Abb. 9 dargestellt.

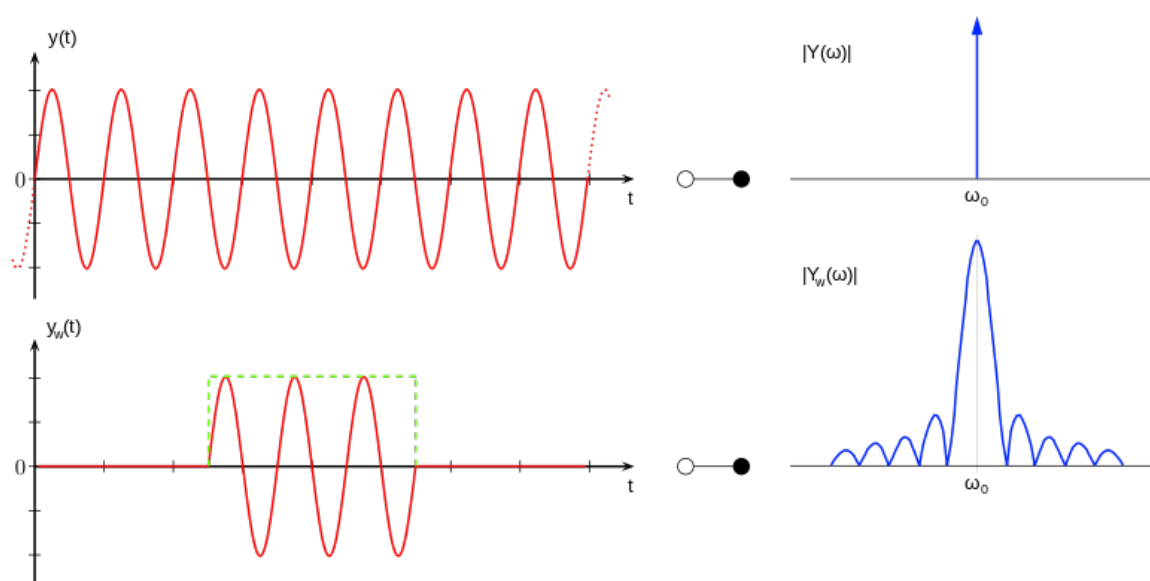


Abbildung 9: Zeitlich nicht limitierte („ungefensterte“) Sinusschwingung oben, zeitlich limitierte Sinusschwingung unten (limitiert mit Rechteck-Fensterfunktion) und daneben deren Fourier-Transformierten.[1]

Um diesen Effekt zu minimieren kann das Signal mit einer Fensterfunktion multipliziert werden. Im Folgenden wird das „Von-Hann-Fenster“ (auch „Hanning-Fenster“) verwendet, welches der Haversine-Funktion entspricht ( $\frac{1-\cos\vartheta}{2}$ ). Der Unterschied zwischen Spektralanalyse mit und ohne Von-Hann-Fenster ist in Abb. 11 zu erkennen, da dort ohne Fenster-Funktion zusätzliche Frequenzen in der Nähe des Peaks auftreten.

### 4.4 Digitales Oszilloskop ohne ExpressVI

Das Oszilloskopprogramm von zuvor wird ersetzt durch eines, das anstelle des Express-VIs Konfiguration, Messung und Cleanup getrennt enthält. Dieses ist in Abb. 10 darge-

stellt, während in Abb. 11 die Frontpanel bei einem Eingangssignal von 600 Hz abgebildet ist.

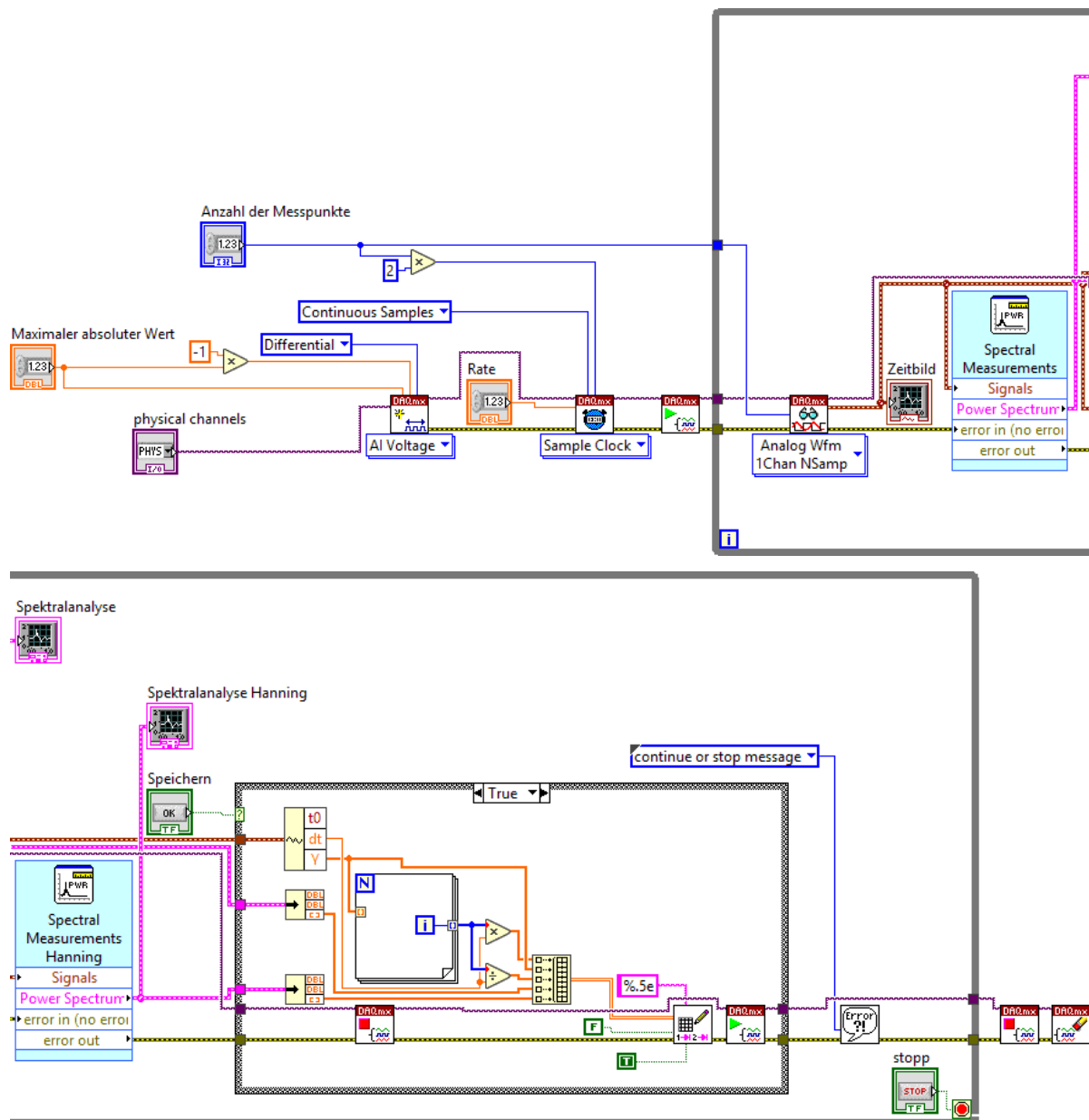


Abbildung 10: Oszilloskopprogramm, das eine Abtastung vornimmt und das Ergebnis im Zeitbild sowie im Frequenzbild mit und ohne Hann-Fenster darstellt. Außerdem ist die Speicherung der Daten in einem Textdokument ermöglicht.

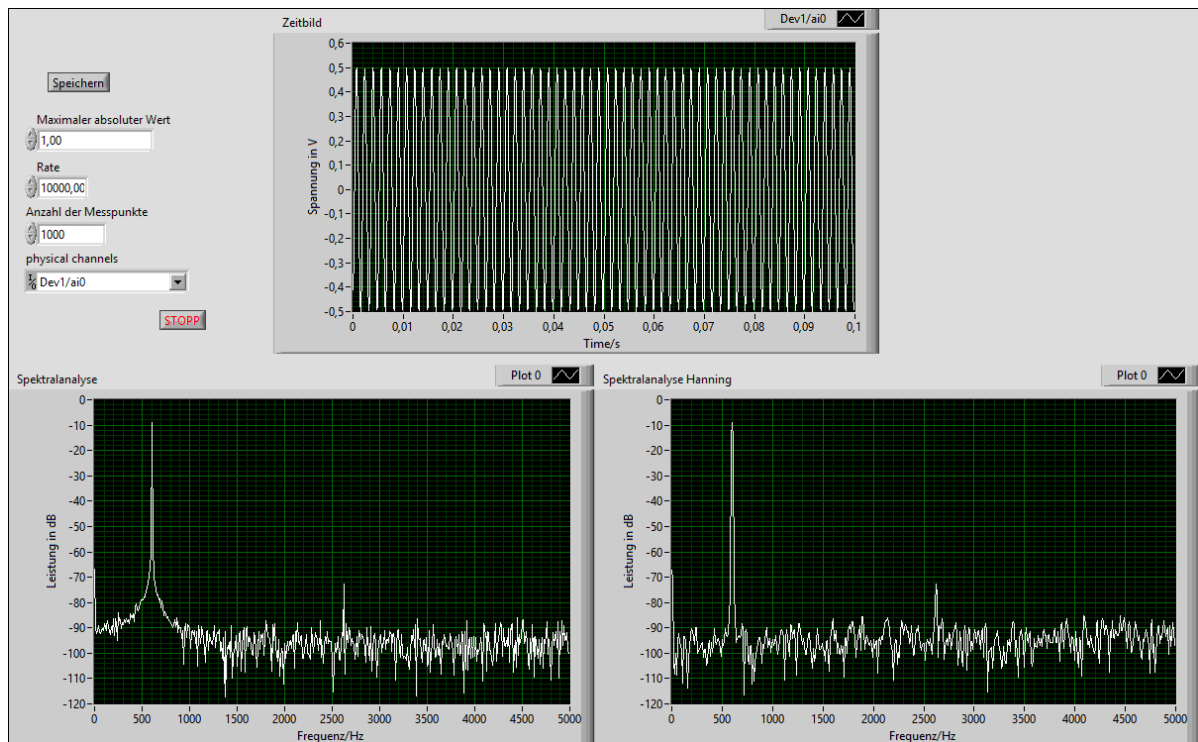


Abbildung 11: Oszilloskopfrontpanel. Dargestellt wird das gemessene Signal im Zeitbild sowie im Frequenzbild mit und ohne Von-Hann-Fenster. Einstellbar ist die Abtastfrequenz, die Anzahl der Messpunkte, der Eingangskanal des Messgerätes und der maximale messbare Wert, wobei der minimale auf das Negative des maximalen gesetzt wird. Mithilfe des Stopp-Knopfes kann das Programm gestoppt werden und mit dem Speichern-Knopf werden die aktuellen Messwerte aus allen drei Diagrammen in eine Textdatei exportiert.

Hierbei wird ebenfalls die Möglichkeit, die aktuellen Messwerte zu speichern, eingeführt. Dazu werden die Strukturen aus verschiedenen Datentypen, die in den Diagrammen dargestellt werden, in ihre Bestandteile zerlegt und die Arrays der Y-Werte mit zwei Arrays für fortlaufende Zeit- und Frequenzwerte kombiniert und der Speicherfunktion zugeführt. Der Speichervorgang befindet sich innerhalb eines case-Konstrukts, das ignoriert wird, solange der Speichern-Knopf nicht gedrückt wurde. Außerdem werden Fehlermeldungen einer Fehlerdialogfunktion zugeführt, die dem Nutzer erlaubt bei Fehlern wahlweise das Programm anzuhalten oder weiterlaufen zu lassen. Um ein Volllaufen des Mess-Buffers zu vermeiden, wird dieser doppelt so groß wie die Anzahl der Messpunkte pro Zyklus gewählt.

## 4.5 Aliasing

Um den Effekt des Aliasings absichtlich herbeizuführen, werden bei einer Abtastfrequenz von 1000 Hz zwei verschiedene Signale abgetastet. Da bei dieser Abtastfrequenz

die höchste Frequenz im Signal kleiner als 500 Hz sein muss, ist hierbei zu erwarten, dass ein Signal von 400 Hz korrekt abgetastet wird, während eines mit 600 Hz falsch abgetastet wird. Das Spektrum des abgetasteten Signals bei diesen beiden Signalfrequenzen ist in Abb. 12 dargestellt.

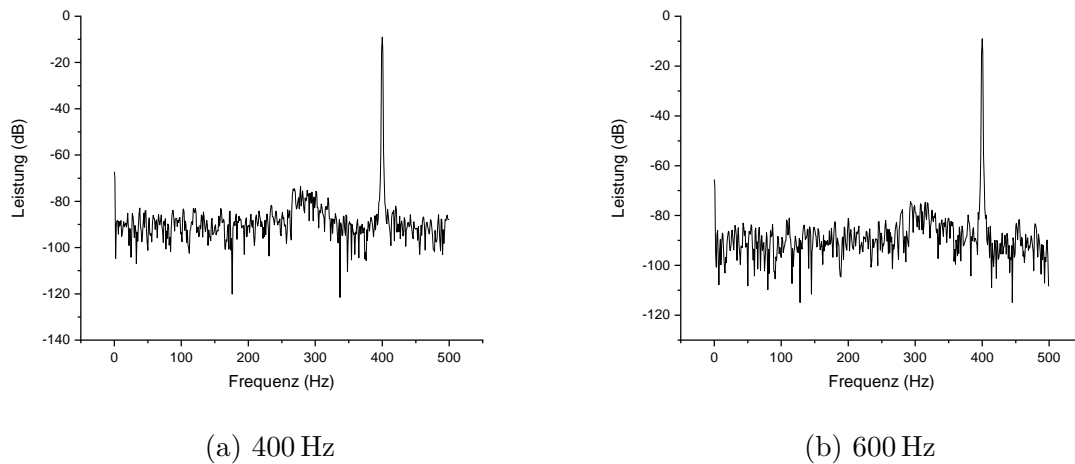


Abbildung 12: Mit einer Abtastfrequenz von 1000 Hz abgetastete Signale, deren Frequenz einmal 400 Hz und einmal 600 Hz beträgt, sodass einmal korrekt und einmal falsch abgetastet wird. Um die Sichtbarkeit zu verbessern und die übliche Oszilloskop-Optik zu erreichen, wurden die Messpunkte hier und in den folgenden Darstellungen mit einer Linie verbunden.

Es fällt auf, dass sich zwischen diesen beiden Signalen kein Unterschied erkennen lässt. Dies liegt daran, dass bei einer Signalfrequenz von 600 Hz die eigentliche Signalfrequenz nicht mehr vom idealen Tiefpass übertragen wird, während die Differenzfrequenz von Abtastfrequenz und Signalfrequenz 400 Hz beträgt und noch übertragen wird. Es ist an den beiden Signalen zu erkennen, dass sich ein falsch abgetastetes Signal nicht mehr von einem korrekt abgetasteten Signal bei der entsprechenden Differenzfrequenz unterscheiden lässt. Man stellt fest, dass man vor der Abtastung bereits wissen muss, welche Frequenzen im Signal vorkommen.

## 4.6 Unpassende Abtastrate

Um den Leakage-Effekt, der in Abschnitt 4.3 beschrieben wurde, zu bebildern, wird er in Abb. 13 bis 15 absichtlich herbeigeführt. Dabei wird hier das Produkt aus Abtastfrequenz und Anzahl der Messpunkte pro Messdurchgang als „Abtastrate“ bezeichnet. Hier beträgt die Abtastfrequenz 1000 Hz und die Anzahl der Messpunkte pro Messdurchgang 100, woraus sich eine Abtastrate von 10 Hz ergibt. Wenn die Signalfrequenz kein Vielfaches der Abtastrate ist, führt dies dazu, dass eine eigentlich scharfe Frequenz im Spektrum unscharf wird und Frequenzen geringerer Intensität in der Nähe dieser überdeckt werden. Man stellt fest, dass man, wenn man die Signalfrequenz kennt, die



Abtastrate richtig wählen muss und sich, wenn dem nicht so ist, die Verwendung einer Fensterfunktion empfiehlt.

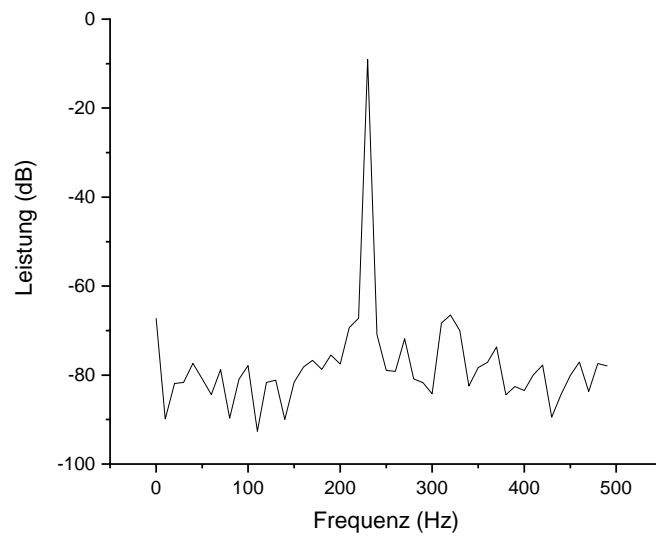


Abbildung 13: Spektrum eines Signals mit einer Frequenz von 230 Hz, welche ein Vielfaches der Abtastrate ist, weshalb hier die Signalfrequenz scharf zu erkennen ist. Eine Fensterfunktion wurde hier nicht verwendet.

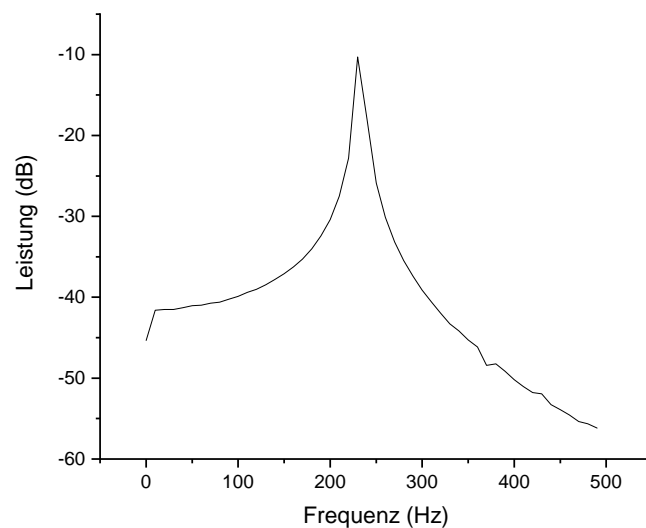


Abbildung 14: Spektrum eines Signals mit einer Frequenz von 233 Hz, welche kein Vielfaches der Abtastrate ist, weshalb hier zu erkennen ist, dass die Signalfrequenz unscharf wird. Eine Fensterfunktion wurde hier nicht verwendet.

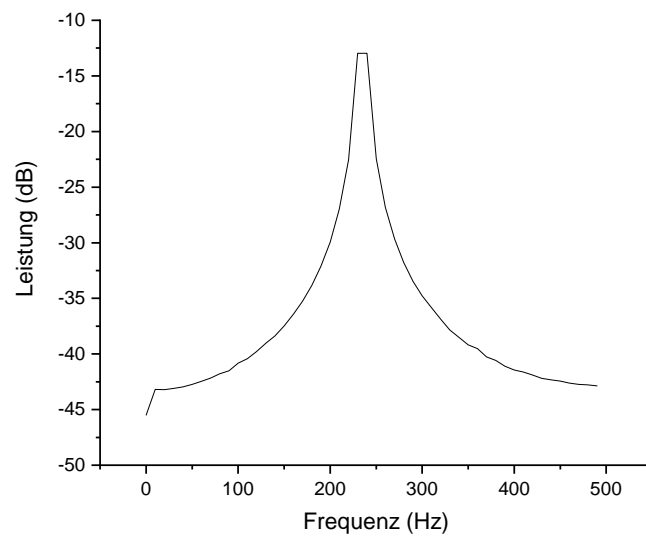


Abbildung 15: Spektrum eines Signals mit einer Frequenz von 235 Hz, welche kein Vielfaches der Abtastrate ist, weshalb hier zu erkennen ist, dass die Signalfrequenz unscharf wird. Eine Fensterfunktion wurde hier nicht verwendet.

## 5 Amplitudenmodulierte Signale

### 5.1 Erzeugung eines amplitudenmodulierten Signals

Es soll nun ein amplitudenmoduliertes Signal (auch als AM-Signal bezeichnet) erzeugt werden und über die Soundkarte des verwendeten PCs ausgegeben werden. Dann wird das Signal der Soundkarte wieder auf den Analog-Digital-Wandler gegeben und mit dem zuvor erstellten Oszilloskopprogramm verarbeitet. In Abb. 16 ist das dazu verwendete LabView-Programm dargestellt. Moduliert werden soll die Überlagerung zweier Signale mit einstellbarer Frequenz und Amplitude. Auch Trägerfrequenz und Modulationsgrad soll einstellbar sein. Die Amplitudenmodulation eines Signals  $s(t)$  lässt sich durch die Gleichung

$$S_{AM}(t) = S_T(1 + m \cdot s(t)) \cdot \cos(2\pi f_T t) \quad (7)$$

beschreiben. Dabei ist  $m$  der Modulationsgrad,  $f_T$  die Trägerfrequenz und  $S_T$  die Amplitude. Da die Überlagerung von zwei sinusförmigen Signalen beliebiger aber fester Frequenz moduliert werden soll, gilt:

$$s(t) = A_1 \cos(2\pi f_1 t) + A_2 \cos(2\pi f_2 t) . \quad (8)$$

Dabei sind  $A_1$  und  $A_2$  die Amplituden und  $f_1$  und  $f_2$  die Frequenzen der Signale.

Diese beiden Gleichungen sollen im Folgenden durch Operationen auf Arrays realisiert und als Audiosignal ausgegeben werden. Dazu wird das in Abschnitt 1 erstellte Programm verwendet, um Arrays von Sinusfunktionen zu erstellen. Diesem wird in jedem Fall die Phase 0 zugeführt und es werden 44100 Stützstellen gewählt, weil dies der Abtastrate bei CDs entspricht. Dem Programm wird zunächst eine Amplitude von 1 gegeben und dann werden alle Array-Elemente mit der gewählten Amplitude multipliziert. Dies hängt lediglich damit zusammen, dass zuvor im Sinus-erstellenden Programm ein Fehler vorhanden war, der dies nötig machte (vgl. Abschnitt 1). In Abb. 16 ist zu erkennen, dass zunächst zwei Sinus-Arrays erstellt werden, für die jeweils eine Signalfrequenz und -amplitude gewählt werden. Die beiden resultierenden Arrays werden addiert, mit dem Modulationsgrad multipliziert und dann mit dem Array der Trägerfrequenz multipliziert. Dieses wurde zuvor in gleicher Art und Weise mit dem Sinus-Programm bei der einstellbaren Trägerfrequenz erstellt. Für die Amplitude des Trägersignals wird der Kehrwert der maximalen Signalamplitude gewählt, damit die resultierende Amplitude immer auf 1 liegt, da die Soundkarte höhere Werte nicht verarbeitet. Das resultierende Array wird im Zeit- und Frequenzbild dargestellt und dann mithilfe der entsprechenden VIs über die Soundkarte ausgegeben. Hierfür wird die Abtastrate auf 44 100 Hz, die Anzahl der Kanäle auf 1 eingestellt und der Modus „Continuos Samples“ gewählt, da das Signal fortlaufend erzeugt werden soll. Außerdem wird das Element zur Ausgabe von Fehlern verwendet, das es erlaubt das Programm beim Auftreten von Fehlern wahlweise zu beenden oder weiterlaufen zu lassen.

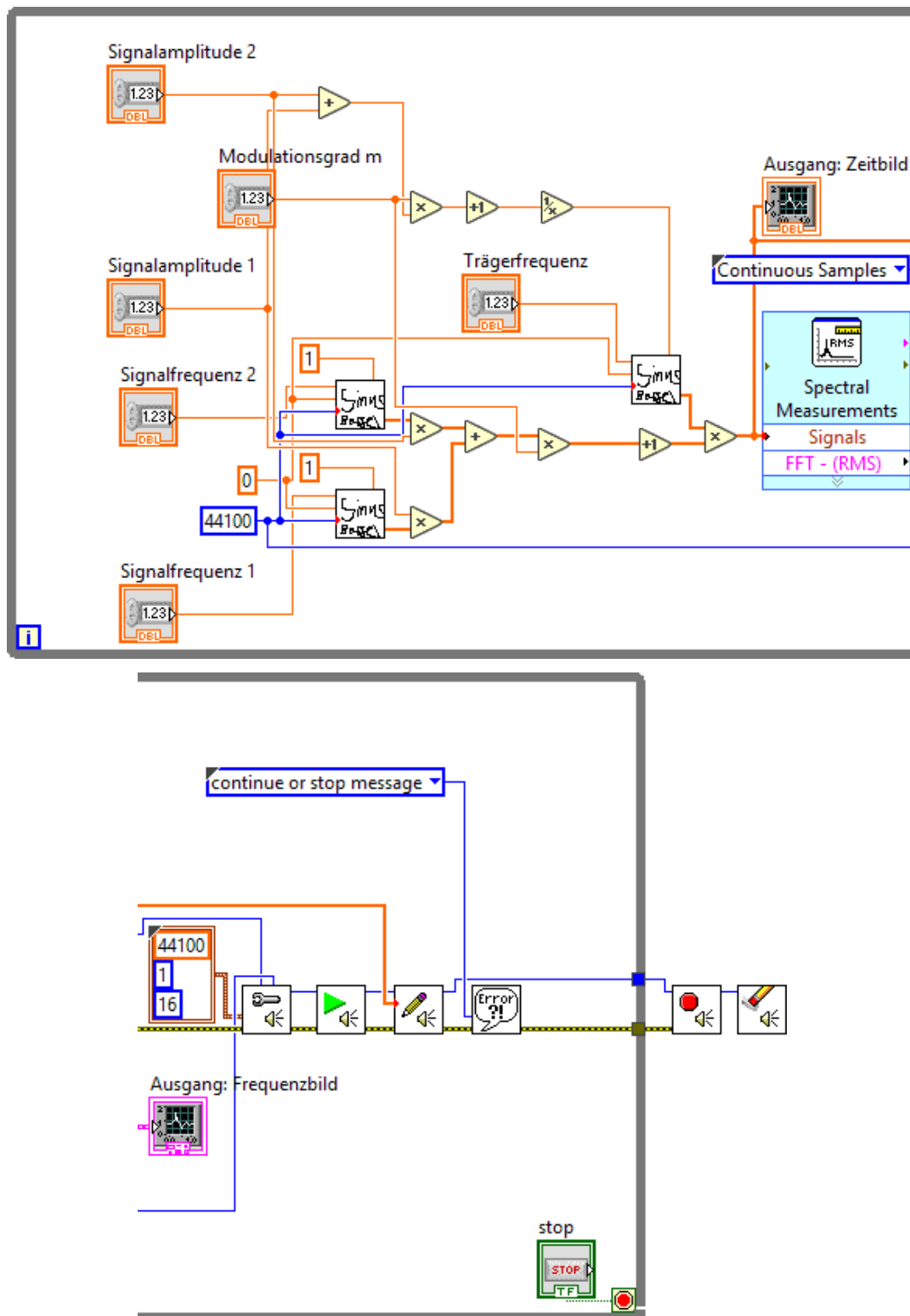


Abbildung 16: Blockdiagramm des Programms zur Erzeugung eines amplitudenmodulierten Signals und Ausgabe desselben über die Soundkarte.

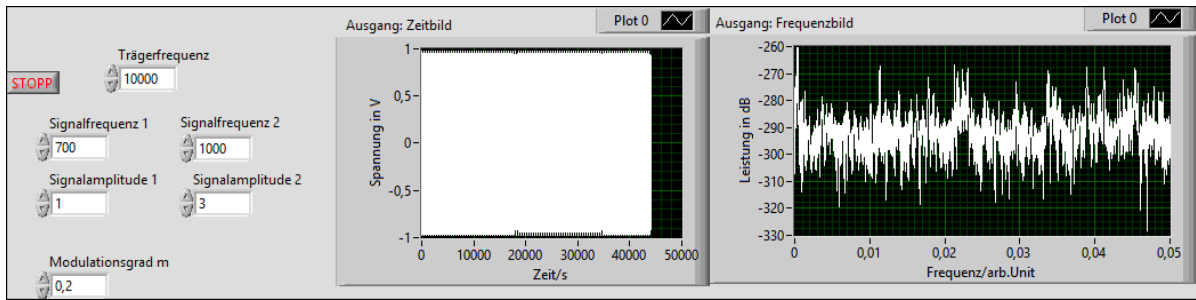


Abbildung 17: Frontpanel des Programms zur Erzeugung eines amplitudenmodulierten Signals und Ausgabe dessen über die Soundkarte.

Auf das Ergebnis der Modulation wird im folgenden Abschnitt eingegangen.

## 5.2 Demodulation eines AM-Signals durch Quadrieren und Betragsbildung

Um nun das amplitudenmodulierte Signal im Oszilloskopprogramm wieder demodulieren zu können, wird ein Case-Konstrukt eingeführt, welches nach Auswahl des Benutzers entweder das Signal nicht demoduliert, mit sich selbst multipliziert oder den Betrag des Signals bildet, da in den letzten beiden Fällen unter anderem wieder die Ursprungsfrequenzen vorliegen. Nach diesem Schritt wird das Signal im Zeit- und Frequenzbild dargestellt und danach zusätzlich durch einen Tiefpass geführt, der auftretende höhere Frequenzen herausfiltert, und dann erneut im Zeit- und Frequenzbild dargestellt. Außerdem wird die Speicherung von zuvor erweitert, sodass das Signal vor der Demodulation, vor dem Tiefpass und nach diesem gespeichert werden kann. Das fertige Programm ist in Abb. 18 und das zugehörige Frontpanel in Abb. 19 dargestellt. Dass man in der Zeitdomäne auf letzterem wenig erkennen kann, liegt an der Vielzahl und der Größe der auftretenden Frequenzen. Diese Tatsache gilt für alle kommenden Untersuchungen, weshalb im Folgenden nur noch die Frequenzdomäne dargestellt wird.

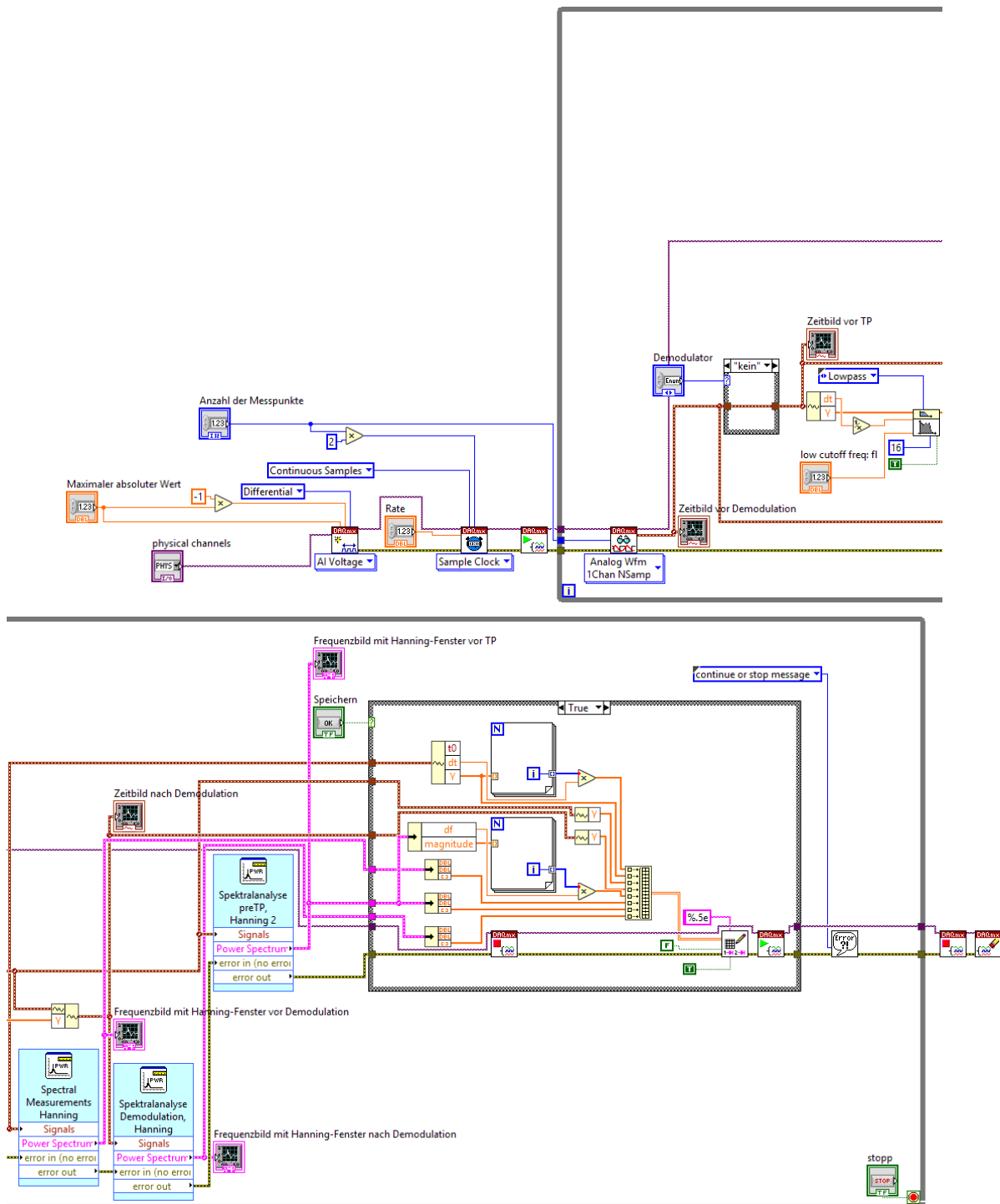


Abbildung 18: Blockdiagramm des Programms zur Demodulation und Darstellung eines amplitudenmodulierten Signals. Zur Demodulation kann Quadrieren und Bildung des Betrags gewählt werden. In diesen Fällen steht im Case-Konstrukt ein Quadrierer bzw. ein Multiplizierer.

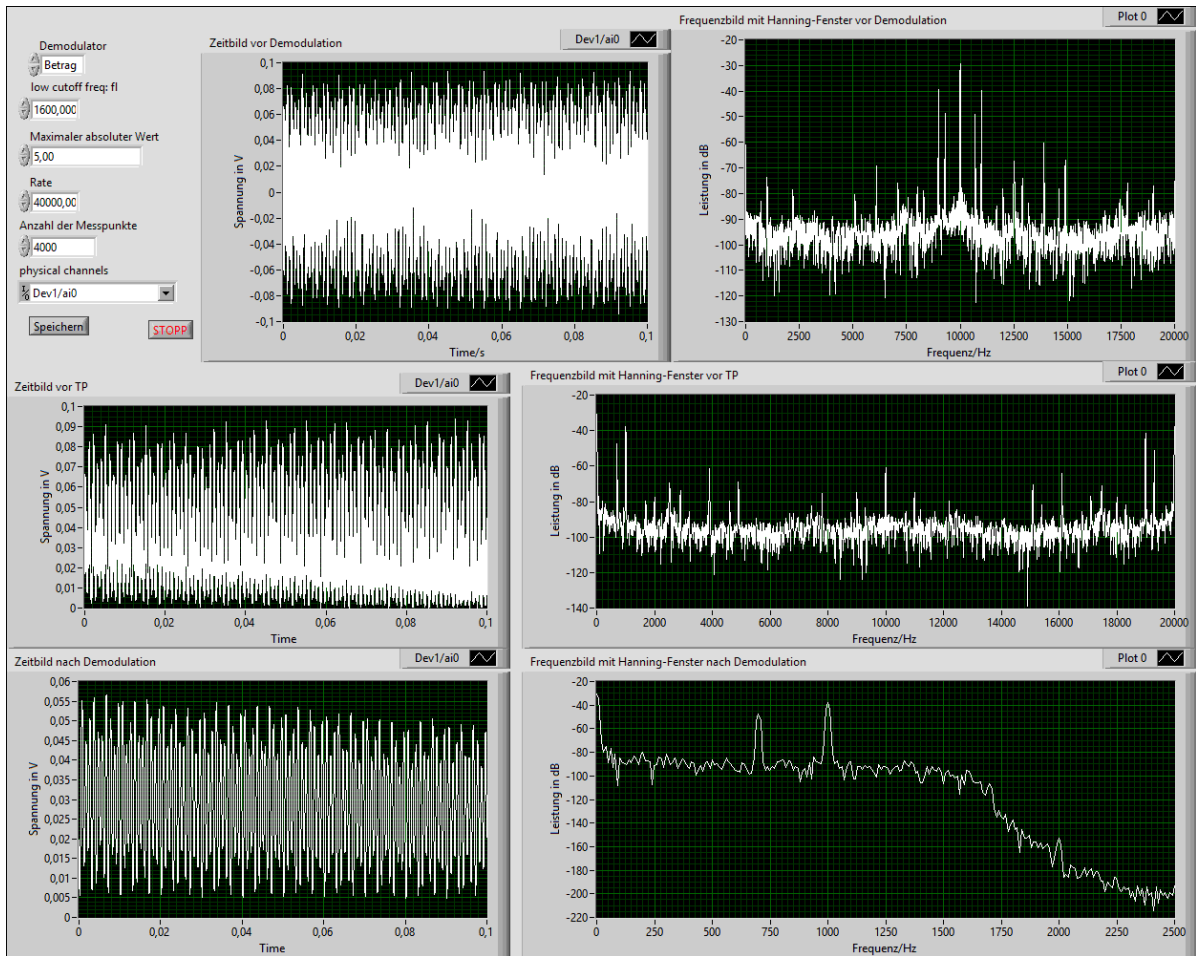


Abbildung 19: Frontpanel des Programms zur Demodulation und Darstellung eines amplitudenmodulierten Signals. Einstellbar ist die Art der Demodulation (keine, Quadrieren, Betragsbildung), die Grenzfrequenz des Tiefpasses nach der Demodulation, maximal messbarer Wert, Messfrequenz, Anzahl der Messpunkte pro Messdurchgang und Eingangskanal, an dem der Analog-Digital-Wandler angeschlossen ist. Die Grafiken zeigen jeweils im Zeit- und Frequenzbild (mit Hanning-Fensterfunktion) das Signal vor, nach und während der Demodulation. Die aktuellen Punkte in allen sechs Grafiken können in einer Textdatei gespeichert werden.

In Abb. 20 bis 24 sind die Graphen, die sich aus der durch den „Speichern“-Knopf erstellten Textdatei ergeben, dargestellt. All diesen ist gemein, dass neben den rechnerisch durch die Modulation entstehenden Frequenzen noch einige andere von signifikanter Stärke vorhanden sind. Diese müssen auf Störeffekte durch Netzfrequenz, deren Oberfrequenzen und andere Effekte bei der Erzeugung durch die Soundkarte zurückgeführt werden. Im Folgenden wird immer ein amplitudenmoduliertes Signal mit einer Trägerfrequenz von 10 000 Hz verwendet. Vor der Modulation beinhaltet das Signal die Frequenz 700 Hz mit einer relativen Amplitude von 1 und die Frequenz 1000 Hz mit einer relativen

Amplitude von 3. Als Modulationsgrad wird 0,2 gewählt.

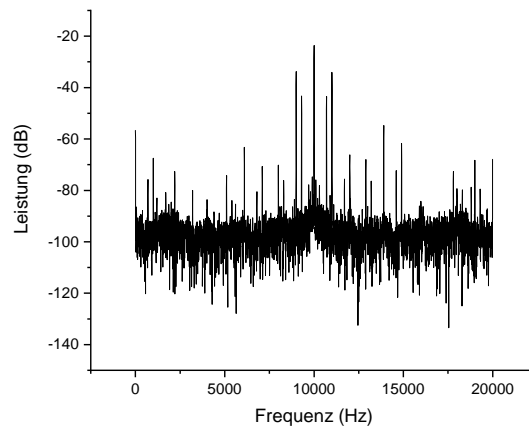


Abbildung 20: Ausgabe des Oszilloskopprogramms für das Eingangssignal in der Frequenzdomäne bei Eingabe eines amplitudenmodulierten Signals. Hierbei fällt auf, dass die Leistung des Trägersignals bei 10 000 Hz am stärksten ist. Außerdem sind unmittelbar daneben Summen- und Differenzfrequenzen des Trägersignals mit den beiden Signalfrequenzen bei 9000 Hz, 9300 Hz, 10 700 Hz und 11 000 Hz zu sehen.

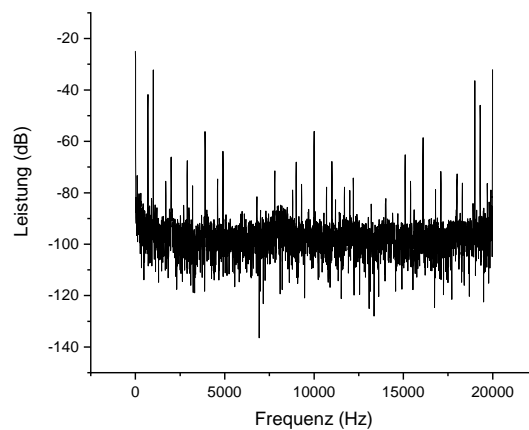
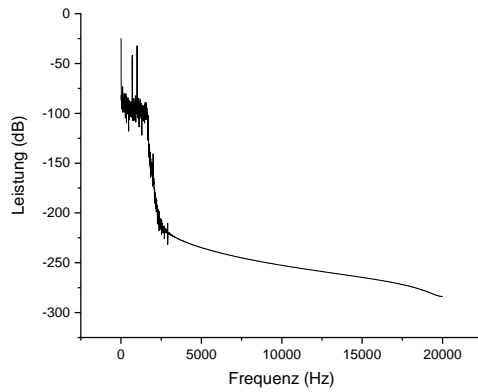
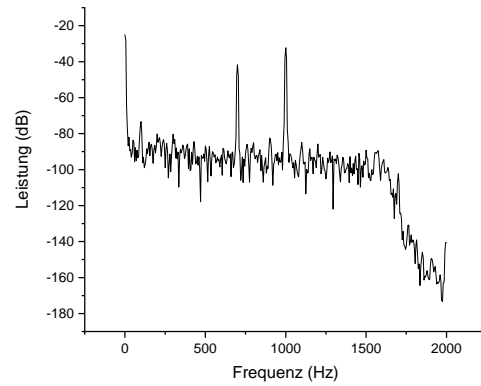


Abbildung 21: Ausgabe des Oszilloskopprogramms in der Frequenzdomäne nach Bilden des Betrags eines amplitudenmodulierten Signals. Die Tiefpassfilterung wurde noch nicht durchgeführt. Dass hierin bei 700 Hz und 1000 Hz bereits die Signalfrequenzen vorhanden sind, kann nur festgestellt werden, wenn bereits bekannt ist, wo diese liegen.





(a) Gesamter Frequenzbereich



(b) Geringer Frequenzbereich

Abbildung 22: Ausgabe des Oszilloskopprogramms in der Frequenzdomäne nach vollständiger Demodulation durch Bilden des Betrags und Tiefpassfilterung des amplitudenmodulierten Signals. Als Grenzfrequenz für den Tiefpass wurde 1600 Hz gewählt. Hierfür muss man voraussetzen, dass bekannt ist, in welchem Frequenzbereich sich die Frequenzen im Signal befinden. In der Darstellung des Frequenzbereichs von 0 Hz bis 2000 Hz sind beide Signalfrequenzen sowie ein signifikanter Gleichanteil zu erkennen.

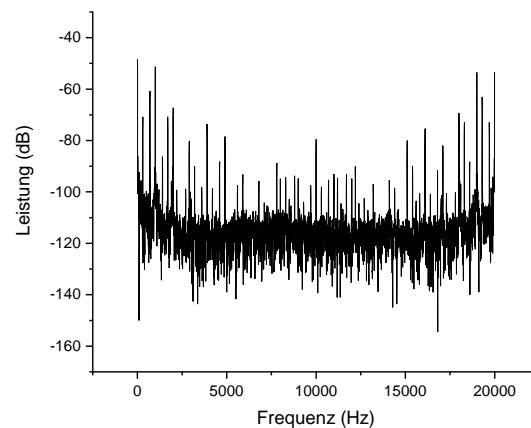
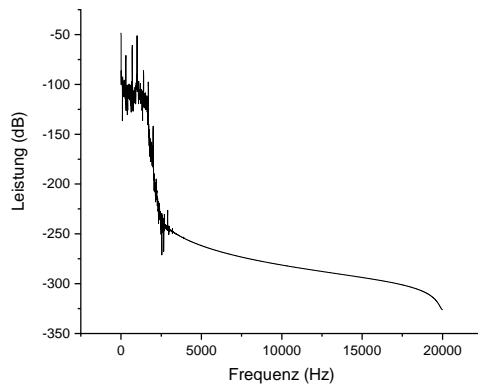
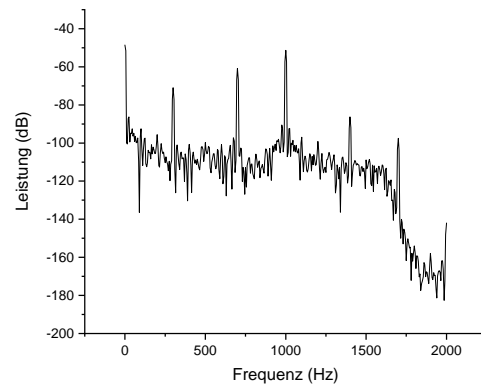


Abbildung 23: Ausgabe des Oszilloskopprogramms in der Frequenzdomäne nach Quadrieren eines amplitudenmodulierten Signals. Die Tiefpassfilterung wurde noch nicht durchgeführt. Dass hierin bei 700 Hz und 1000 Hz bereits die Signalfrequenzen vorhanden sind, kann nur festgestellt werden, wenn bereits bekannt ist, wo diese liegen.



(a) Gesamter Frequenzbereich



(b) Geringer Frequenzbereich

Abbildung 24: Ausgabe des Oszilloskopprogramms in der Frequenzdomäne nach vollständiger Demodulation durch Quadrieren und Tiefpassfilterung des amplitudenmodulierten Signals. Als Grenzfrequenz für den Tiefpass wurde 1600 Hz gewählt. Hierfür muss man voraussetzen, dass bekannt ist, in welchem Frequenzbereich sich die Frequenzen im Signal befinden. In der Darstellung des Frequenzbereichs von 0 Hz bis 2000 Hz sind beide Signalfrequenzen sowie ein signifikanter Gleichanteil zu erkennen. Hier treten allerdings noch einige zusätzliche Frequenzen auf, weshalb die Bildung des Betrages gegenüber dem Quadrieren zu bevorzugen ist.

### 5.3 Demodulation eines AM-Signals mittels Trägerfrequenzmultiplikation

Das allgemeine Ziel dieses Abschnitts ist die Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, amplitudenmodulierten Signals unter Zuhilfenahme der Multiplikation mit der Trägerfrequenz. Mathematisch führt das Multiplizieren der amplitudenmodulierten Signalfunktion mit der Trägerfunktion, welche lediglich der Trägerfrequenz des AM-Signals unterliegt, dazu, dass sich mit dem passenden Additionstheorem als Ergebnis eine Überlagerung bzw. additive Aneinanderreihung von Schwingungstermen unterschiedlichster Frequenzen bildet. Letztere lassen sich unter Verwendung einer Fourier-Transformation im Frequenzspektrum dieser Überlagerung betrachten. Neben den diversen Peaks an Stellen mit vergleichsweise hohen Frequenzen, ergeben sich darin ebenfalls Peaks bei relativ niedrigen Frequenzwerten. Demnach handelt es sich um ein breites Frequenzspektrum, in dem man die Frequenz(en) des modulierten, ursprünglichen Signals deutlich erkennen kann. Damit ist die theoretische Betrachtung der kompletten Demodulation des AM-Signals abgeschlossen. Zusätzlich lassen sich durch einen Tiefpass mit hinreichend großer Grenzfrequenz die hohen Frequenzanteile aus dem Spektrum herausfiltern, sodass letztendlich nur noch die besagten Frequenzen des Signals übrig bleiben, welches

es zu amplitudenmodulieren galt.

Dieses Verfahren soll nun in ein LabVIEW-Programm übertragen werden, dessen zugehöriger Programmcode in Abb. 25 zu sehen ist.

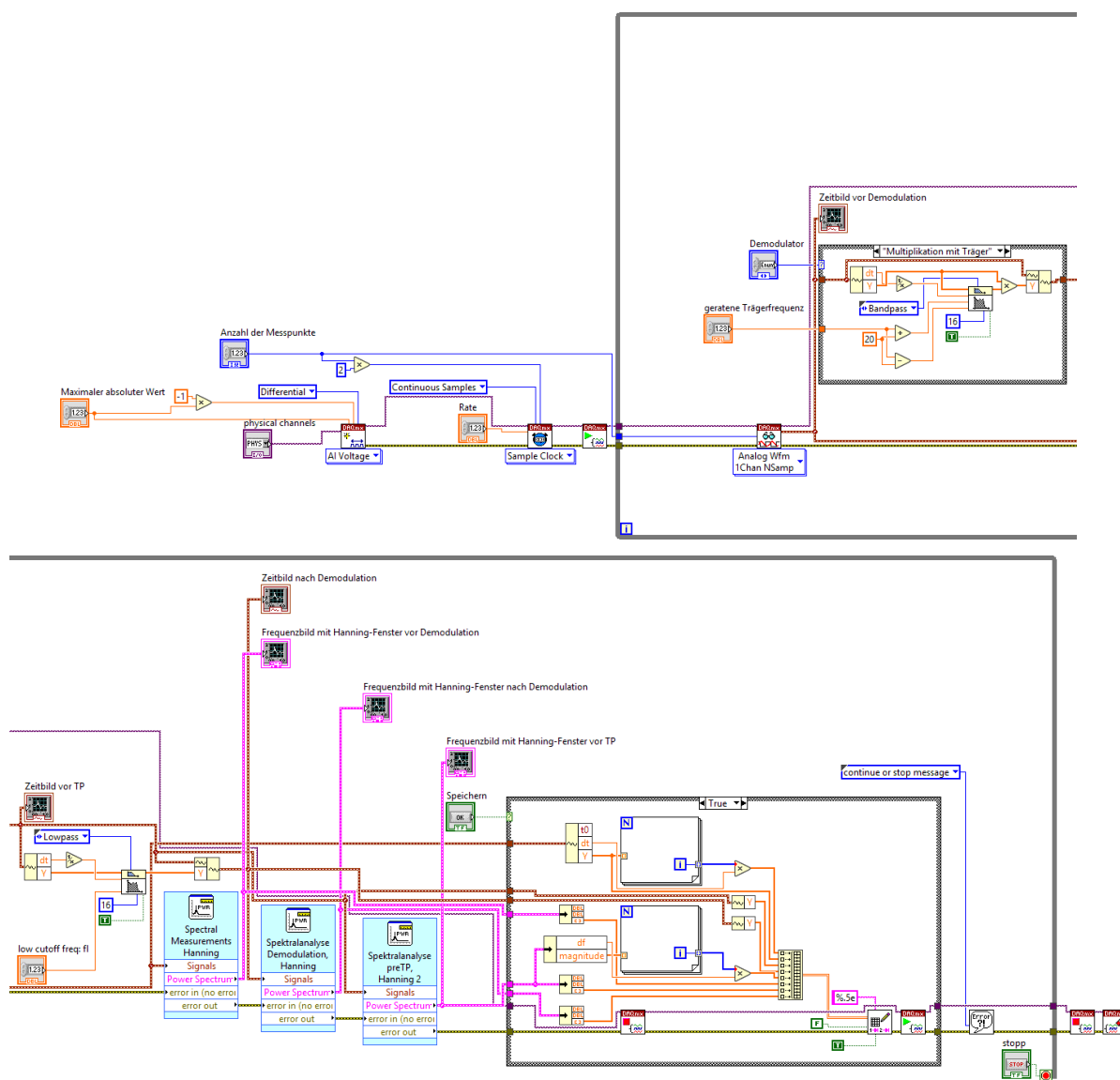


Abbildung 25: Die Abbildung zeigt das LabVIEW-Blockdiagramm bzw. den Programmcode zur Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, amplitudenmodulierten Signals mittels Trägerfrequenzmultiplikation.

Voraussetzung ist dabei das in Abschnitt 5.2 bereits erstellte Programm. Alle zuvor eingefügten Methoden zur Demodulation („Quadrat“ und „Betrag“) sollen frei auswählbar bleiben. Dafür wird zunächst die Demodulator-Case-Konstruktion um eine weitere Option mit dem Namen „Multiplikation mit Träger“ ergänzt und somit ein neues, ausfüllbares

Case-Feld erzeugt. Darin werden diverse, später näher beschriebene Blockdiagrammobjekte angelegt, welche die Demodulation des von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, amplitudenmodulierten Signals umsetzen. Zuerst muss hierbei vom Nutzer die Trägerfrequenz des amplitudenmodulierten Signals geraten werden. Dies entspricht dem Vorgang, bei dem am herkömmlichen UKW-Radio ein Sender eingestellt wird. Dazu wird sowohl ein weiteres Bedienelement auf dem Frontpanel hinzugefügt, als auch ein entsprechendes Blockdiagrammobjekt im Programmcode erstellt. Mit einem Bandpass sechzehnter Ordnung soll die geratene Trägerfrequenz aus dem Frequenzspektrum des AM-Signals herausgefiltert werden. Die obere bzw. untere Grenzfrequenz des Bandpassfilters bildet der geratene Trägerfrequenzwert zuzüglich bzw. abzüglich eines Werts von 20 Hz. Dieser 20 Hz-Wert ist hinreichend groß gewählt, um einerseits Überlappungen mit anderen Frequenz-Peaks zu vermeiden und andererseits die Gesamtheit des Trägerfrequenz-Peaks zu erfassen. Es wird also ein Bereich von  $\pm 20$  Hz um die Trägerfrequenz „herausgeschnitten“. Die dazugehörige Schwingungsfunktion wird nun mit dem unveränderten, gemessenen amplitudenmodulierten Signal multipliziert. Das Ergebnis dieser Operation ist, wie zuvor im Theorie-Teil erwähnt, ein breites Frequenzspektrum mit zahlreichen Peaks an unterschiedlich hohen und niedrigen Frequenzwerten, was im rechten, mittleren Diagramm in Abb. 26 erkennbar ist. Überdies sind in Abb. 26 noch fünf weitere Anzeigeelemente in Diagramm-Form dargestellt: Die beiden Oberen zeigen das von der Computer-Soundkarte ausgegebene und anschließend über den Analog-Digital-Wandler gemessene, amplitudenmodulierte Signal ohne Demodulation, zum einen im Zeitbild und zum anderen im Frequenzbild. Im linken, mittleren Anzeigeelement ist ein Diagramm zu sehen, in dem der zeitliche Signalverlauf nach der Durchführung der Trägerfrequenzmultiplikation und somit auch nach der kompletten Demodulation veranschaulicht ist. Die beiden unteren Anzeigeelemente stellen, sowohl im Zeit- als auch im Frequenzbild, das demodulierte AM-Signal nach der bereits im Theorie-Teil angesprochenen Tiefpassfilterung dar. Genauso wie in Abschnitt 5.2 findet die Tiefpassfilterung außerhalb der Demodulator-Case-Konstruktion statt. Dabei bedarf es eines zusätzlichen Bedienelements auf dem Frontpanel sowie eines dazugehörigen Blockdiagrammobjekts im Programmcode, um die entsprechende Grenzfrequenz für den Tiefpass festzulegen. Diese liegt in Abb. 26 bei 1.600 Hz. Zudem befinden sich auf der Benutzeroberfläche in Abb. 26 sämtliche Bedien- bzw. Eingabelemente oben links.

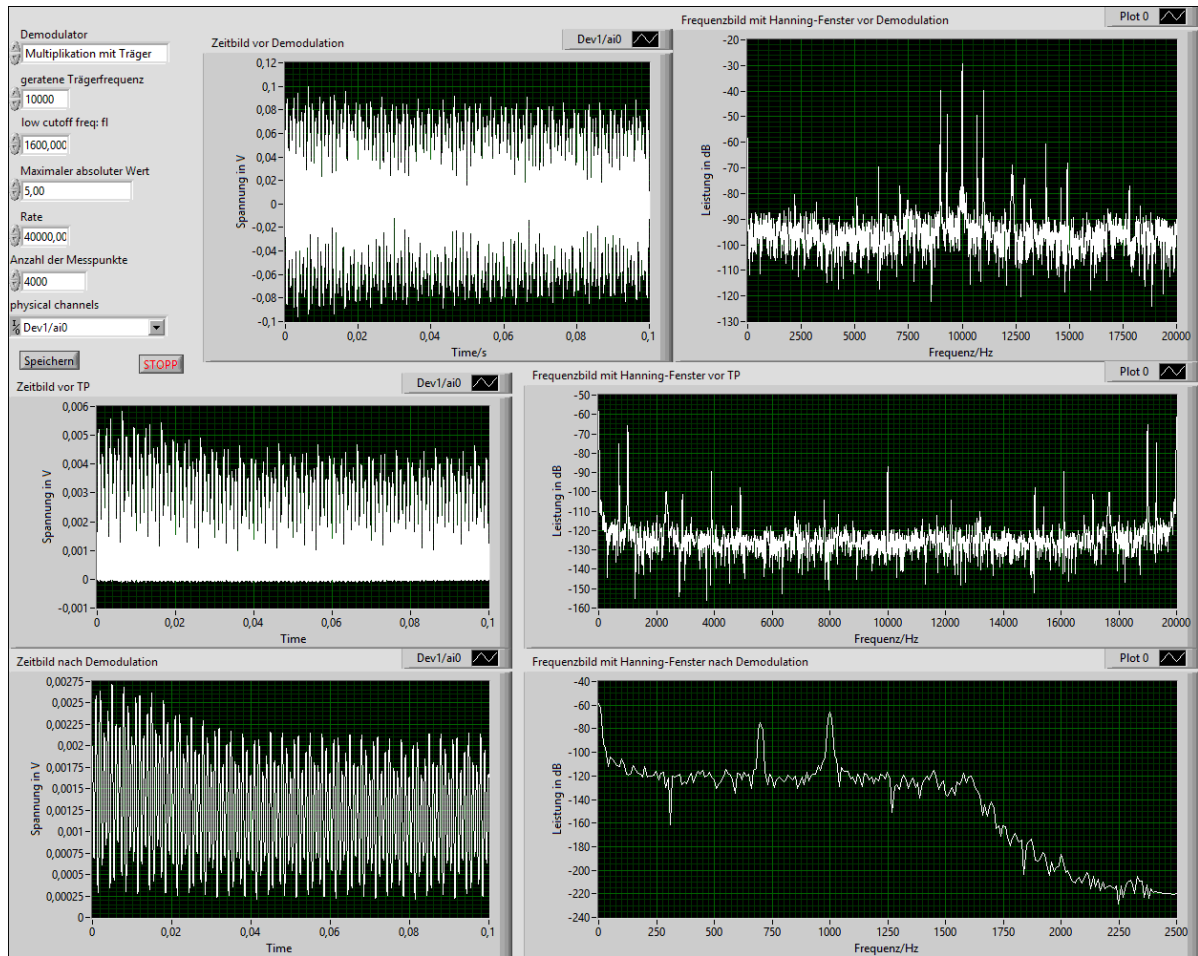


Abbildung 26: Die Abbildung veranschaulicht die LabVIEW-Benutzeroberfläche bzw. das LabVIEW-Frontpanel des Programms zur Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen AM-Signals mittels Trägerfrequenzmultiplikation. Alle sechs Anzeige- bzw. Ausgabeelemente zeigen in zweidimensionalen, kartesischen Diagrammen das an bestimmten Stellen des Bearbeitungsprozesses abgegriffene AM-Signal, jeweils im Zeitbild und im Frequenzbild mit Hanning-Fenster. Die beiden oberen Diagramme stellen das von der Computer-Soundkarte ausgegebene und anschließend über den Analog-Digital-Wandler gemessene AM-Signal vor der Demodulation dar. In den beiden mittleren Diagrammen sieht man das Ergebnis der Demodulation des amplitudenmodulierten Signals ohne Tiefpassfilterung. Die unteren beiden Diagramme veranschaulichen das demodulierte AM-Signal nach der Tiefpassfilterung.

Das für den in Abb. 26 gezeigten Programmdurchlauf verwendete Signal, welches in Abschnitt 5.1 amplitudenmoduliert wurde und dieselbe Gestalt wie in Gleichung (8) besitzt, wird unter anderem durch die Frequenzen  $f_1 = 700 \text{ Hz}$  und  $f_2 = 1.000 \text{ Hz}$  be-

stimmt. Dies lässt sich insbesondere anhand der Bedienelemente-Einstellung des Frontpanels in Abb. 17 nachvollziehen. Die Demodulation des von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, amplitudenmodulierten Signals mittels Trägerfrequenzmultiplikation sowie die anschließende Tiefpassfilterung führten schlussendlich zu dem Frequenzbild, was unten rechts auf der Benutzeroberfläche in Abb. 26 dargestellt ist. Dieses Frequenzspektrum weist zwei deutliche Peaks bei 700 Hz und 1.000 Hz auf, welche genau den  $f_1$ - und  $f_2$ -Werten des zu amplitudenmodulierenden Signals entsprechen. Daher kann man als Fazit sagen, dass es sich wohl um eine erfolgreich funktionierende Amplitudenmodulation sowie Demodulation handelt.

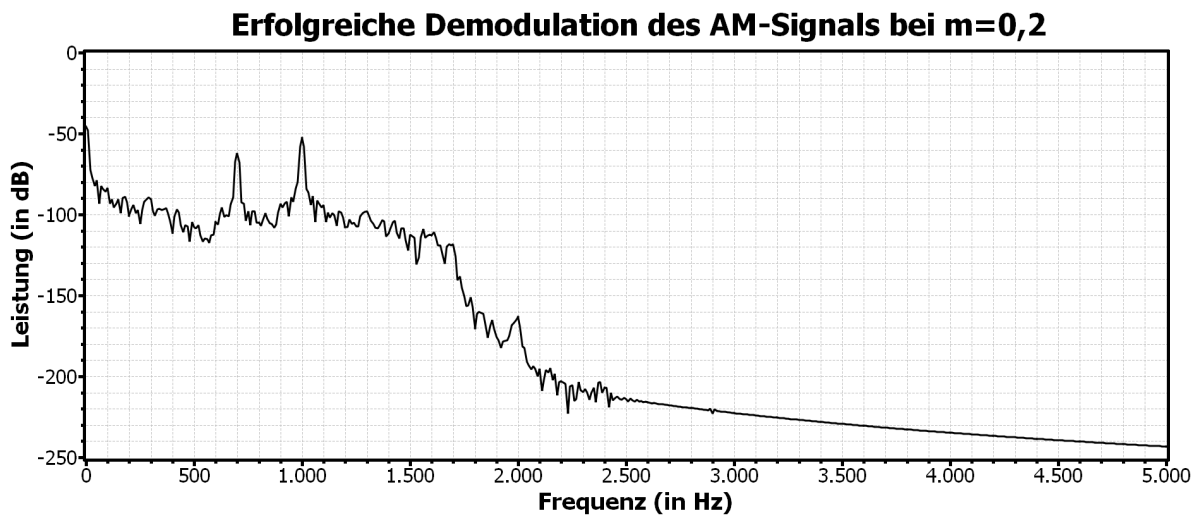


Abbildung 27: In dieser Abbildung ist das Ergebnis einer erfolgreichen Demodulation des von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen AM-Signals zu sehen. Bei diesem Frequenzbild mit Hanning-Fenster handelt sich um das aus Abb. 26 bekannte, rechts unten auffindbare Frontpanel-Diagramm. Der Trägerfrequenz-Wert von 10.000 Hz wurde exakt geraten. Der Modulationsgrad des AM-Signals beträgt 0,2. Die Daten für das hier dargestellte Frequenzspektrum entstammen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen Textdatei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet.

Anzumerken ist noch, dass bei einem größer gewählten Modulationsgrad  $m$  in Gleichung (7) die Peaks im Frequenzspektrum bei den eingestellten  $f_1$ - und  $f_2$ -Werten höher ausfallen und damit leichter zu erkennen sind.

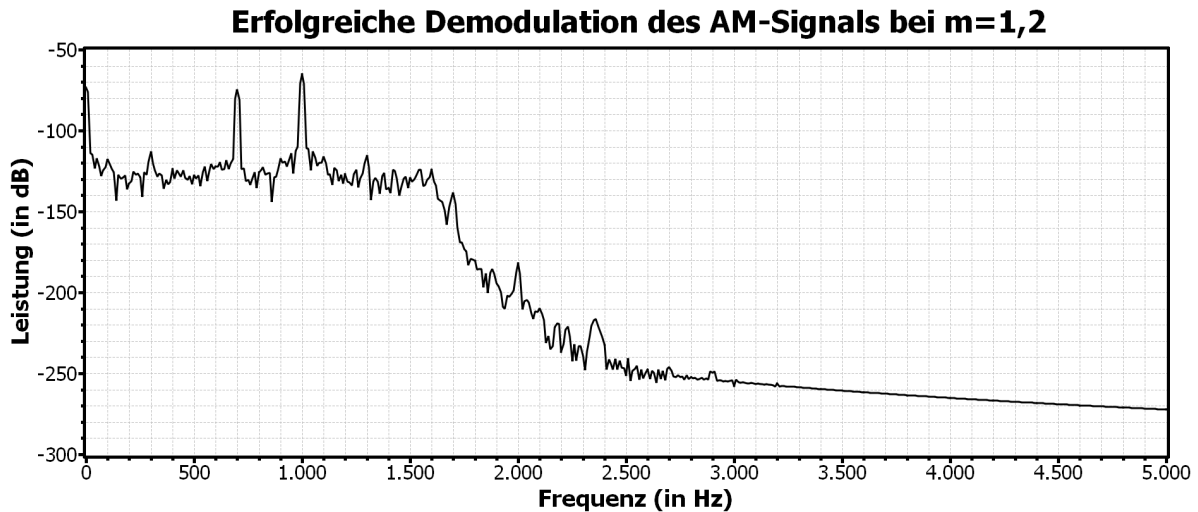


Abbildung 28: Die Abbildung veranschaulicht das Ergebnis einer erfolgreichen Demodulation des von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen AM-Signals bei einem Modulationsgrad von  $m = 1,2$ . Bei diesem Frequenzbild mit Hanning-Fenster handelt es sich um das aus Abb. 26 bekannte, rechts unten auffindbare Frontpanel-Diagramm. Der Trägerfrequenz-Wert von 10.000 Hz wurde exakt geraten. Die Daten für das hier dargestellte Frequenzspektrum entstammen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen txt-Datei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet.

Dieser Effekt wird vor allem beim Vergleich von Abb. 27 mit Abb. 28 deutlich. Hierbei liegt eine Versechsfachung des Modulationsgrads  $m$  vor.

Anhand der Abb. 29 lässt sich erkennen, welche Folgen sich beim Versuch der Demodulation ergeben, sobald man die (geratene) Trägerfrequenz zu weit außerhalb des  $\pm 20$  Hz-Bereichs wählt. Hierbei wurde der Trägerfrequenz-Wert nämlich auf 10.050 Hz geraten, obwohl die tatsächliche Trägerfrequenz des AM-Signals bei 10.000 Hz liegt. Bei dem in Abb. 29 gezeigten Frequenzbild mit Hanning-Fenster, welches äquivalent zum aus Abb. 26 bekannten, sich rechts unten befindenden Frontpanel-Diagramm ist, kann man daher nicht die beiden vom ursprünglichen Signal stammenden Frequenzen  $f_1 = 700$  Hz und  $f_2 = 1.000$  Hz ablesen. Denn bei diesen Frequenz-Werten sind keine Peaks zu sehen. Stattdessen lassen sich in dem Teil des Frequenzspektrums, der nach der Tiefpassfilterung übrig geblieben ist, einige verstreute Peaks unterschiedlicher Höhe und Breite an verschiedenen Frequenz-Werten lokalisieren. Folglich ist die Demodulation mittels Trägerfrequenzmultiplikation in diesem Fall nicht gelungen.

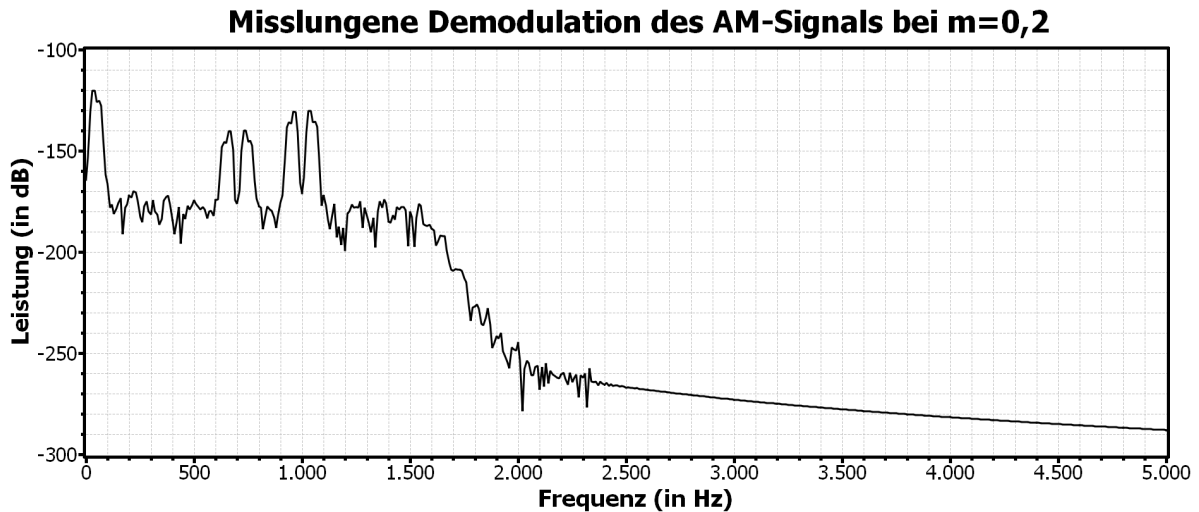


Abbildung 29: Die Abbildung zeigt das Ergebnis einer misslungenen Demodulation des von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen AM-Signals. Der geratenen Trägerfrequenz-Wert liegt bei 10.050 Hz, die tatsächliche Trägerfrequenz des AM-Signals beträgt jedoch 10.000 Hz. Der Wert des Modulationsgrads lautet 0,2. Die Daten für das hier dargestellte Frequenzspektrum entstammen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen txt-Datei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet.



## 6 Phasen- und frequenzmodulierte Signale

### 6.1 Erzeugung eines phasen- bzw. frequenzmodulierten Signals

In diesem Abschnitt soll ein LabVIEW-Programm entstehen, welches ein Signal der in Gleichung (8) präsentierten Form zunächst wahlweise phasen- oder frequenzmoduliert und anschließend über die Soundkarte des verwendeten Computers ausgibt. Der daraus hervorgehende Programmcode ist in Abb. 30 und Abb. 33 aufgeführt.

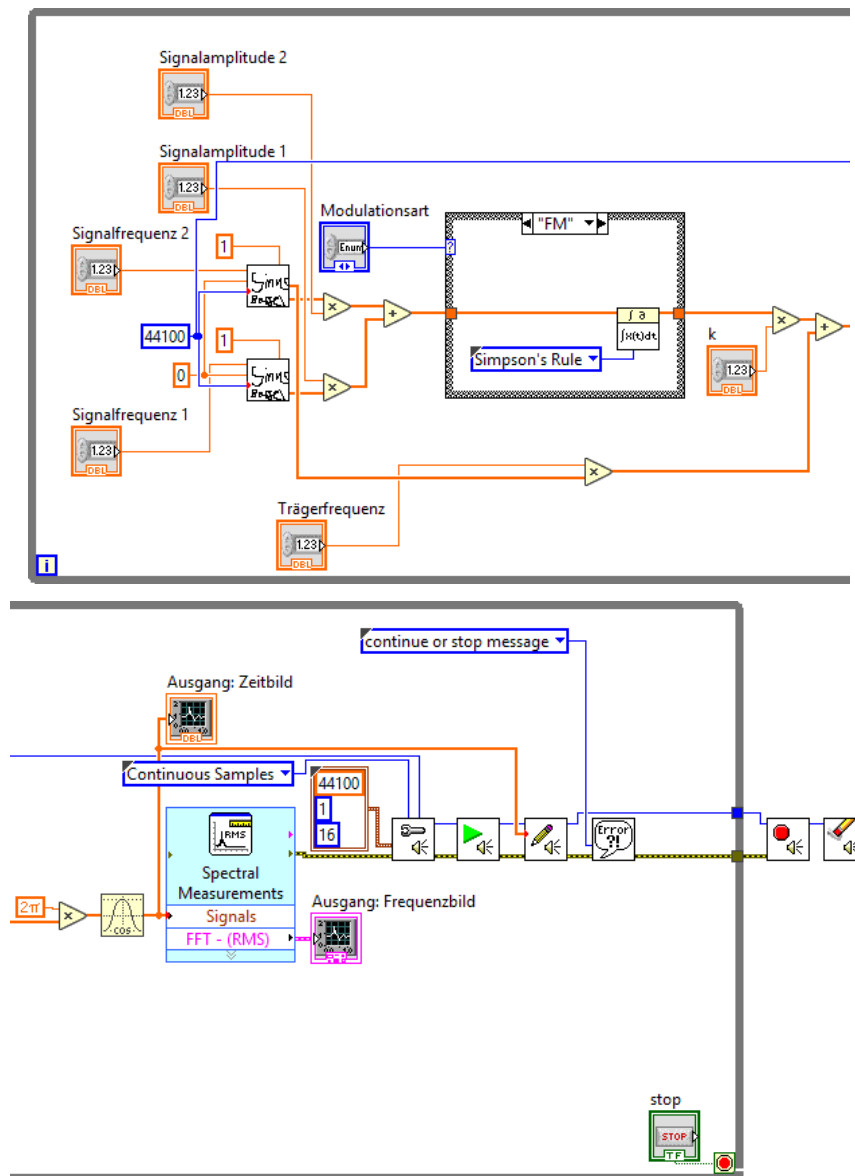


Abbildung 30: In dieser Abbildung ist das LabVIEW-Blockdiagramm bzw. der Programmcode zu sehen, mit dem ein Signal zuerst wahlweise phasen- oder frequenzmoduliert und anschließend über die Computer-Soundkarte ausgegeben wird.

Bei der dazugehörigen Benutzeroberfläche, welche in Abb. 31, Abb. 32 sowie Abb. 34 dargestellt ist, befinden sich alle Bedien- bzw. Eingabeelemente auf der linken und alle Anzeige- bzw. Ausgabeelemente auf der rechten Seite.

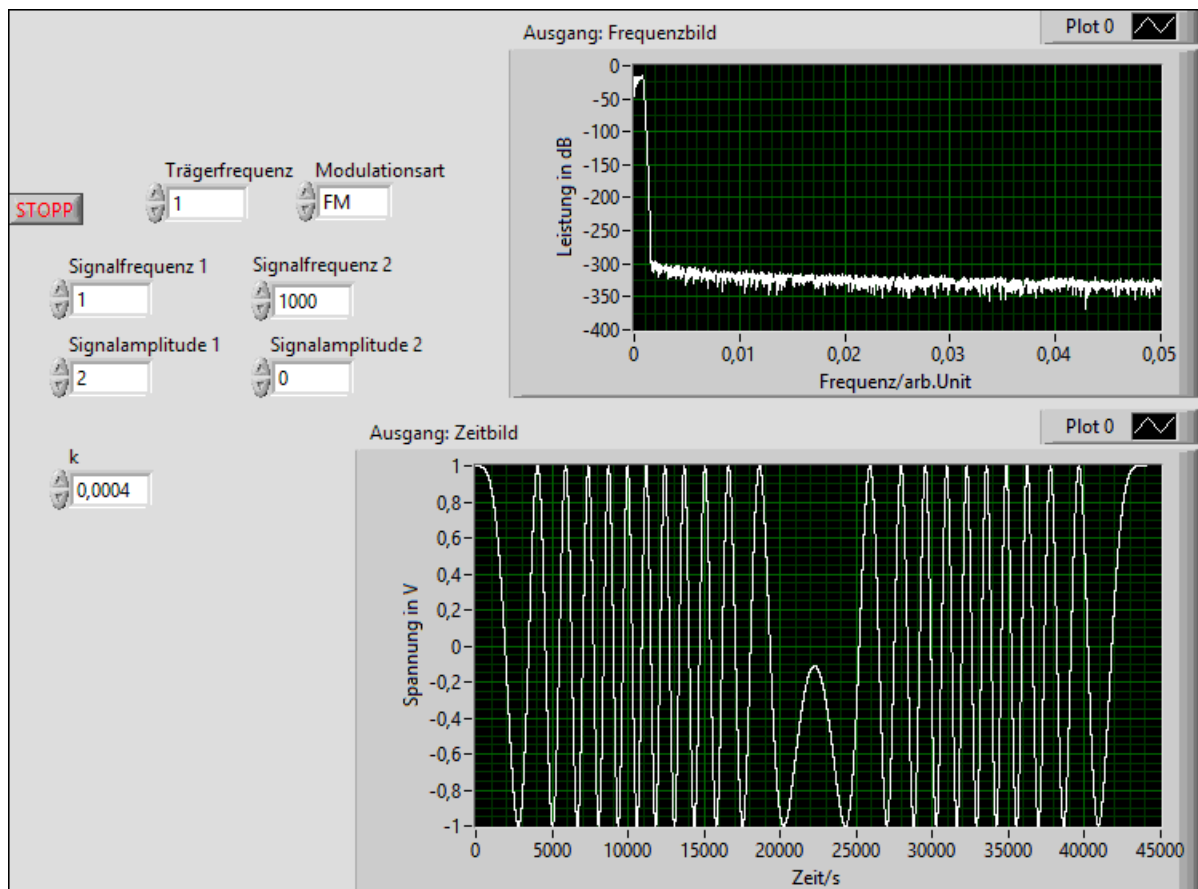


Abbildung 31: Die Abbildung zeigt das Frontpanel bzw. die Benutzeroberfläche des LabVIEW-Programms zur Umsetzung der Frequenzmodulation. Der hierbei verwendete  $k$ -Wert ist sehr klein. Allgemein besteht mit dem entsprechenden, sich oben auf dem Frontpanel befindenden Bedienelement die Möglichkeit die Art der Modulation auszuwählen: Entweder Frequenzmodulation (Abkürzung: FM) oder Phasenmodulation (Abkürzung: PM).

In diesem Fall handelt es sich bei den Anzeigeelementen um zwei Diagramme, die das phasen- oder frequenzmodulierte Signal, je nachdem welche Modulationsart ausgewählt ist, im Zeit- und Frequenzbild anzeigen. Die Blockdiagrammobjekte, welche die Ausgabe über die Computer-Soundkarte realisieren, liegen rechtsseitig im Programmcode und teilweise außerhalb der While-Schleife. Deren Anordnung und Funktionsweise ist die gleiche wie beim LabVIEW-Programm in Abschnitt 5.1. Genauso wie in Abb. 16 aus Abschnitt 5.1 sieht man auf der linken Seite des Programmcodes zunächst die Erzeugung des Signals  $s(t)$  gemäß Gleichung (8) mit den Frequenzen  $f_1$  und  $f_2$ . Weiterhin ist dabei die zusätzliche, aufgrund des in Abschnitt 1 aufgetretenen Programmierfehlers

notwendige nachträgliche Multiplikation mit der Amplitude zu beachten. Außerdem soll  $N = 44.100$  gelten. Da der Vorgang der  $s(t)$ -Konstruktion komplett identisch zu der in Abschnitt 5.1 bereits ausführlich beschriebenen Vorgehensweise ist, wird an dieser Stelle nicht weiter auf die Signalentstehung eingegangen. Die gesamte Berechnung (FM- und PM-Modulation sowie Signalerzeugung) und der Großteil der Signalausgabe finden in einer While-Schleife statt, was auch aus Abb. 30 hervorgeht. Ebenso wie bei den zuvor behandelten Abschnitten ist der While-Schleife ein „Stopp“-Knopf beigelegt, um das Programm bei Bedarf anhalten zu können. Zwischen den Blockdiagrammobjekten zur Benutzeroberflächenkonfiguration und der Signalentstehung findet im Programmcode die Modulation des Signals statt. Durch das Hinzufügen einer Case-Konstruktion kommt die Auswahlmöglichkeit zwischen Frequenz- und Phasenmodulation zustande. Dazu werden zwei Case-Optionen mit den jeweiligen Markierungen „FM“ und „PM“ erstellt. Die zwei dadurch erscheinenden, zunächst freien sowie ausfüllbaren Case-Felder sollen somit jeweils der Frequenzmodulation und der Phasenmodulation zur Verfügung stehen. Der Wechsel zwischen diesen beiden erfolgt durch ein mit dem Namen „Modulationsart“ gekennzeichnetes Bedienelement auf der Benutzeroberfläche samt dazugehörigem Element (auch als „Control“ bezeichnet) an der entsprechenden Stelle im Programmcode.

Im Folgenden wird zwischen Frequenz- und Phasenmodulation („FM“ und „PM“) unterschieden. Außerdem wird die Funktionsweise des LabVIEW-Programms näher erläutert.

Die Frequenzmodulation des Signals  $s(t)$  soll gemäß der Gleichung

$$S_{FM}(t) = \cos(2\pi(f_0 t + k \cdot \int s(t) dt)) \quad (9)$$

geschehen, wobei  $t$  die Zeit,  $f_0$  die Trägerfrequenz und  $k$  eine beliebige Konstante ist, welche die Stärke der Modulation bestimmt. Am Programmcode in Abb. 30 lässt sich die unter Verwendung der von LabVIEW zur Verfügung gestellten numerischen Operationen, Funktionen und Konstanten erfolgte Umsetzung dieser Formel betrachten. Unter anderem können die dafür benötigten  $t$ -Werte, welche nach Gleichung (2) für alle  $i = 0, 1, 2, \dots, 44.100$  gegeben sind, am Ausgangsanschluss des aus Abschnitt 1 stammenden, Sinus-erstellenden Programms abgegriffen werden. Das dazugehörige Blockdiagrammobjekt befindet sich im linken Programmcode-Bereich, welcher der Signalerzeugung dient. Durch zusätzliche „Controls“ und entsprechende Bedienelemente auf dem Frontpanel wird die Eingabe der Gleitkommazahlenwerte für  $f_0$  und  $k$  realisierbar. Innerhalb des für die Frequenzmodulation reservierten, freien, ausfüllbaren Case-Feldes („FM“) wird die in Gleichung (9) auftretende Integration von  $s(t)$  ermöglicht. Dabei wird auf die Simpsonregel (in Abb. 30 mit „Simpson’s Rule“ gekennzeichnet) als numerisches Integrationsverfahren zurückgegriffen.

Vergleicht man die Diagramme des frequenzmodulierten Signals in Abb. 31 mit denen aus Abb. 32, so lässt sich feststellen, dass im Zeitbild die tausendfach höhere Trägerfrequenz deutlich zum Tragen kommt, sodass zahlreiche Nulldurchgänge stattfinden.

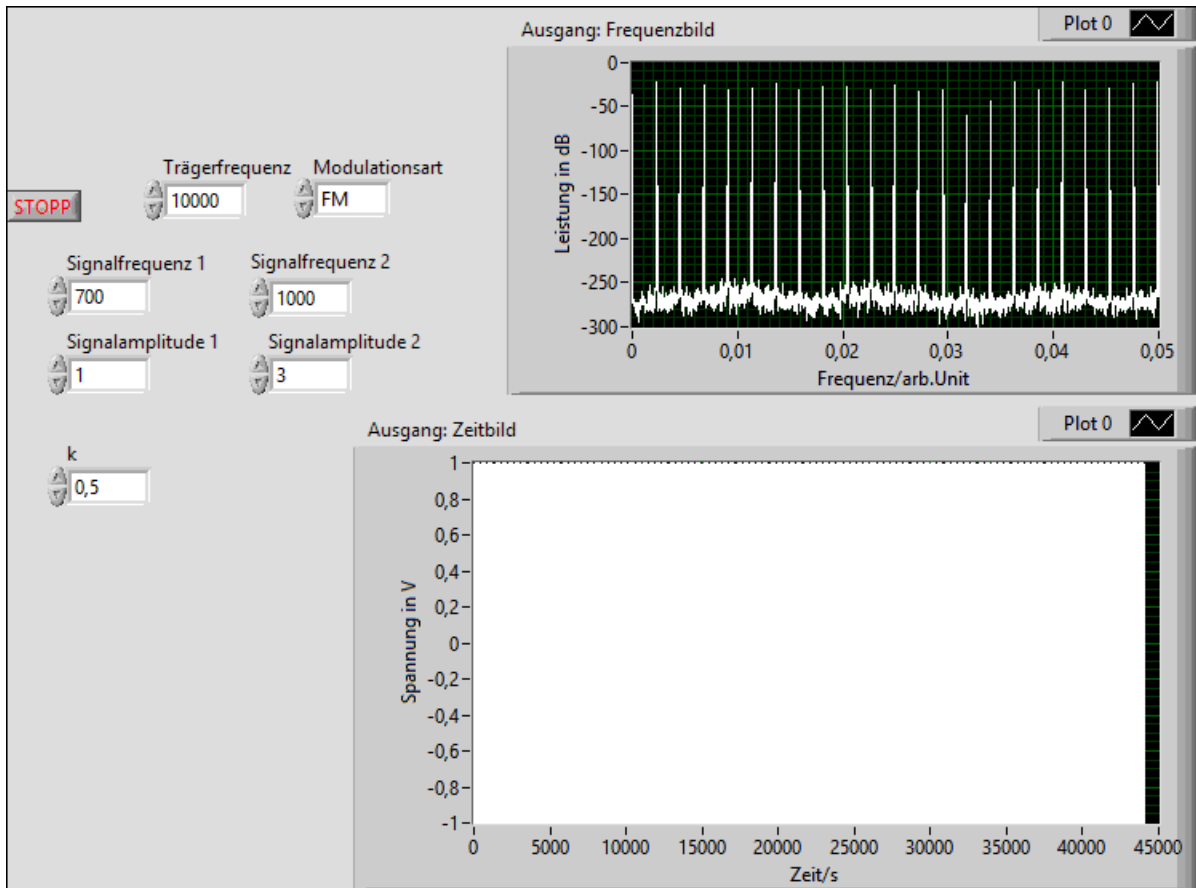


Abbildung 32: In der Abbildung ist das Frontpanel bzw. die Benutzeroberfläche des LabVIEW-Programms zur Realisierung der Frequenzmodulation zu sehen. Der dabei betrachtete  $k$ -Wert ist sehr groß gewählt, weshalb die Peaks im Frequenzbild sehr stark ausgeprägt sind.

Der Graph erscheint daher wie ein „weißer Block“ aus Linien. Der Vergleich der beiden Frequenzbilder zeigt, dass die Vergrößerung des  $k$ -Wertes zu schärfer definierten Peaks im Frequenzspektrum führt. Dies geht auch aus Gleichung (9) hervor.

Phasenmoduliert man das Signal  $s(t)$ , so nimmt das daraus entstehende PM-Signal  $S_{PM}(t)$  die Form

$$S_{PM}(t) = \cos(2\pi(f_0 t + k \cdot s(t))) \quad (10)$$

an. Dabei bildet  $f_0$  die Trägerfrequenz und  $t$  die Zeit.  $k$  ist ein fester, aber beliebig wählbarer Faktor. Aufgrund der Ähnlichkeit von Gleichung (10) zu Gleichung (9), verläuft auch die Realisierung der Phasenmodulation als LabVIEW-Programm ähnlich zur Programm-Umsetzung der Frequenzmodulation. Daher besteht lediglich die Notwendigkeit, das für die Phasenmodulation reservierte, freie, ausfüllbare Case-Feld passend zu besetzen.

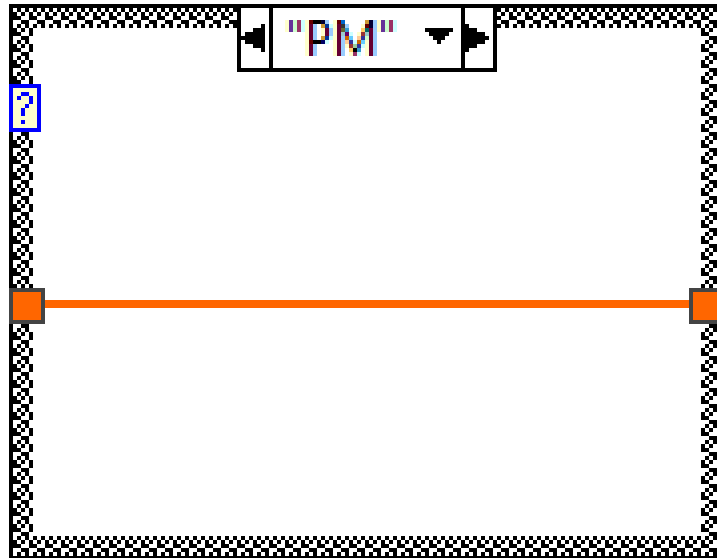


Abbildung 33: Es ist der in Abb. 30 nicht sichtbare Teil des LabVIEW-Programmcodes dargestellt, der notwendig ist, um die Art der Modulation auf der Benutzeroberfläche des Programms auszuwählen zu können. Bei diesem hier abgebildeten Programmcodeteil handelt es sich um eine weitere Option in der Case-Konstruktion, welche darauf abzielt die Phasenmodulation zu ergänzen.

Wie im Programmcode in Abb. 33 zu erkennen, wird hierbei die Leitung direkt durch das Case-Feld gezogen, ohne das übertragene Signal zu bearbeiten oder zu verändern. In Kombination mit der bereits durch die Frequenzmodulation existierenden Programmstruktur ist damit die Umsetzung der Phasenmodulation abgeschlossen.

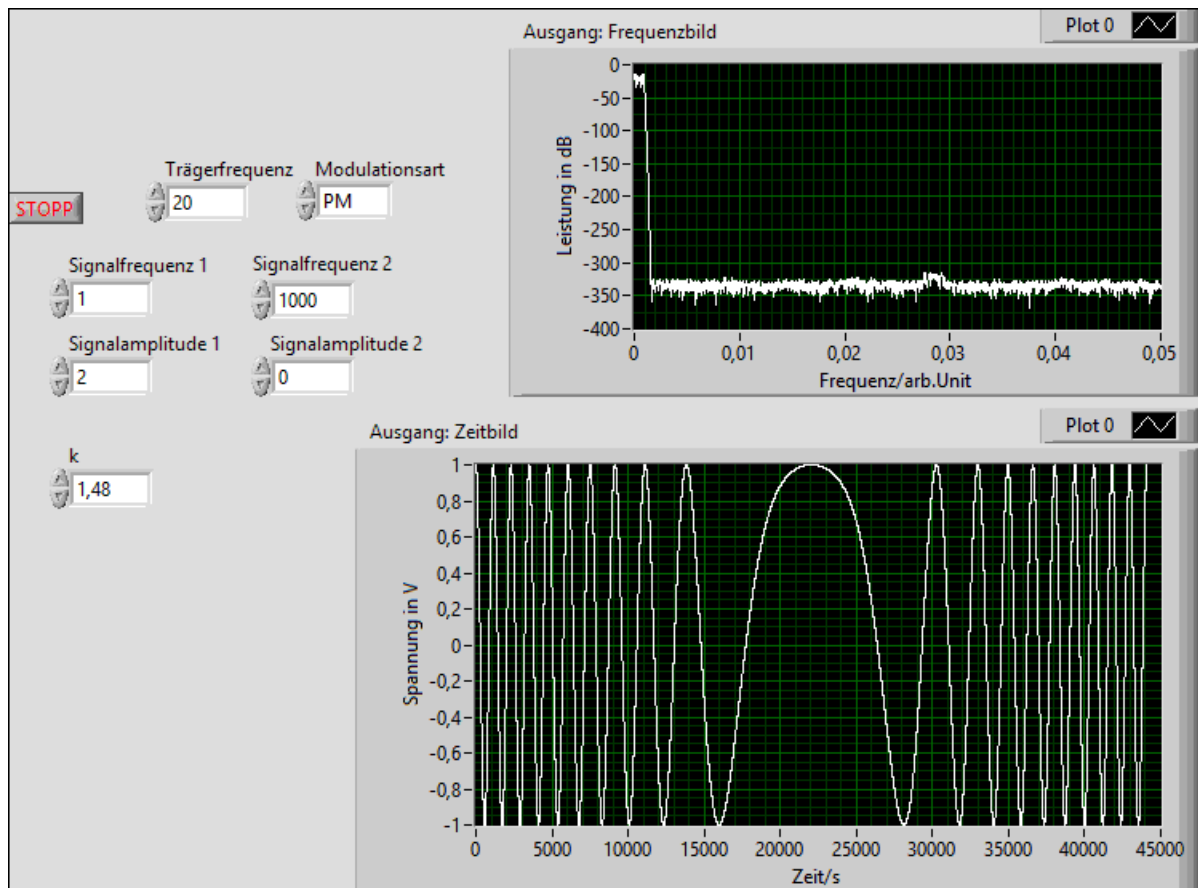


Abbildung 34: Die Abbildung zeigt das Frontpanel bzw. die Benutzeroberfläche des LabVIEW-Programms zur Realisierung der Phasenmodulation.

Das sich ergebene phasenmodulierte Signal  $S_{PM}(t)$  sowie dessen Fourier-Transformierte  $\tilde{S}_{PM}(f)$  sind in den beiden Diagrammen auf der Benutzeroberfläche des Programms in Abb. 34 dargestellt.

*Hinweis:* Bei diesem LabVIEW-Programm zur Erzeugung von phasen- und frequenzmodulierten Signalen ist zu beachten, dass zwischen den  $k$ -Werten der Frequenzmodulation und den  $k$ -Werten der Phasenmodulation stets eine Diskrepanz um einen Faktor der Größenordnung  $10^4$  vorliegt. Denn dadurch ergeben sich für beide Modulationsarten die verlässlichsten Ergebnisse.

## 6.2 Demodulation eines phasen- bzw. frequenzmodulierten Signals

Im Folgenden soll ein von der Computer-Soundkarte ausgegebenes (im Hintergrund wird das in Abschnitt 6.1 entstandene LabVIEW-Programm ausgeführt) und anschließend über den Analog-Digital-Wandler gemessenes Signal, welches entweder phasen- oder frequenzmoduliert wurde, demoduliert werden. Der Programmcode des dafür erstellten LabVIEW-Programms ist in Abb. 35 und Abb. 38 zu sehen.

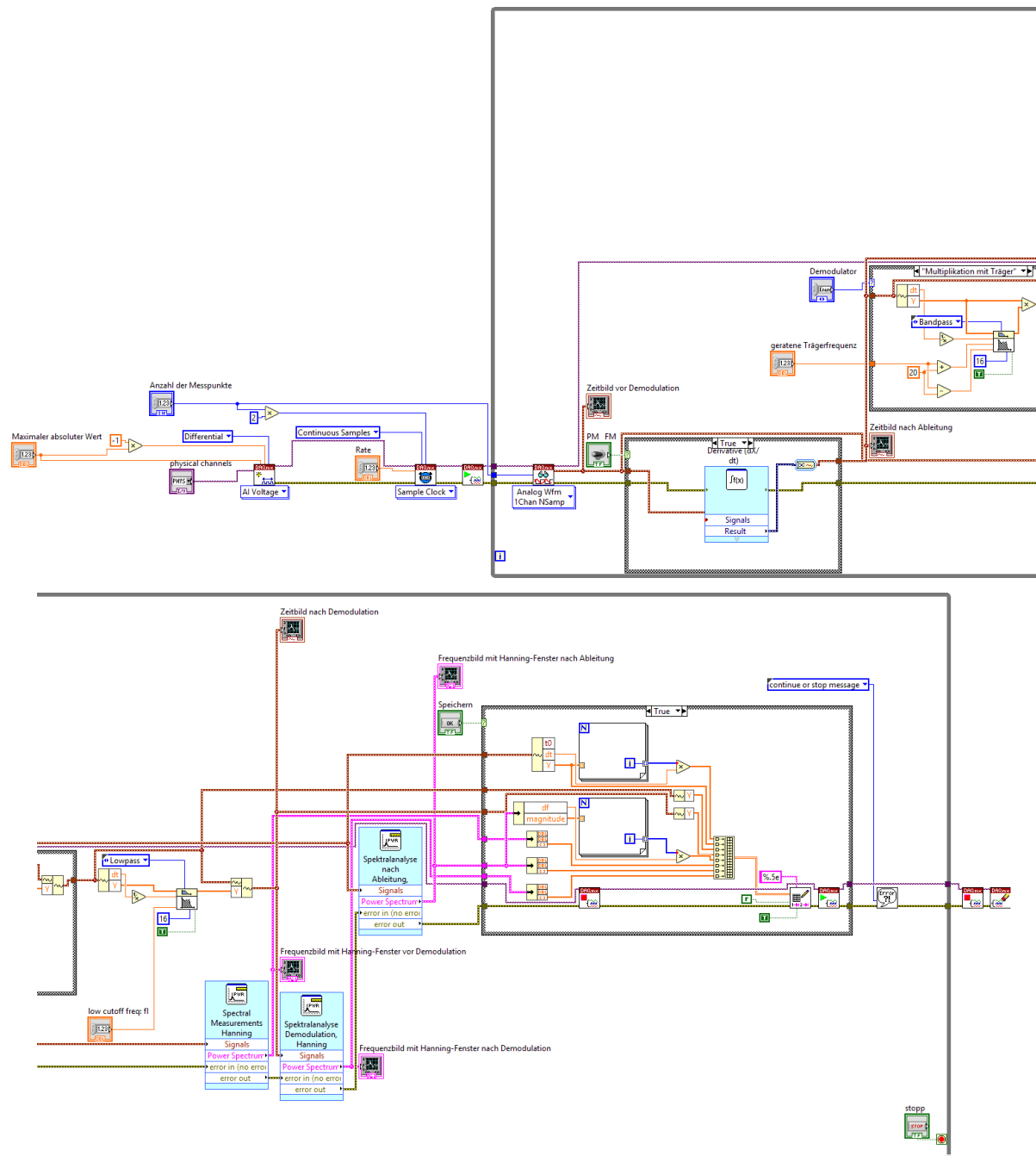


Abbildung 35: Die Abbildung veranschaulicht das LabVIEW-Blockdiagramm bzw. den Programmcode zur Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, phasen- bzw. frequenzmodulierten Signals.

Die Abb. 36 zeigt das dazugehörige Frontpanel. Das in Abschnitt 5.3 erhaltene LabVIEW-Programm dient dabei als Vorlage. Von links nach rechts gelesen, beinhaltet dessen Programmcode, welcher in Abb. 25 dargestellt ist, einen Programmteil zur Messdatenauf-

nahme mittels Analog-Digital-Wandler, einen für die Demodulation von AM-Signalen zuständigen Programmbereich, einen Tiefpassfilter sechzehnter Ordnung mit einer Grenzfrequenz von 1.600 Hz, einige Blockdiagrammobjekte zur Realisierung der Anzeige- bzw. Ausgabeelemente auf dem Frontpanel sowie eine Speicher- und eine Anhalte-Funktion für das Programm mit den entsprechenden „Speichern“- und „Stopp“-Knöpfen auf der Benutzeroberfläche.

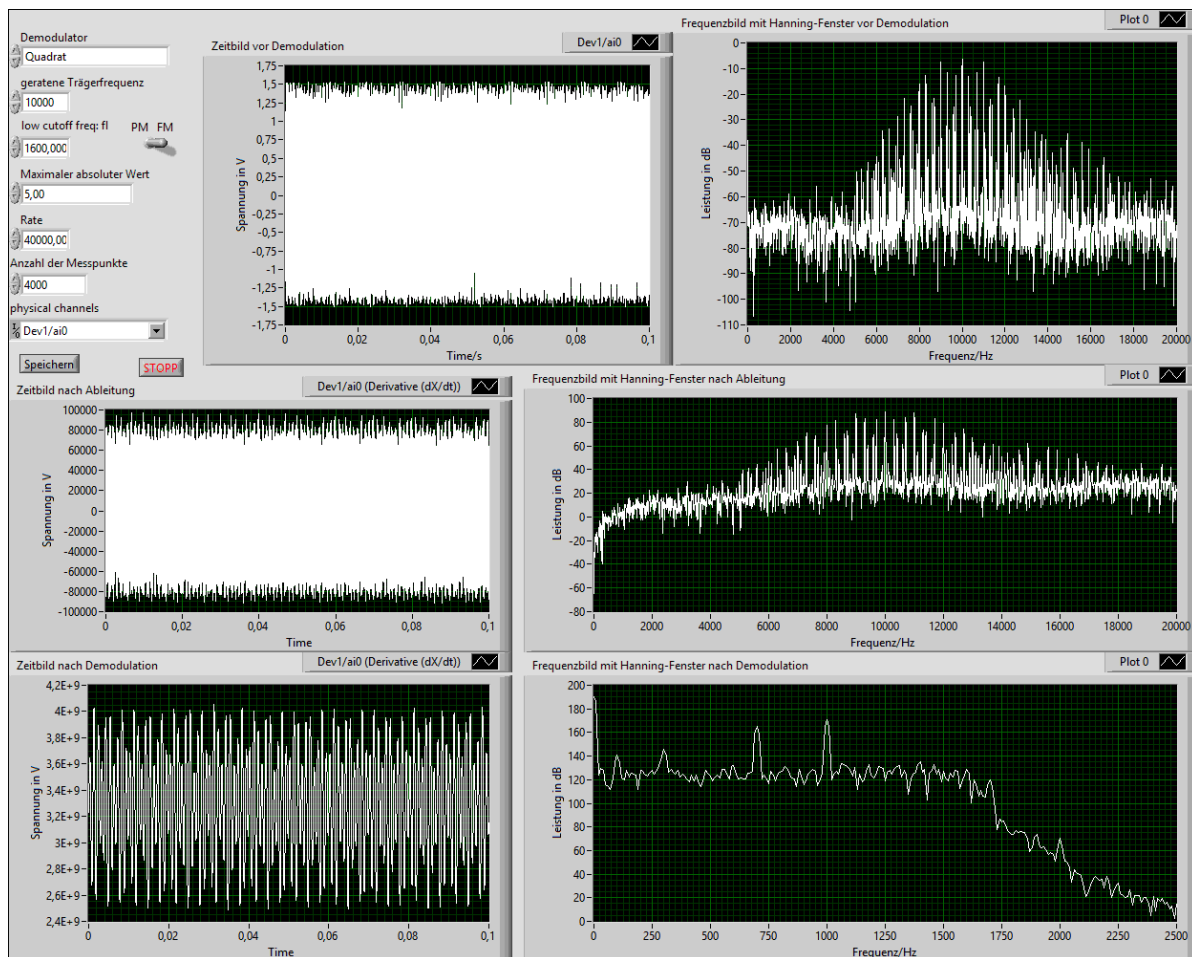


Abbildung 36: In dieser Abbildung ist die LabVIEW-Benutzeroberfläche bzw. das LabVIEW-Frontpanel des Programms zur Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, frequenzmodulierten Signals („FM“) dargestellt. Die AM-Demodulation findet hierbei mittels „Quadrat“-Methode statt.

Mit den passenden Erweiterungen und Veränderungen des Programmcodes lässt sich nun die Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen FM- bzw. PM-Signals umsetzen. Dazu wird eine Case-Konstruktion zwischen dem Programmteil zur Messdatenaufnahme und



dem Programmbereich für die Demodulation von amplitudenmodulierten Signalen hinzugefügt, welche insgesamt zwei Case-Optionen umfasst. Auf diese Weise entstehen zwei zunächst freie, ausfüllbare Case-Felder, wobei das Erste für die Frequenzmodulation und das Zweite für die Phasenmodulation vorgesehen ist. Der Wechsel zwischen diesen beiden Optionen findet dabei mit Hilfe eines sich auf der Benutzeroberfläche befindenden Kippschalters statt, welcher die der Case-Feld-Reservierungen entsprechenden Markierungen „FM“ und „PM“ besitzt.

Zur Demodulation eines gemessenen, frequenzmodulierten Signals wird die FM-Signalfunktion  $S_{FM}(t)$  zunächst differenziert bzw. abgeleitet. Im Programmcode in Abb. 35 wird diese numerische Differentiation innerhalb des für die Frequenzmodulation reservierten Case-Feldes („FM“) realisiert. Das abgeleitete FM-Signal kann anschließend als AM-Signal betrachtet werden. Daher lässt es sich nun mit Hilfe einer der drei bereits aus Abschnitt 5.3 bekannten und auch hier zur Verfügung stehenden AM-Demodulationsmethoden („Quadrat“, „Betrag“ und „Multiplikation mit Träger“) demodulieren. Anhand der auf dem Frontpanel in Abb. 36 ersichtlichen Anzeigeelemente kann der Demodulationsprozess eines FM-Signals verfolgt werden: Die beiden oberen Diagramme bilden das Zeitbild und das Frequenzbild mit Hanning-Fenster des gemessenen FM-Signals vor der Demodulation. In der Mitte befinden sich zwei Diagramme, welche das Zeitbild und das Frequenzbild mit Hanning-Fenster des abgeleiteten FM-Signals veranschaulichen.

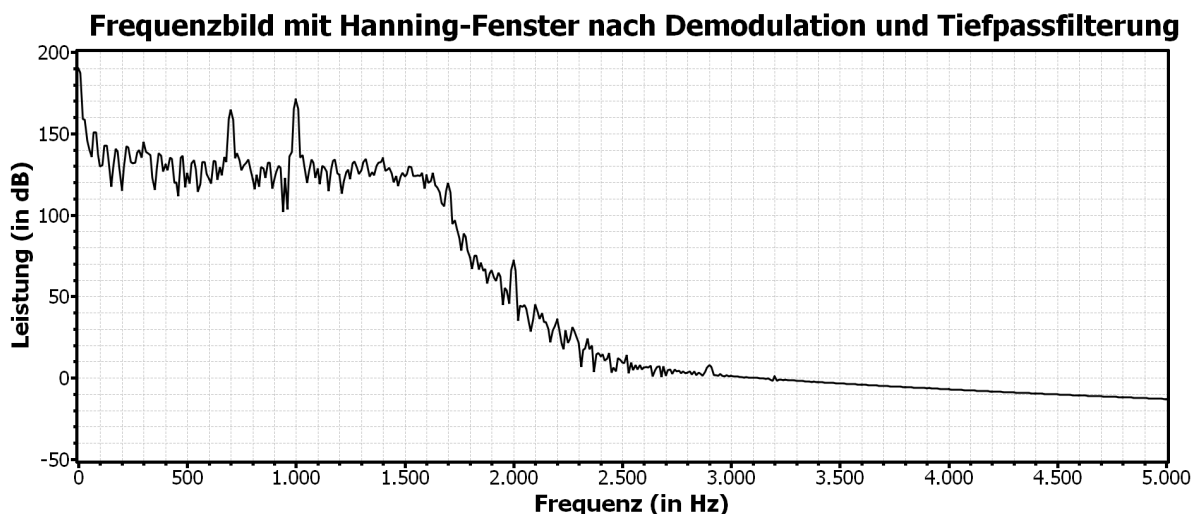


Abbildung 37: Die Abbildung zeigt das Frequenzspektrum eines FM-Signals mit  $k = 0,005$  nach Abschluss der Demodulation und Tiefpassfilterung. Bei diesem Frequenzbild mit Hanning-Fenster handelt es sich um das Äquivalent zu dem aus Abb. 36 bekannten, sich rechts unten befindenden Frontpanel-Diagramm. Die Daten für das hier dargestellte Frequenzspektrum entspringen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen txt-Datei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet. Die verwendete AM-Demodulation findet mittels „Quadrat“-Methode statt.

Die beiden unteren Diagramme stellen das Zeitbild und das Frequenzbild mit Hanning-Fenster des demodulierten FM-Signals nach der Tiefpassfilterung dar. Ebenjenes Frequenzbild mit Hanning-Fenster und das in Abb. 37 zu sehende Diagramm bilden somit das Ergebnis der Demodulation des FM-Signals, anhand derer man auch erkennen kann, dass es sich um eine erfolgreich umgesetzte FM-Demodulation handelt. Denn die beiden deutlich sichtbaren Peaks im Frequenzspektrum liegen bei 700 Hz und 1.000 Hz, was genau den Frequenzen  $f_1$  und  $f_2$  des Signals  $s(t)$  entspricht, welches es zu frequenzmodulieren galt.

Im Gegensatz zur FM-Demodulation wird bei der Demodulation eines gemessenen, phasenmodulierten Signals dieses PM-Signal von vornherein als AM-Signal betrachtet. Wie im Programmcodeteil in Abb. 38 zu sehen ist, werden die Leitungen daher direkt durch das für die Phasenmodulation reservierte Case-Feld („PM“) gezogen, sodass das übertragene Signal nicht verändert oder beeinflusst wird.

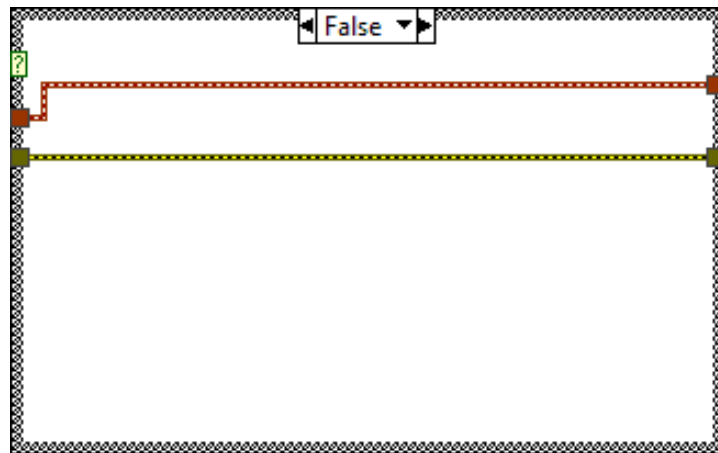


Abbildung 38: In dieser Abbildung ist der in Abb. 35 nicht sichtbare Teil des LabVIEW-Programmcodes zu sehen, der notwendig ist, um auf der Benutzeroberfläche des Programms zwischen der Demodulation des FM-Signals und der Demodulation des PM-Signals wechseln zu können. Bei diesem hier abgebildeten Programmcodeteil handelt es sich um die zweite Option in der Case-Konstruktion, mit dessen Hilfe die Demodulation des phasenmodulierten Signals umgesetzt wird.

Dementsprechend durchläuft das PM-Signal anschließend eine AM-Demodulation nach den bereits erwähnten, verfügbaren AM-Demodulationsmethoden. Abb. 39 zeigt das Ergebnis der Demodulation und Tiefpassfilterung des PM-Signals.

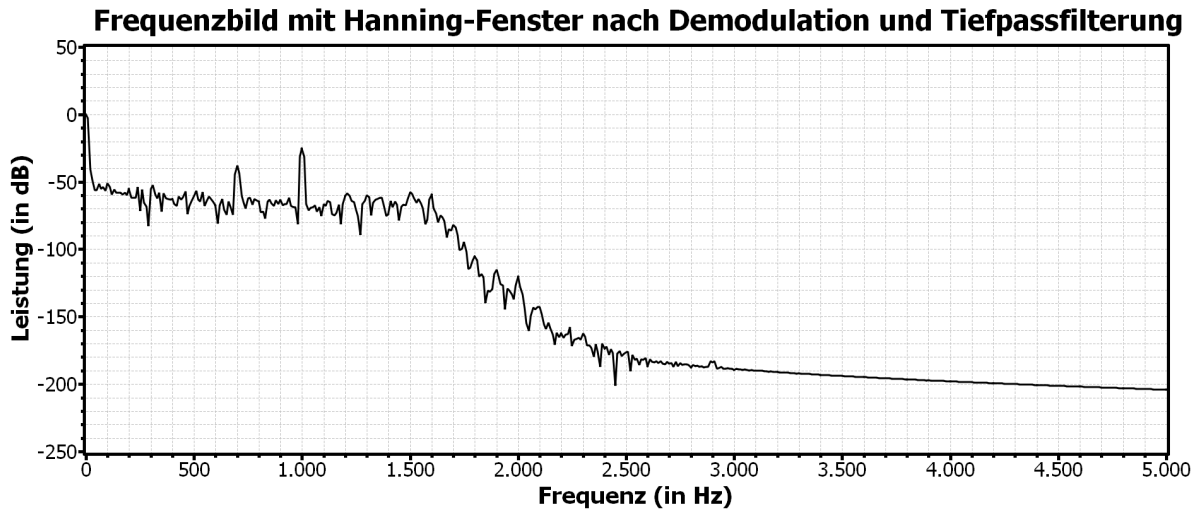


Abbildung 39: In dieser Abbildung ist das Ergebnis der Demodulation und Tiefpassfilterung eines PM-Signals mit  $k = 0,1$  zu sehen. Dieses Frequenzbild mit Hanning-Fenster ist das Äquivalent zu dem aus Abb. 36 bekannten, sich rechts unten befindenden Frontpanel-Diagramm. Die Daten für das hier dargestellte Frequenzspektrum entspringen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen txt-Datei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet. Die verwendete AM-Demodulation findet mittels „Quadrat“-Methode statt.

Da sich die beiden in diesem Frequenzspektrum deutlich sichtbaren Peaks bei 700 Hz und 1.000 Hz befinden und diese genau den Frequenzen  $f_1$  und  $f_2$  des Signals  $s(t)$  entsprechen, kann man von einer erfolgreichen Demodulation des gemessenen, phasenmodulierten Signals sprechen.

### 6.3 Erweiterte Demodulation mit Bandpass und zusätzlicher Integration des Signals

Da es sich hierbei um eine Programm- und Demodulationserweiterung handelt, bildet der in Abschnitt 6.2 entstandene Programmcode die Basis für das folgende LabVIEW-Programm. Dessen Blockdiagramm ist in Abb. 41 und Abb. 42 dargestellt.

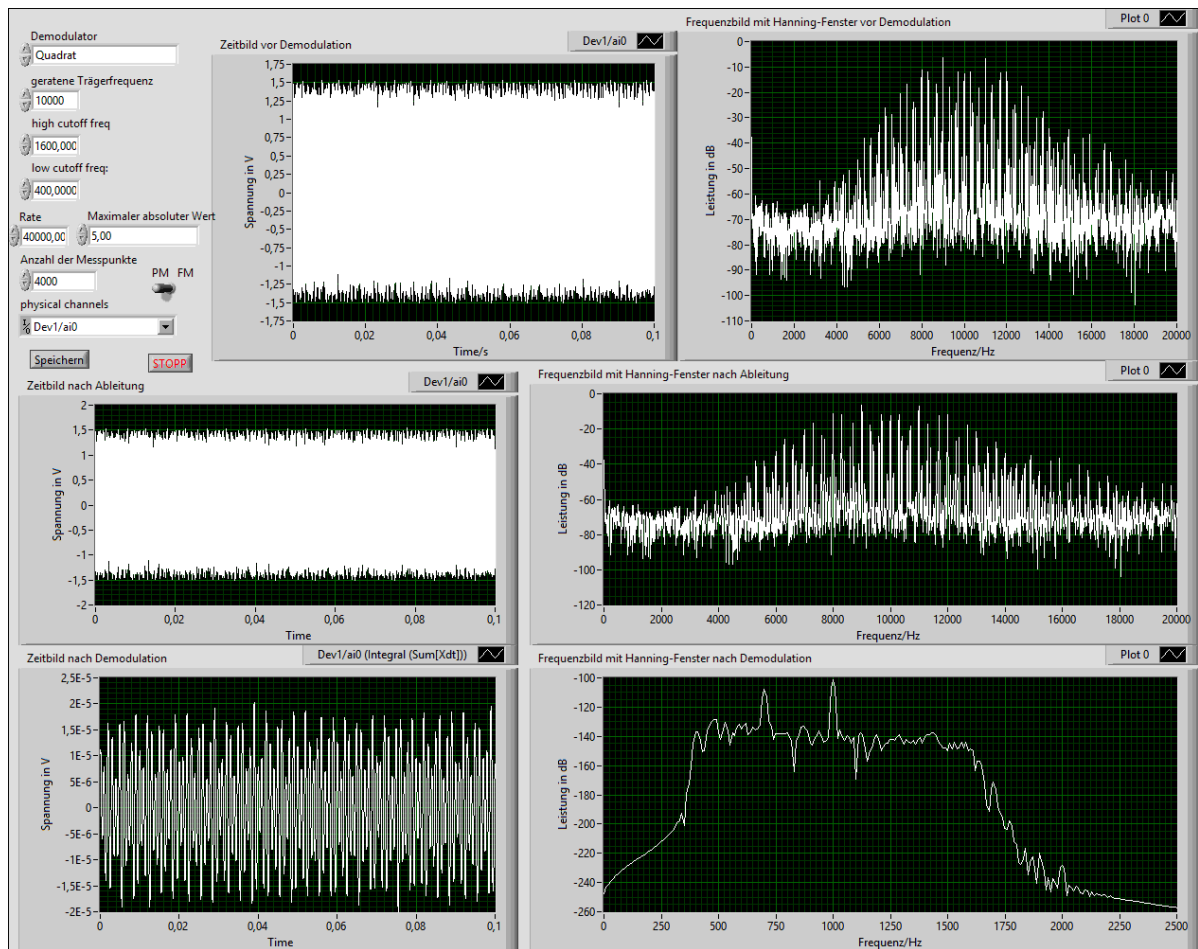


Abbildung 40: In dieser Abbildung ist die LabVIEW-Benutzeroberfläche bzw. das LabVIEW-Frontpanel des Programms zur erweiterten Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, phasenmodulierten Signals („PM“) dargestellt. Die AM-Demodulation findet hierbei mittels „Quadrat“-Methode statt.

Abb. 40 zeigt das dazugehörige Frontpanel.

Zunächst wird der der AM-Demodulation nachstehende Tiefpassfilter durch einen Bandpassfilter sechzehnter Ordnung ersetzt, was unter anderem im Programmcodeteil in Abb. 41 zu sehen ist. Die obere und untere Grenzfrequenz lässt sich dabei durch entsprechende auf dem Frontpanel hinzugefügte Bedienelemente einstellen. Die untere Grenzfrequenz soll 400 Hz und die obere Grenzfrequenz 1.600 Hz betragen.

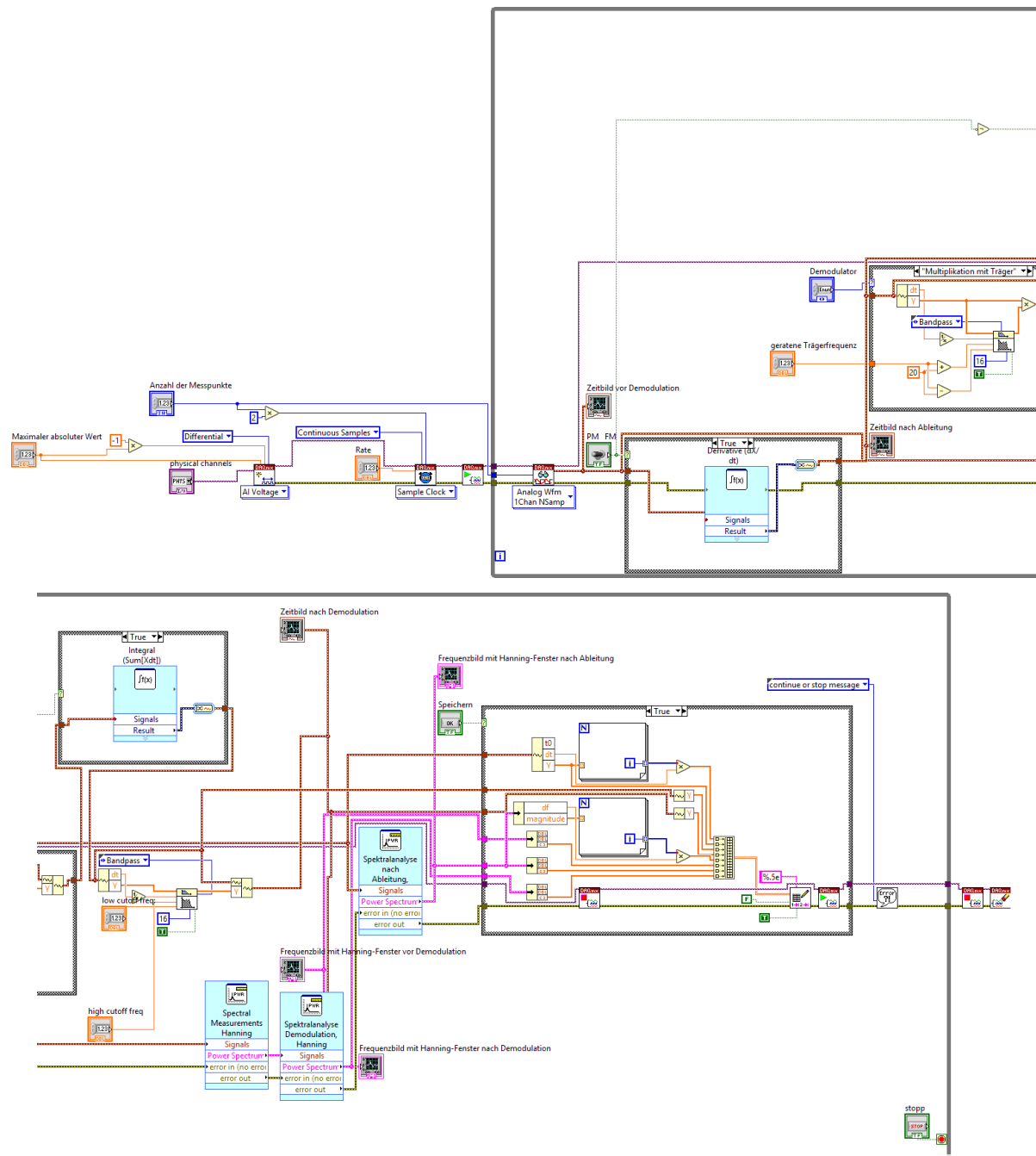


Abbildung 41: Die Abbildung veranschaulicht das LabVIEW-Blockdiagramm bzw. den Programmcode zur erweiterten Demodulation eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen, phasen- oder frequenzmodulierten Signals.

Im Programmcode wird zwischen dem für die Demodulation von AM-Signalen zuständigen Programmbereich und dem Bandpassfilter eine Case-Konstruktion ergänzt, welche insgesamt zwei Case-Optionen umfasst. Dadurch entstehen zwei zunächst freie,

ausfüllbare Case-Felder, wobei das Erste für demodulierte FM-Signale und das Zweite für demodulierte PM-Signale vorgesehen ist. Über eine Leitung mit NOT-Gate und in Verbindung mit dem bereits aus Abschnitt 6.2 bekannten Kippschalter lässt sich die Case-Konstruktion ansteuern, sodass beim Weiterverfahren mit dem demodulierten Signal zwischen ursprünglich frequenzmodulierten Signalen und ursprünglich phasenmodulierten Signalen unterschieden werden kann.

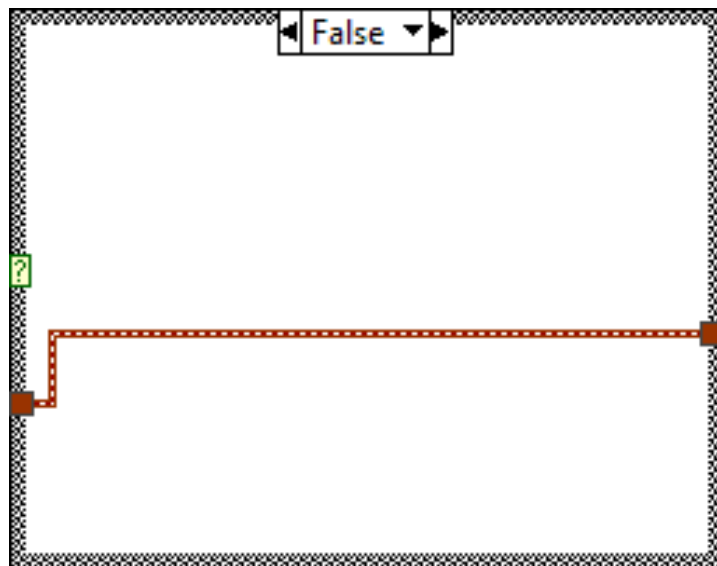


Abbildung 42: In dieser Abbildung ist der in Abb. 41 nicht sichtbare Teil des LabVIEW-Programmcodes zu sehen. Dieser Programmcodeteil bildet die zweite Option in der Case-Konstruktion und sorgt dafür, dass das demodulierte FM-Signal vor der Bandpassfilterung nicht verändert wird.

Wie im Programmcodeteil in Abb. 42 zu erkennen ist, wird die Leitung direkt durch das für demodulierte FM-Signale reservierte Case-Feld gezogen, sodass das übertragene Signal unverändert bleibt. Innerhalb des für demodulierte PM-Signale reservierten Case-Feldes wird das Signal hingegen mit Hilfe der Simpsonregel numerisch integriert und anschließend an den Bandpassfilter weitergegeben. Dies ist im Programmcode in Abb. 41 zu sehen. Auf diese Weise verschwinden die Gleichanteile und die Frequenzabhängigkeit des demodulierten PM-Signals.

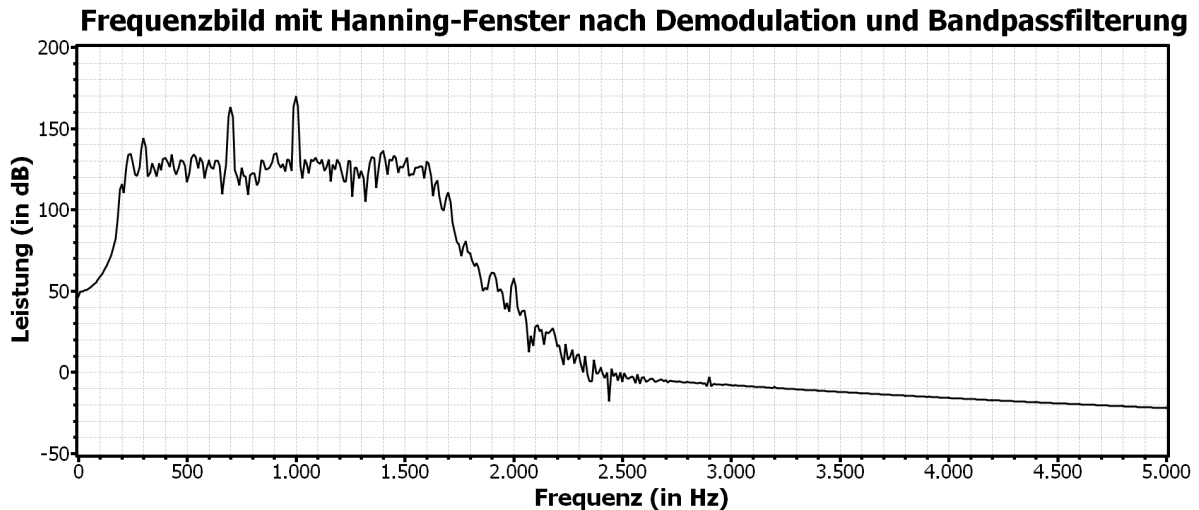


Abbildung 43: In dieser Abbildung ist das Ergebnis der Demodulation und Bandpassfilterung eines FM-Signals zu sehen. Dieses Frequenzbild mit Hanning-Fenster ist das Äquivalent zu dem aus Abb. 40 bekannten, sich rechts unten befindenden Frontpanel-Diagramm. Die Daten für das hier dargestellte Frequenzspektrum entspringen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen txt-Datei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet. Die verwendete AM-Demodulation findet mittels „Quadrat“-Methode statt.

Das Ergebnis der erweiterten Demodulation sowie Bandpassfilterung eines von der Computer-Soundkarte ausgegebenen und anschließend über den Analog-Digital-Wandler gemessenen FM- bzw. PM-Signals ist in Abb. 43 bzw. Abb. 43 als Diagramm dargestellt. In diesen beiden Frequenzspektren lassen sich zwei deutliche Peaks bei 700 Hz und 1.000 Hz ausmachen. Diese entsprechen genau den Frequenzen  $f_1$  und  $f_2$  des zu modulierenden Signals  $s(t)$ , weshalb man von einer erfolgreich umgesetzten, erweiterten Demodulation und Bandpassfilterung des gemessenen, frequenzmodulierten bzw. phasenmodulierten Signals sprechen kann.

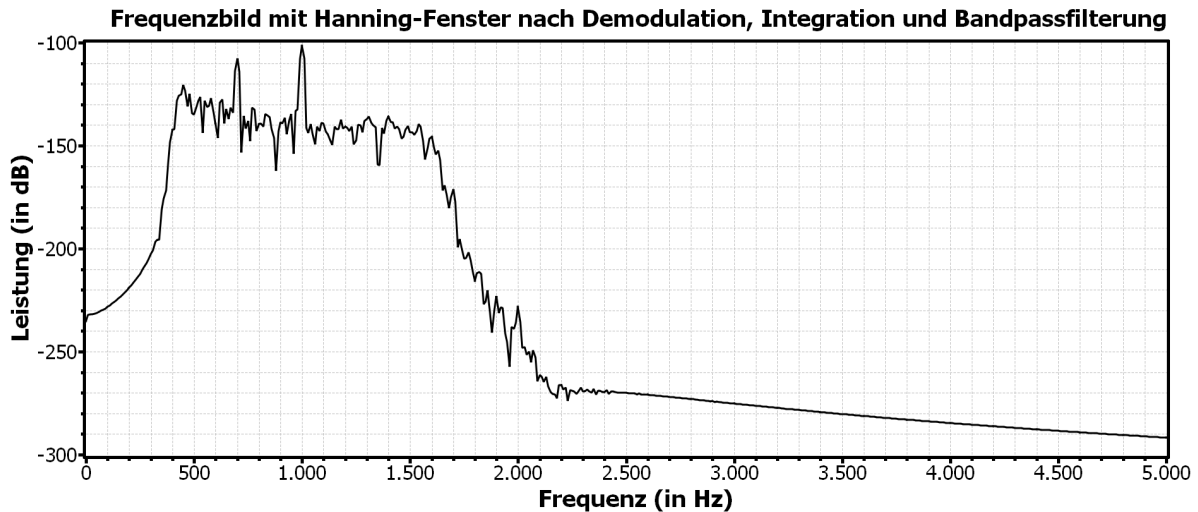


Abbildung 44: Die Abbildung zeigt das Frequenzspektrum eines PM-Signals nach Abschluss der Demodulation, Integration und Bandpassfilterung. Bei diesem Frequenzbild mit Hanning-Fenster handelt es sich um das Äquivalent zu dem aus Abb. 40 bekannten, sich rechts unten befindenden Frontpanel-Diagramm. Die Daten für das hier dargestellte Frequenzspektrum entspringen der durch die Speicherfunktion des LabVIEW-Programms erhaltenen txt-Datei und wurden mit dem Datenanalyse-Programm „SciDAVis“ bearbeitet. Die verwendete AM-Demodulation findet mittels „Quadrat“-Methode statt.

*Hinweis:* Sobald  $k$  innerhalb eines bestimmten Wertebereichs liegt, werden die sich nach der erweiterten Demodulation und Bandpassfilterung eines FM- bzw. PM-Signals ergebenden Frequenzspektren anschaulicher und die Peaks sind deutlicher zu erkennen. Ist der  $k$ -Wert jedoch zu hoch, kommen zu viele Oberwellen zustande. Wenn der  $k$ -Wert wiederum zu klein gewählt wird, verschwindet das demodulierte Signal im Hintergrundrauschen.

## Literatur

- [1] wdwd CC BY 3.0. *Spectral leakage Sine*. URL: <https://commons.wikimedia.org/w/index.php?curid=16996723> (besucht am 17.09.2018).
- [2] Public Domain. *Aliased Spectrum*. URL: <https://commons.wikimedia.org/w/index.php?curid=1062676> (besucht am 17.09.2018).