

R. Brigola, TH Nürnberg Georg Simon Ohm, April 2018

Mathematica - Notebooks als Bonusmaterial zum Lehrbuch

**[1] Rolf Brigola Fourier-Analysis und Distributionen,
Eine Einführung mit Anwendungen,
edition swk, Hamburg 2018 (in Vorbereitung)**

**Demonstrationsbeispiele zum OFDM-Übertragungsverfahren
(Orthogonal Frequency Divison Multiplexing) mit *Mathematica***

Referenzen zu Kapiteln, Abschnitten, Seitenzahlen beziehen sich im Folgenden meist auf das genannte Lehrbuch des Autors. Einige wenige weitere Referenzen findet man am Ende des Notebooks. Hier verwendete Mathematica-Version: *Mathematica* 12.0, später auch 12.1. Die URL aller meiner *Mathematica*-Notebooks zur Fourier-Analysis ist: <http://www.stiftung-swk.de/mathematica/>

In diesem Notebook wird versucht, das Prinzip des Mehrträger-Übertragungsverfahrens OFDM verständlich zu machen. **Referenzen sind [2]-[6] am Notebook-Ende .**
OFDM-Übertragungsverfahren werden heute von nahezu jedem täglich verwendet.
Sie sind z.B. im Einsatz bei WLAN, DSL, LTE, DAB, DRM, DVB-T u.a.m.

Zur Erklärung beginne ich mit einem **Analog-Modell** mit einigen wenigen komplexen Trägerfunktionen, so dass mit grundlegender Kenntnis über Fourierreihen und die diskrete Fouriertransformation (DFT) schon klar wird, wie ein OFDM-Signal und sein Spektrum aussehen. Dabei werden harmonische Schwingungen als Träger verwendet, die in ihren Amplituden und Phasen die Nutzinformation transportieren können.

Im Anschluss konstruieren wir mit *Mathematica* eine “**diskrete Variante**” zur Aufbereitung eines “Sende-Signals” mit Hilfe der diskreten Fouriertransformation.

Der Vorteil von Mathematica gegenüber einer Darstellung mit einem System zur Nachbildung, das allein Sampling-Werte benutzt, ist aus meiner Sicht die Möglichkeit, in Teilen mit kontinuierlichen Funktionen zu arbeiten und damit “realistischer” als in rein diskreten Modellen/Simulationen. Außerdem kann man damit die “*Fourier-Analysis-Maschine*” rekapitulieren und sein Wissen darüber festigen -- so hoffe ich jedenfalls.

1. Komplexe Amplituden, die durch OFDM übertragen werden

Daten, die digital übertragen werden sollen, bestehen aus einem Strom von Bits, m.a.W. aus einer 0-1-Folge. Das können Daten für einen Verbindungsaufbau, Inhalte von Web-Seiten, Bild-, Video- oder Audio-Daten sein u.v.a.m. Der Datenstrom wird in eine 0-1-Folge gewandelt und als Vorstufe vor einer Übertragung *codiert* (*fehlerkorrigierender Code*, *Interleaving* seien als Stichworte genannt). Die resultierende Bitfolge wird für eine Übertragung mittels OFDM dann z.B. mit 16QAM (oder andern Modulationen) auf komplexe Zahlen abgebildet. Dies ist das Verfahren, nach dem hierzulande DVB-T u.a. ähnliche Mehrträgerverfahren arbeiten.

Wenn Sie einen DVB-T-Receiver haben, sehen Sie bitte einmal die Informationen zu einem Programm an. Da steht z.B. zu lesen:

Kanal	34
Frequenz	578 MHz
FFT	8 k
Modulation	16QAM
GI	1/4

Wir wollen verstehen lernen, was die Angaben bedeuten: Der DVB-T-Kanal hat die Nr. 34, sendet in einem geeignet breiten Frequenzband um 578 MHz, verwendet ein OFDM-Verfahren auf der Grundlage der DFT/FFT im 8k-Mode mit 8192 Abtastwerten, mit einem Guard-Intervall ("Schutz-Intervall") von 1/4 der OFDM-Symboldauer und verwendet die 16QAM-Modulation. Diese soll zuerst erklärt werden, der Rest danach. Auf andere Modulationsarten, die in der Praxis verwendet werden - wie BPSK, QPSK - gehe ich hier nicht ein.

Die 16QAM-Modulation

Bei 16QAM wird der serielle 0-1-Bitstrom parallelisiert (Seriell-Parallel-Umsetzung oder kurz S/P-Umsetzung), indem für eine gewisse Anzahl M von vorgesehenen Trägern pro Zeitschritt und Träger des Übertragungsverfahrens **je 4 Bit auf komplexe Zahlen abgebildet werden**, deren Gesamtheit als **komplexes Alphabet** bezeichnet werden kann. Die Elemente dieses Alphabets nenne ich **komplexe Amplituden**. 4 Bit pro komplexer Amplitude ergeben in allen Kombinationen 16 Möglichkeiten, daher der Name **16QAM** für diese **Quadratur-**

amplitudenmodulation. Diese Abbildung einmal als Liste, dann grafisch:

$$\text{In[1]= } A = 1 / \sqrt{10} ;$$

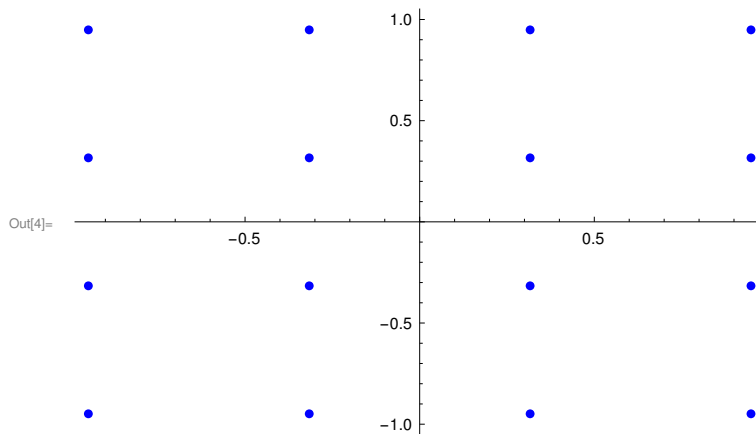
```
alphabet = {"0101", A (1 + I)}, {"0001", A (1 + 3 I)}, {"0000", A (3 + 3 I)},
  {"0100", A (3 + I)}, {"1101", A (1 - I)}, {"1001", A (1 - 3 I)}, {"1100", A (3 - I)},
  {"1000", A (3 - 3 I)}, {"0111", A (-1 + I)}, {"0011", A (-1 + 3 I)}, {"0010", A (-3 + 3 I)},
  {"0110", A (-3 + I)}, {"1111", A (-1 - I)}, {"1011", A (-1 - 3 I)},
  {"1110", A (-3 - I)}, {"1010", A (-3 - 3 I)} (* Symboltabelle 16QAM *)
```

$$\text{Out[2]= } \left\{ \left\{ 0101, \frac{1+i}{\sqrt{10}} \right\}, \left\{ 0001, \frac{1+3i}{\sqrt{10}} \right\}, \left\{ 0000, \frac{3+3i}{\sqrt{10}} \right\}, \left\{ 0100, \frac{3+i}{\sqrt{10}} \right\}, \right. \\ \left. \left\{ 1101, \frac{1-i}{\sqrt{10}} \right\}, \left\{ 1001, \frac{1-3i}{\sqrt{10}} \right\}, \left\{ 1100, \frac{3-i}{\sqrt{10}} \right\}, \left\{ 1000, \frac{3-3i}{\sqrt{10}} \right\}, \right. \\ \left. \left\{ 0111, -\frac{1-i}{\sqrt{10}} \right\}, \left\{ 0011, -\frac{1-3i}{\sqrt{10}} \right\}, \left\{ 0010, -\frac{3-3i}{\sqrt{10}} \right\}, \left\{ 0110, -\frac{3-i}{\sqrt{10}} \right\}, \right. \\ \left. \left\{ 1111, -\frac{1+i}{\sqrt{10}} \right\}, \left\{ 1011, -\frac{1+3i}{\sqrt{10}} \right\}, \left\{ 1110, -\frac{3+i}{\sqrt{10}} \right\}, \left\{ 1010, -\frac{3+3i}{\sqrt{10}} \right\} \right\}$$

Hier die Lage der komplexen Amplituden, die später mit OFDM übertragen werden, in der komplexen Ebene. Sie werden später mit der Tabelle beim Empfänger der Übertragung wieder in die zugehörigen 4-Bit-Zahlen transformiert und mit P/S-Umsetzung zu einem seriellen Bitstrom zusammengefügt. **Zusätzlich nehmen wir noch Null als Amplitude für später unbesetzte Träger hinzu (hier nicht in der Liste und nicht gezeichnet).**

```
In[3]:= camp = Table[alphabet[[n, 2]], {n, 1, 16}] (* die komplexen Symbole,
die 16QAM übertragen werden können *)
ListPlot[(Tooltip[{Re[#1], Im[#1]}] &)/@ camp,
PlotStyle -> Directive[Blue, Thickness[0.7]], ImageSize -> Medium]
(* und ihre Lage in der komplexen Ebene; bitte AspectRatio beachten *)
```

$$\text{Out[3]= } \left\{ \frac{1+i}{\sqrt{10}}, \frac{1+3i}{\sqrt{10}}, \frac{3+3i}{\sqrt{10}}, \frac{3+i}{\sqrt{10}}, \frac{1-i}{\sqrt{10}}, \frac{1-3i}{\sqrt{10}}, \frac{3-i}{\sqrt{10}}, \frac{3-3i}{\sqrt{10}}, \right. \\ \left. -\frac{1-i}{\sqrt{10}}, -\frac{1-3i}{\sqrt{10}}, -\frac{3-3i}{\sqrt{10}}, -\frac{3-i}{\sqrt{10}}, -\frac{1+i}{\sqrt{10}}, -\frac{1+3i}{\sqrt{10}}, -\frac{3+i}{\sqrt{10}}, -\frac{3+3i}{\sqrt{10}} \right\}$$



Bemerkung: Der Faktor $1/\sqrt{10}$ ist so gewählt, dass bei einer Zufallsfolge von Bits in einem stochastischen Signal die normalisierte Leistung beim 16QAM auf Eins normiert ist. Ich lasse den Faktor nachfolgend mehrfach weg, damit man Zahlenergebnisse besser vergleichen kann.

2. Fourierreihen und die diskrete Fouriertransformation DFT (als FFT programmiert natürlich) als Grundlage der Übertragung mit OFDM

Ein einfaches Analog-Modell mit komplexen Trägerfunktionen

OFDM soll zunächst an einem **Analog-Modell mit komplexen Trägerfunktionen** erklärt werden. **Träger sind in der Folge harmonische Schwingungen, die in ihrer Amplitude und Phase 2 Informationen "tragen können", daher der Name.**

Das Sendesignal hat für jede gleichzeitig gesendete 4-Bit-Sequenz/Amplitudengruppe eine gewisse Zeitdauer, kurz **Symboldauer** T_s . Wir schneiden aus einer auf ganz \mathbb{R} definierten

Funktion einen Abschnitt der Dauer T_s heraus, indem wir diese Funktion mit der “fenster” genannten **Rechteckfunktion** multiplizieren. Wir betrachten nur den Zeitausschnitt $[0, T_s[$.

Als **Träger** verwenden wir dann für M komplexe Amplituden/Buchstaben c_k des 16QAM-Alphabets (mit $c_0=0$), die wir in der Zeitdauer T_s gemeinsam übertragen wollen, **die Funktionen** $e^{i 2 \pi f[k] t}$ für $k=0, \dots, M-1$, mit $f[k]=k/T_s$ (o.E.d.A. M gerade).

Da M Amplituden gleichzeitig übertragen werden und **die Funktionen** $e^{i 2 \pi f[k] t}$ **fenster(t)** ein orthogonales Funktionensystem in $L^2([0, T_s[)$ bilden, die sich das Frequenzband von 0 bis $f[M-1]$ teilen, hat man das Verfahren **OFDM** getauft (Orthogonal Frequency Division Multiplexing), siehe [2].

Um ein einfaches Demo-Beispiel zu sehen, wählen wir also T_s , damit den Trägerabstand (Frequenzabstand) $TrAbstand = 1/T_s$, und das Zeitfenster als Rechteckfunktion. Außerdem wählen wir die Trägeranzahl $M=5$ für das folgende Beispiel und (willkürlich) gewisse komplexe Amplituden als “ampauswahl”, dazu die Trägerfrequenzen fr und die Bandbreite des benötigten Frequenzbandes. Die Zuordnung der Trägerfrequenzen zu den komplexen Amplituden nachfolgend ist $fr[k] \leftrightarrow ampauswahl[k]$.

$$\sum_{n=0}^{M-1} c_n e^{i 2 \pi n / T_s t} = \sum_{k=0}^{M-1} ampauswahl[k+1] e^{i 2 \pi fr[k+1] t}.$$

Konvention: $c_0=0$, damit das Signal keinen Gleichanteil besitzt.

Beispiel 1

```
In[5]:= Ts = 1/2; (* im Beispiel: Zeitdauer Kernsymbol *)
TrAbstand = 1/Ts; (* Trägerabstand *)
fenster[t_] = (HeavisideTheta[t] - HeavisideTheta[t - Ts]);
Fenster[t_] = (UnitStep[t] - UnitStep[t - Ts]);
(* Pulsform HeavisideTheta[t] - HeavisideTheta[t - Ts]; *)
M = 5; (* Trägeranzahl inkl. Gleichanteil, den wir Null setzen *)
ampauswahl = A Table[{0, (1 + 3 I), (3 + I), (-3 + 3 I), (-1 - 1 I)}]
(* Flatten[Prepend[Table[camp[[k+3]], {k, 1, 4}], 0]]
(* eine Auswahl von 5 komplexen Amplituden,
die zur Frequenz Null ist ebenso Null: Dieser Träger wird nicht moduliert *)*)
fr = Table[k TrAbstand, {k, 0, M - 1}] (* die M Frequenzen, die dann benutzt werden *)
```

$$\text{Out}[8]= \left\{ 0, \frac{1+3i}{\sqrt{10}}, \frac{3+i}{\sqrt{10}}, -\frac{3-3i}{\sqrt{10}}, -\frac{1+i}{\sqrt{10}} \right\}$$

$$\text{Out}[9]= \{0, 2, 4, 6, 8\}$$

1) Übertragung einer einzigen komplexen Amplitude:

Zur Veranschaulichung sehen wir zuerst, wie die Verhältnisse bei der Übertragung einer einzigen komplexen Amplitude mit einem einzigen Träger wäre. Danach wird das einfach auf den Fall mit mehr Amplituden und Trägern durch Superposition fortgesetzt:

4 Bit der Nutzdaten könnten z.B. die Bits "1000" sein. Sie werden auf die komplexe Amplitude $\frac{3-3i}{\sqrt{10}}$ abgebildet. Ich wähle daher das 3. Element dieser Liste und **die komplexe Exponentialfunktion mit der Frequenz $fr[3]$** als Träger, multipliziert mit der Fensterfunktion wie oben gesagt. Wir wissen, dass das Spektrum der Funktion dem des Spektrums der Fensterfunktion ähnlich ist: dieses wird mit der komplexen Amplitude $\frac{3-3i}{\sqrt{10}}$ multipliziert und um $fr[3] = 4$ Hz nach rechts verschoben (vgl. [1], o.a. Literatur zur Fourier-Analyse).

Hier die Bilder der Absolutbeträge der Fouriertransformierten, a) der Fensterfunktion, b) der Modulation: *Man erkennt die paarweise Orthogonalität der gefensterten Träger auch daran, dass das Spektrum eines einzelnen Trägers dort Nullstellen besitzt, wo die Beträge der anderen Träger ihre Maxima besitzen, d.h. eine Verfälschung der Amplitude eines Trägers hat keinen Einfluss auf die Maximalbeträge der Amplituden anderer Träger, es gibt also im Modell keine Interchannel-Interferenz (ICI).*

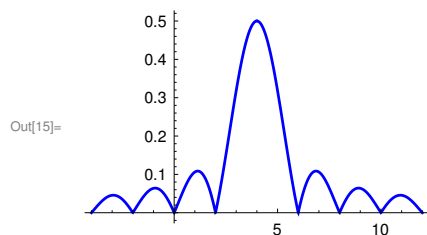
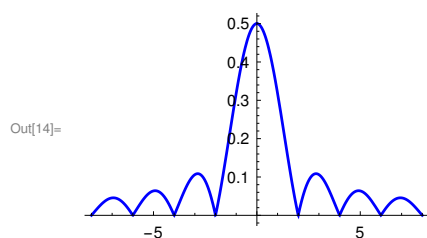
*Man erkennt außerdem, dass durch die Dauer des Zeitfensters die Amplitude im Spektrum beeinflusst wird, entsprechend der Tatsache, dass $|\text{Spektrum}|^2$ physikalisch eine **Energiedichte** des Signals ist, die mit der Dauer des Signals variiert.*

```

In[10]:= specpuls[w_] = FourierTransform[ fenster[t], t, w, FourierParameters -> {1, -1}];
ft1[w_] = FourierTransform[
  ampauswahl[[3]] Exp[I 2 Pi fr[[3]] t] fenster[t], t, w, FourierParameters -> {1, -1}];
spec1abs[w_] = Abs[ft1[w]];
fr[[3]] (* Frequenz, um die verschoben wird *)
Plot[Abs[specpuls[2 Pi f]], {f, -8, 8},
  PlotStyle -> Directive[Blue, Thickness[0.008]], ImageSize -> Small] (* zur fenster *)
Plot[spec1abs[2 Pi f], {f, -4, 12}, PlotStyle -> Directive[Blue, Thickness[0.008]],
  ImageSize -> Small] (* zur Modulation *)

```

Out[13]= 4



2) Mehrträger - Übertragung im Analog-Modell mit komplexen Trägerfunktionen

Bei der Übertragung mehrerer Bitgruppen/komplexer Amplituden gleichzeitig entsteht durch Superposition der einzelnen Träger eine (endliche) **Ts-periodische Fourierreihe**, zeitbegrenzt durch das Zeitfenster. Die beteiligten Funktionen bilden ein orthogonales Funktionensystem im Hilbertraum $L^2([0, T_s])$, wie aus der Theorie zu Fourierreihen wohl bekannt ist. Dass die Träger orthogonal sind, hat - als Grund zur Namensgebung für das Verfahren - zur Folge, dass Amplitudenänderungen bei einem Träger sich im Gesamtsignal nicht auf die Komponenten der anderen Träger auswirken. Anders gesagt, können dadurch **Intersymbol-Interferenz (ISI)** und **Inter-Carrier-Interferenz (ICI)** minimiert werden - ein großer Vorteil des Verfahrens (ISI und ICI sind zu einem gewissen Grad bei Mehrwegeausbreitung und Übertragung auf einem bandbegrenzten Kanal unvermeidbar.)

Nachfolgend nun die Superposition mit den oben gewählten 4 besetzten Trägern und das zugehörige Betragsspektrum.

```

In[16]:= trigpol[t_] = Sum[ampauswahl [[k + 1]] Exp[I 2 Pi fr[[k + 1]] t], {k, 0, M - 1}]
signal[t_] = trigpol[t] * fenster[t];
specanalog[w_] =
  N[Sum[ampauswahl [[k + 1]] * FourierTransform [Exp[I 2 Pi fr[[k + 1]] t] fenster[t],
    t, w, FourierParameters -> {1, -1}], {k, 0, M - 1}]];
absspecanalog[w_] = N[Abs[specanalog[w]]];
Plot[absspecanalog [2 Pi f], {f, -20, 40}, PlotRange -> {0, 0.8},
  PlotStyle -> Directive[Blue, Thickness[0.008]], ImageSize -> Medium]

```

$$\text{Out[16]= } \frac{(1 + 3i) e^{4i\pi t}}{\sqrt{10}} + \frac{(3 + i) e^{8i\pi t}}{\sqrt{10}} - \frac{(3 - 3i) e^{12i\pi t}}{\sqrt{10}} - \frac{(1 + i) e^{16i\pi t}}{\sqrt{10}}$$

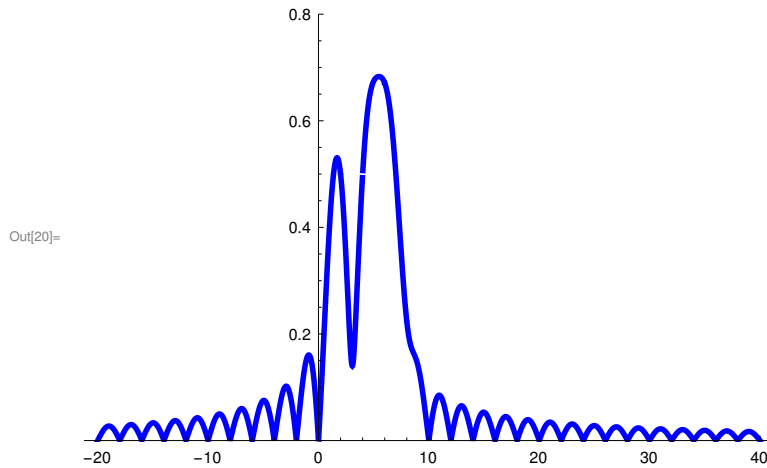


Bild 1 Betragsspektrum zu Beispiel 1

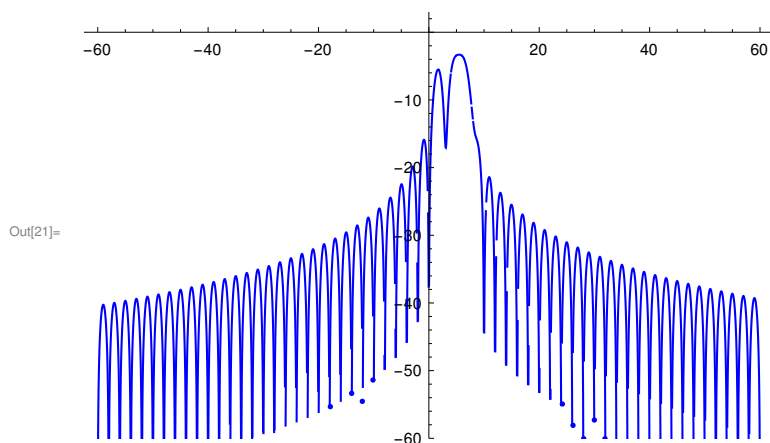
Hier noch das Amplitudenspektrum in dB. Es fällt sofort auf, dass es für wachsende Frequenzbeträge sehr langsam abfällt (wie $\sin(x)/x$ für $|x| \rightarrow \infty$).

Die “**Side Lobes**” über die Signal-Bandbreite hinaus sind eine Folge der Zeitbegrenzung mit dem Rechteck-Fenster (zeitbegrenzte Signale sind nicht bandbegrenzt). Diese “Außerbandstrahlung” ist in der Praxis höchst unerwünscht, da sie im Nachbarbereich als Störung wirkt. Im Interesse einer Frequenzbändernutzung durch möglichst viele Sender (Frequenzbänder werden ja für hohe Beträge vom Staat versteigert) muss man daher in der Praxis mit anderen als Rechteckfenstern arbeiten. Ein probates Gegenmittel gegen zuviel Außerbandstrahlung ist der Gebrauch von Zeitfenstern, die an den Ecken glatter gerundet sind (man vgl. hierzu [1], Zusammenhänge zwischen Glattheitseigenschaften von Funktionen und dem Abfallen ihres Amplitudenspektrums).

Bild 2 Betragsspektrum zu Beispiel 1 in dB

Die dB-Angaben beziehen sich auf eine Referenzamplitude 1 V.


```
In[21]:= Plot[20 Log[10, absSpecanalog [2 Pi f]], {f, -60, 60}, PlotRange -> {-60, 3},
PlotStyle -> Directive[Blue, Thickness[0.003]], ImageSize -> Medium]
```



3) Rückgewinnung der komplexen Amplituden

Abtastung des Signals zu den Zeiten $t_n = n \cdot T_s / M$ ergibt eine Wertefolge y_n , die gerade eine IDFT der ausgewählten komplexen 16QAM-Amplituden ist.

Eine DFT der y_n liefert dann natürlich die Amplitudenauswahl zurück.

Ich multipliziere mit $\sqrt{10}$, damit die Amplituden besser mit der nachgestellten Auswahl verglichen werden können.

Zuerst die berechneten Amplituden, dann zum Vergleich die gesendeten im komplexen Signal. Wir erkennen Gleichheit. Die Gesamtheit aller in einem Zeitabschnitt der Länge T_s übermittelten Bitgruppen/komplexen Amplituden nennt man ein **OFDM-Symbol**.

```
In[22]:= y = Table[trigpol[n Ts / M], {n, 0, M - 1}]; (* Abtastwerte *)
ergebnisamp = Chop[N[Sqrt[10] Fourier[y, FourierParameters -> {-1, -1}]]]
(* Sqrt[10] DFT der y[k] *)
amplituden = 1/A ampauswahl; (* Durch Abtastung und DFT berechnete Amplituden *)
amplituden (* ursprüngliche Auswahl
komplexer Amplituden zum Vergleich ohne den Faktor A *)
```

```
Out[23]= {0, 1. + 3. i, 3. + 1. i, -3. + 3. i, -1. - 1. i}
```

```
Out[25]= {0, 1 + 3 i, 3 + i, -3 + 3 i, -1 - i}
```

Damit haben wir die Grundidee der Übertragung und Rückgewinnung von 16QAM-Amplituden schon kennengelernt. Aus den durch eine DFT von Abtastwerten errechneten komplexen Amplituden wird dann die ursprüngliche eigentlich interessierende 0-1-Bitfolge rekonstruiert. (Vorausgesetzt dabei: Identisches Signal beim Sender und Empfänger. Störeinflüsse des Übertragungskanals müssen vor einer Zuordnung von DFT-Werten zu 16QAM-Amplituden beim Empfänger geschätzt und

korrigiert werden.)

3. Von komplexen Amplituden und Funktionen zu reellen Signalen: Quadraturamplitudenmodulation

3.1 Modulation

Um statt komplexer Amplituden und Funktionen reelle Amplituden (und Phasen) und reelle Trägerfunktionen zu verwenden, verwendet man die sog. **Quadraturamplitudenmodulation (QAM)**, den zweiten Teil des Namens von **16QAM**, oder auch kürzer **Quadraturmodulation**.

Statt eines komplexen OFDM-Signals mit "Zwischenfrequenz" f_c

$$(C) \quad s_{c,m}(t) = e^{j2\pi f_c t} \sum_{k=0}^{M-1} c_k e^{j2\pi f_k t} \cdot \text{fenster}(t-mTs)$$

kann man im m-ten Zeitintervall den **Realteil als reelles Signal** bilden:

$$\begin{aligned} (I) \quad y_m(t) &= \text{Re} \left(e^{j2\pi f_c t} \sum_{k=0}^{M-1} c_k e^{j2\pi f_k t} \text{fenster}(t-mTs) \right) \\ &= \cos(2\pi f_c t) \sum_{k=0}^{M-1} (\text{Re}(c_k) \cos(2\pi f_k t) - \text{Im}(c_k) \sin(2\pi f_k t)) \text{fenster}(t-mTs) \\ &\quad - \sin(2\pi f_c t) \sum_{k=0}^{M-1} (\text{Re}(c_k) \sin(2\pi f_k t) + \text{Im}(c_k) \cos(2\pi f_k t)) \text{fenster}(t-mTs) \\ &= (I(t) \cos(2\pi f_c t) - Q(t) \sin(2\pi f_c t)) \text{fenster}(t-mTs). \end{aligned}$$

Wir setzen wie vorher $m=0$, lassen den Index m weg und betrachten nur einen Zeitabschnitt der Dauer T_s . Real- und Imaginärteile der 16QAM-Amplituden c_k und damit die gesamte Information werden mit **wiederum sämtlich orthogonalen reellen T_s -periodischen Funktionen $\cos(2\pi f_k t)\text{fenster}(t)$, $\sin(2\pi f_k t)\text{fenster}(t)$** ($k=0, \dots, M-1$) als Träger moduliert. **Die Funktion $I(t)$ heißt Inphase-Komponente und die Funktion $Q(t)$ Quadratur-Komponente**. $I(t)$ ist der Realteil von $e^{-j2\pi f_c t} s_c(t)$, $Q(t)$ der Imaginärteil (vgl. etwa auch [7]).

Beispiel fortgeführt: f_c = Zwischenfrequenz der Quadratur-Modulation

```
In[26]:= fc = 32;
wc = 2  $\pi$  fc ;
Inphase[t_] = ComplexExpand [Re[trigpol[t]]]; (* Inphase-Komponente *)
Quadratur[t_] = ComplexExpand [Im[trigpol[t]]]; (* Quadratur-Komponente *)
ofdmmod[t_] = Inphase[t] Cos[wc t] - Quadratur[t] Sin[wc t];
(* Quadratur-Modulation mit Zwischenfrequenz *)
ofdmchip[t_] = ofdmmod[t] * fenster[t]; (* Modulation mit Zeitfenster der Dauer Ts *)
(* reelles OFDM-Signal von t=0 bis t=Ts *)
```

Bild 3 Das resultierende reelle Zeitsignal mit Zwischenfrequenz f_c

```
In[32]:= Plot[ofdmchip[t], {t, -0.1, 0.6}, PlotRange -> All,
PlotStyle -> Directive[Blue, Thickness[0.007]]]
```

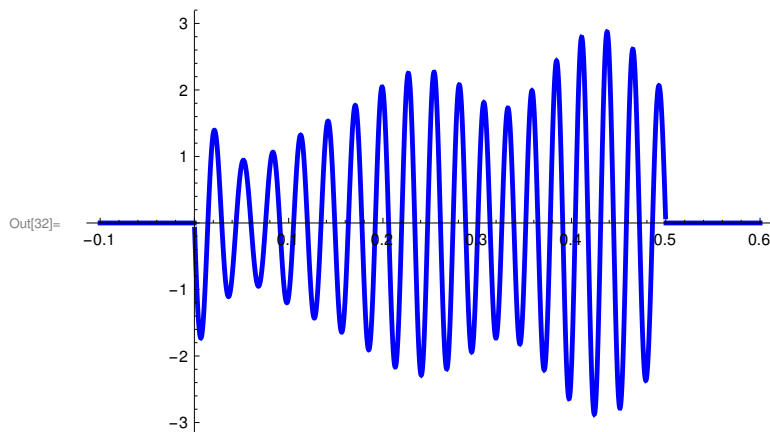


Bild 4 Das halbseitige Amplitudenspektrum des nun reellen Signals mit gleicher Bandbreite wie das Basisband-Signal :

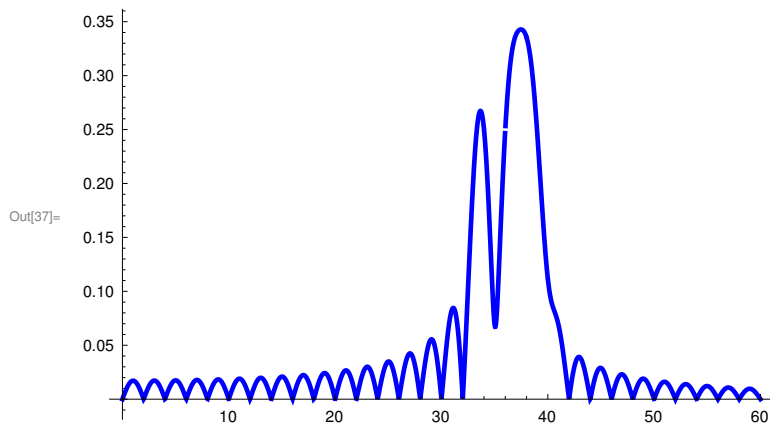
(Wegen der Modulation auf die Zwischenfrequenz sind die Amplitudenwerte halbiert)

```
In[33]:= (* spectofdmchip[w_]=FourierTransform [ofdmchip[t],t,w,FourierParameters ->{1, -1}];
In Mathematica 11.3 or 12.0 ging dies, leider nicht mehr Mathematica 12.1,
daher die Berechnung des Amplitudenspektrums nun anders: *)
```

```

In[34]:= f1[w_] = FourierTransform [fenster [t], t, w, FourierParameters -> {1, -1}];
f2[w_] = FourierTransform [ofdmmod [t], t, w, FourierParameters -> {1, -1}];
spectofdmchip [s_] = Convolve [f1[w], f2[w], w, s]/(2 Pi);
Plot[Abs[spectofdmchip [2 Pi f]], {f, 0, 60},
      PlotRange -> All, PlotStyle -> Directive [Blue, Thickness [0.007]]]

```



3.2 Demodulation

Die Demodulation erfolgt durch Multiplikation des empfangenen Signals, einmal mit $2 \cos(2 \pi f_c t)$, einmal mit $-2 \sin(2 \pi f_c t)$. Im ersten Fall erhält man die Inphase-Komponente $I(t)$ + höherfrequenten Rest, im zweiten die Quadratur-Komponente $Q(t)$ + höherfrequenten Rest. Die Reste werden durch ein Tiefpassfilter der Bandbreite $2 \pi f_c$ unterdrückt. **Dann ist $I(t) + j Q(t)$ wieder unser komplexes Basisband-Signal analog zum Beispiel oben unter 2. 2).**

Nachfolgend die Demodulation und wieder das schon gesehene Amplitudenspektrum des komplexen Basisband-Signals, nun wieder im Basisband:

(Hier muss man Mathematica etwas Zeit geben)

```

In[38]:= demod1[t_] = 2 ofdmmod[t] Cos[wc t];
specInphaser[ω_] = FourierTransform[demod1[t], t, ω, FourierParameters -> {1, -1}]
  (HeavisideTheta[ω + wc] - HeavisideTheta[ω - wc]);
Inphaser[t_] = InverseFourierTransform[specInphaser[ω],
  ω, t, FourierParameters -> {1, -1}];
demod2[t_] = - 2 ofdmmod[t] Sin[wc t];
specQuadraturr[ω_] = FourierTransform[demod2[t], t, ω, FourierParameters -> {1, -1}]
  (HeavisideTheta[ω + wc] - HeavisideTheta[ω - wc]);
Quadraturr[t_] = InverseFourierTransform[specQuadraturr[ω],
  ω, t, FourierParameters -> {1, -1}];
signalr[t_] = (Inphaser[t] + i Quadraturr[t]) Fenster[t];
specsignalr[ω_] = FourierTransform[signalr[t], t, ω, FourierParameters -> {1, -1}];
spec1 = Plot[Abs[specsignalr[2 Pi f]], {f, -20, 30}, PlotRange -> {0, 1},
  PlotStyle -> Directive[Blue, Thickness[0.007]], ImageSize -> Medium]
(* wie oben unter 2. in 2) *)

```

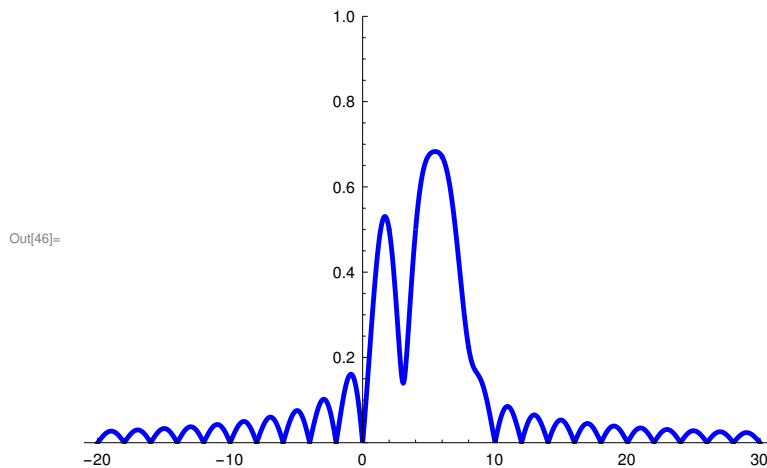


Bild 5 : Wieder das Ausgangsspektrum

3.3 Rückgewinnung der komplexen Amplituden des OFDM-Symbols durch Abtastung und eine DFT

Durch Abtastung und eine DFT der Abtastwerte gewinnt man die komplexen Amplituden des übertragenen OFDM-Symbols zurück (zum Vergleich mit dem Faktor $A = \sqrt{10}$ multipliziert).

```

In[47]:= abtastung = N[Table[signalr[n Ts/M], {n, 0, M-1}]]; (* Tabelle der Abtastwerte *)
ergebnisse = Sqrt[10] Chop[N[Fourier[abtastung, FourierParameters -> {-1, -1}]]]
(* Ergebnis der Demodulation * Sqrt[10] *)
amplituden (* ursprüngliche Auswahl komplexer Amplituden zum Vergleich *)

Out[48]= {0, 1. + 3. i, 3. + 1. i, -3. + 3. i, -1. - 1. i}

Out[49]= {0, 1 + 3 i, 3 + i, -3 + 3 i, -1 - i}

```

4. Diskretes Modell

Die heute weite Verbreitung hat das OFDM dadurch erreicht, dass man große Teile des Verfahrens durch diskrete Signalverarbeitung bewältigen kann, was enorme Kosten im Bereich der Übertragungs-Hardware einspart. Statt von Oszillatoren, Multiplizierern, Mischern usw. wird die Arbeit von Algorithmen und darauf ausgelegten Signalprozessoren übernommen.

Im diskreten Modell müssen wir Näherungen in Kauf nehmen: Wenn wir mit Abtastwerten von bandbegrenzten Funktionen arbeiten, dann sagt die Theorie, dass solche Signale nicht zeitbegrenzt sein können. Tiefpassfilterung von endlich vielen Abtastwerten eines Signals - wie oben in (I) der Funktionen $I(t)$ und $Q(t)$ - wird dann immer nur eine Näherung für das Signal $y(t)$ in (I) erlauben. Die Näherung wird eine Partialsumme der Abtastreihe im Shannonschen Abtasttheorem sein (dort mit idealem nicht-realisiertem Tiefpass). Aber auch diese ist wieder eine Orthogonal-Entwicklung. Die Näherung wird umso besser sein, je mehr Abtastwerte man verwendet. Wir verwenden nachfolgend **NN=64 Abtastwerte**.

Das Vorgehen im diskreten Modell lässt sich wie folgt beschreiben:

Man hat die Auswahl komplexer 16QAM-Amplituden für M Kanäle c_0, \dots, c_{M-1} und bildet für $n=0, \dots, NN-1$ und $t_n=n \cdot Ts/NN$ die **NN Abtastwerte** unseres "ofdmteil" (vgl. oben)

(II) $Y_n = \text{ofdmteil}[t_n]$

Aus der Folge $\frac{\pi}{B} Y_n$ wird durch Tiefpassfilterung das kontinuierliche Signal $s_m(t)$ im m -ten Zeitintervall der Dauer T_s erzeugt (vgl. [1], Abtasttheorem von Shannon), im Beispiel dann mit $m=0$ mit dem Rechteckfenster multipliziert. Dabei muss gelten: **$t_n = n\pi / B$, also $B = \pi NN / T_s$** für die Bandbreite des Tiefpassfilters (vgl. Abtast-Theorem).

(III) $s_m(t) = \sum_{n=0}^{NN-1} Y_n \frac{\sin(Bt-n\pi)}{Bt-n\pi} \cdot \text{fenster}(t-mT_s)$

Real – bzw. Imaginärteil von s_m interpolieren $I(t)$ bzw. $Q(t)$ aus (I) an den Stellen t_n und je mehr Abtaststellen man verwendet (siehe unten "upsampling"), desto besser wird s_m auch insgesamt approximiert.

Da die Fouriertransformation stetig ist, werden sich auch die Spektren beider Signale umso mehr annähern, je mehr Abtastwerte man zur Erzeugung von s_m analog zu (II) verwendet. Wir verwenden nachfolgend 64 Abtastwerte von $I(t)$ und $Q(t)$.

Um die in (II) gewünschten NN Abtastwerte aus den M gegebenen komplexen Amplituden zu bekommen, verwenden wir die IDFT der Amplitudenauswahl:

Eine einfache Möglichkeit, sich diese Werte (oder mehr) zu verschaffen, ist "zeropadding" in der Liste der komplexen Amplituden (man vgl. ggf. mein Notebook Nr.3, das unter der gleichen URL wie dieses zu finden ist.)

Aus dem Studium zur DFT (vgl. [1] oder mein Notebook zur DFT unter dieser URL) wissen wir, dass man durch sog. "zeropadding" vor Ausführung der IDFT ein "upsampling", sprich eine "höhere Zeitauflösung" bekommt und damit die gesuchten Werte im Zeitbereich. Man ergänzt die komplexe Amplitudenfolge einfach durch eine Reihe von Nullen. **Durch die IDFT mit zeropadding ergeben sich Abtastwerte für $I(t)$ und $Q(t)$ zu den Zeiten t_n , $n=0, \dots, NN-1$.**

$$Y_n = \text{Re}(\text{IDFT}\{\text{ofdmsymbol} + \text{zeropadding}\})_n, \quad n=0, \dots, NN-1$$

Die mit "zeropadding" erhaltene Folge Y_n , $n=0, \dots, NN-1$ spalten wir in Real- und Imaginärteil auf, filtern mit einem Tiefpassfilter der Bandbreite B und erzeugen damit Näherungen für die oben schon verwendeten Inphase- und Quadratur-Komponenten des OFDM-Signals. Bis vor das Tiefpassfilter zur D/A-Wandlung mit Fensterung kann also alles mit diskreter Signalverarbeitung gemacht werden. Der Rest ist D/A-Wandlung mit Quadraturmodulation durch Hardware der Nachrichtentechnik (Mischer). In neueren HW-Entwicklungen findet die Quadraturmodulation auch diskret mit sog. Direct Digital Synthesizern (DDS) vor der Tiefpass-Filterung mit einem D/A-Wandler (DAC) statt.

(* Filter-Input im mathematischen Modell: $s_{M,m} = \sum_{n=0}^{NN-1} \frac{\pi}{B} Y_n \delta(t - nTs/NN)$

Durch ein entsprechendes Demodulations-Equipment werden wieder die (nun näherungsweise) Inphase und Quadratur-Komponenten beim Empfänger rekonstruiert, woraus dann

durch Abtastung und DFT wie oben die komplexen Amplituden des OFDM-Symbols - unter der Voraussetzung einer störungsfreien Übertragung - gefunden werden können und das "Demapping" auf die entsprechenden 4 Bit-Gruppen erfolgen kann.

Ich mache in der anschließenden Beispiel-Rechnung **folgende Voraussetzungen**.

Das Beispiel zeigt, wie man auf der Empfangsseite die komplexen 16QAM-Amplituden zurückgewinnt, wenn das **empfangene Signal identisch zum gesendeten Signal ist** (ohne Kanal-Einflüsse).

- 1) Ich verwende zunächst eine Rechteckfensterung von $t=0$ bis $t=T_s$.
- 2) Ich verwende ein **ideales (nicht realisierbares) Tiefpassfilter**, um aus der Impulsfolge mit Impulsstärken $\pi Y_n/B$ ein kontinuierliches Signale mit *Mathematica* zu generieren.
- 3) Ich nehme an, dass die Inphase- und Quadratur-Komponenten genauso beim Empfänger ankommen/so rekonstruiert werden, wie sie hier im Basisband vorliegen. D.h. ich lasse insbesondere jegliche Art von Signalstörungen und Verzerrungen auf dem Übertragungsweg außer Betracht - und da gibt es in der Realität viele - und nehme an, dass Sende- und Empfangsseite vollkommen synchron und störungsfrei sind.

Nun also wieder zum vorangehenden Beispiel:

Zunächst die IDFT im Einsatz wie geschildert, Aufspaltung in Real- und Imaginärteil und Tiefpassfilterung der Impulsfolgen $\text{Re}(Y_n)$ und $\text{Im}(Y_n)$, wodurch Näherungen für die Inphase- und Quadratur-Komponenten als Analog-Signale erhalten werden.

(Ich lasse den Faktor A bei den komplexen Amplituden hier vorerst weg)

Vergleich der Näherung $\text{controll}[t]$ nach Quadraturamplitudenmodulation mit der des analogen Modells oben: Die aus Abtastwerten erzeugte Näherung $\text{controll}[t]$ in rot, die des analogen Modells in blau von $t=0$ bis $t=Ts$ zeitbegrenzt durch das Rechteckfenster $\text{fenster}[t]$:

Beide Analogsignale sind gleich, daher im Bild die rote Kurve mit Werte-Offset $+0.3$, damit man überhaupt zwei Funktionen erkennt.

```
In[60]:= controll[t_] = A (Inphasechip[t] Cos[wc t] - Quadraturchip[t] Sin[wc t]) fenster[t];
plot1 = Plot[ofdmchip[t], {t, 0, Ts},
  PlotRange -> All, PlotStyle -> Directive[Blue, Thickness[0.005]]];
plot2 = Plot[controll[t] + 0.3, {t, 0, Ts}, PlotRange -> All,
  PlotStyle -> Directive[Red, Thickness[0.007]]];
Show[{plot1, plot2}]
```

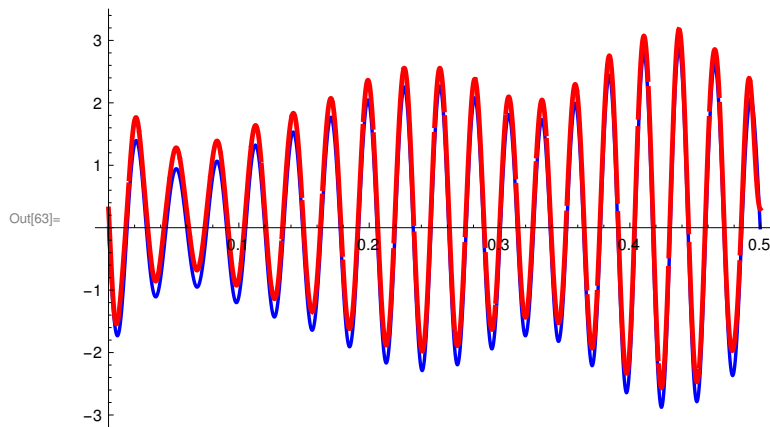


Bild 6: Blau das analoge Sendesignal, Rot das aus den Abtastwerten durch Tiefpass-Filterung mit einem idealen Tiefpass-Filter erzeugte Signal. Beide Graphen decken sich.

Rückgewinnung der 16QAM-Amplitudenauswahl nach Inversion der Quadraturamplitudenmodulation

Wie vorausgesetzt, nehmen wir nun an, dass auf Empfangsseite die beiden Komponenten der interpolierenden Näherungs-QAM nach Demodulation und Tiefpassfilterung **identisch** wieder vorliegen. Dann findet man genau wie oben in 3.3 durch Abtastung und eine DFT der Abtastwerte die komplexen Amplituden des OFDM-Symbols.

Die berechnete Liste als erste Ausgabe, darunter die Originale .

Beide Listen stimmen bis auf Rundungsfehler (wegen der Grenzwertbildungen und DFT-Berechnungen) überein (auch hier ist der Faktor A von oben wieder weggelassen), denn die Abtastung führt ja zurück auf die IDFT-Werte des Ausgangssignals.

```
In[64]:= wertere =
  Chop[N[Table[Limit[Inphasechip[t], t -> n Ts / M, Direction -> "FromBelow"], {n, 0, M - 1}]];
(* Abtastwerte der In-Phase-Komponente *)
werteim = Chop[
  N[Table[Limit[Quadraturchip[t], t -> n Ts / M, Direction -> "FromAbove"], {n, 0, M - 1}]];
(* Abtastwerte der Quadratur-Komponente *)
result = Chop[Fourier[wereiere + i werteim, FourierParameters -> {-1, -1}];
(* Zusammenfügen und DFT *)
Table[result[[k]], {k, 1, M}]
amplituden (* nochmal zum Vergleich *)
Out[67]= {5.93388 x 10-6 - 0.000489961 i, 1.0073 + 2.99911 i,
  3.03714 + 0.997672 i, -3.03715 + 3.00347 i, -1.00729 - 0.999758 i}
Out[68]= {0, 1 + 3 i, 3 + i, -3 + 3 i, -1 - i}
```

5. Realistischer und interessanter:

Verwendung anderer Fensterfunktionen zur Minderung von Außerbandstrahlung, "Cyclic Prefix" und Postfix für die Fensterung, diskrete Quadraturmodulation (wie mit DDS) und realisierbares Tiefpass-Filter als DAC

Wir hatten oben gesehen, dass das Amplitudenspektrum des OFDM-Signals bei Verwendung des Rechteckfensters im Zeitbereich nur sehr langsam abfällt. Dies kann in benachbarten Frequenzbändern zu Störungen führen (Interferenzen, wenn in Nebenbändern ebenfalls OFDM-Signale anderer Nutzer übertragen werden sollen). In der Praxis wird daher eine stärkere Dämpfung im Außerband verlangt. In den Spezifikationen für WLAN, DAB usw. werden dafür Spektralmasken definiert, die für die Übertragung eingehalten werden müssen.

Es gibt verschiedene Methoden, eine bessere Dämpfung im Außenband zu erreichen.

Man findet schnell eine große Anzahl von Publikationen zu diesem Thema unter Stichworten wie "**OFDM Pulse Shaping**". Zum Einsatz können Methoden wie Zeitfenster (W-OFDM für Windowed OFDM) kommen, Filterbänke zur Pulsformung (FBMC, "Filter Bank Based Multicarrier Systems") und andere Varianten des Mehrträgerverfahrens. Ein wichtiger Aspekt dabei ist die

Unschärfe-Relation von Heisenberg und die von Balian-Low, was die Zeit-Frequenz-Lokalisation von Signalen angeht (vgl. [1], Kap. 11).

Im Folgenden wird gezeigt, wie mit Hilfe einer Verlängerung der Signaldauer von T_s mit einem zugehörigen Zeitfenster "window[t]" der Dauer $T_s + T_s \cdot \alpha$ mit $\alpha = T_s/16$ an Stelle des oben verwendeten Rechteckfensters das Signalspektrum verändert werden kann. Dabei wird zugleich durch zyklische Erweiterung des Signals ein sog. "Cyclic Prefix" (CP) und ein "Postfix" eingeführt. Im Sendesignal hat dies eine gewisse "Inter Carrier Interferenz" (ICI) zur Folge. Diese Methode ist andererseits einfach zu implementieren und wird in realen Systemen auch so verwendet (vgl. [10] BROADCOM, L. Montreuil et al.). Das Thema "Reduktion von ICI- und Intersymbol-Interferenz (ISI)" bei OFDM und daraus abgeleiteten Varianten wird intensiv in zahlreichen Publikationen bearbeitet.

Ein besonders wichtiger Vorteil des CP ist, dass die Faltung mit der Kanal-Impulsantwort dann als zyklische Faltung dargestellt werden kann (vgl. [1] oder [3]), was eine zunächst theoretisch einfache Entzerrung von Einflüssen eines zeitinvarianten Übertragungskanals ermöglicht (Deconvolution einfach durch Divisionen mit Abtastwerten $H(2\pi k/T_s)$ des geschätzten Kanal-Frequenzgangs H . In der Praxis ist die Kanalverzerrung ein schwierigeres Thema, da die Faltungsgleichung $r(t) = (S * h + n)(t)$ ein sogenanntes *schlecht-gestelltes Problem* ist (r das empfangene Signal, S das gesendete Signal, h die Impulsantwort eines linearen zeitinvarianten Übertragungskanals, n hier gänzlich unberücksichtigtes hinzukommendes Rauschen). Man wird daher Modifikationen der einfachen Division durch Abtastwerte des Kanalfrequenzgangs benötigen. Verschiedene Entzerrer-Lösungen findet man in [3].

Nun zum Beispiel:

Ich verwende ein "Raised Cosine Window" im Zeitbereich nach den Empfehlungen in [10]. Es erweitert das bisherige Rechteckfenster nach beiden Seiten. In diesen Erweiterungen werden als **Prefix die letzten NN/4 Abtastwerte** aus der schon vorliegenden IDFT untergebracht, als **Postfix im Beispiel die ersten NN/16 unserer IDFT-Werte**, d.h. das Signal wird NN-periodisch fortgesetzt und die **Samples werden mit dem Fenster bewertet**.

Der Prefix-Bereich wird als Schutzintervall/Guard-Intervall (GI) bezeichnet.

Im Beispiel ist $NN=64$, also durch 16 teilbar. Nachfolgend fallen dann die ersten $NN/16 - 1 = 3$ Werte des Guard-Intervalls in die ansteigende Cos-Flanke des Fensters (als zu bezahlender Preis für die angestrebte bessere spektrale Formung des Sendesignals) und das Postfix enthält $NN/16 = 4$ Werte.

Im Anschluss an die Erweiterung der IDFT-Samples um GI und Postfix wird die **Quadraturmodulation wie mit einem "Direct Digital Synthesizer" (DDS) ebenfalls diskret** durch-

geführt und das diskrete Resultat mit einem realisierbaren Tiefpass-Filter interpoliert.

Wir betrachten dann dieses Signal im Vergleich zum OFDM-Signal von oben, erkennen Verzögerungen aufgrund des realisierbaren Tiefpassfilters in unserem W-OFDM-Signal, vergleichen die Spektren der beiden Signale und rekonstruieren unsere verwendeten komplexen Amplituden aus dem neu gefensterten Signal. Man erkennt daran, dass beim Empfänger (aus vielerlei Gründen) gut synchronisiert werden muss.

Zunächst das neue Zeitfenster mit Cos-Flanken; die Erweiterung des Rechteckfensters nach links enthält das Cyclic Prefix, die Erweiterung nach rechts das Postfix.

```
In[69]:= T = Ts
alpha = Ts / 16;
window1[t_] = (UnitStep[t + T (1 - alpha) / 2] - UnitStep[t - T (1 - alpha) / 2]) +
  1 / 2 (1 + Cos[Pi / (alpha T) (-t - T (1 - alpha) / 2]))
  (UnitStep[t + T (1 + alpha) / 2] - UnitStep[t + T (1 - alpha) / 2]) +
  1 / 2 (1 + Cos[Pi / (alpha T) (t - T (1 - alpha) / 2]))
  (UnitStep[t - T (1 - alpha) / 2] - UnitStep[t - T (1 + alpha) / 2]);
window[t_] = window1[t - T (1 + alpha) / 2];
windowduration = N[T + T alpha]
Plot[window[t], {t, 0, T + T alpha + 0.01}, PlotRange -> All]
```

Out[69]=
1
2

Out[73]= 0.515625

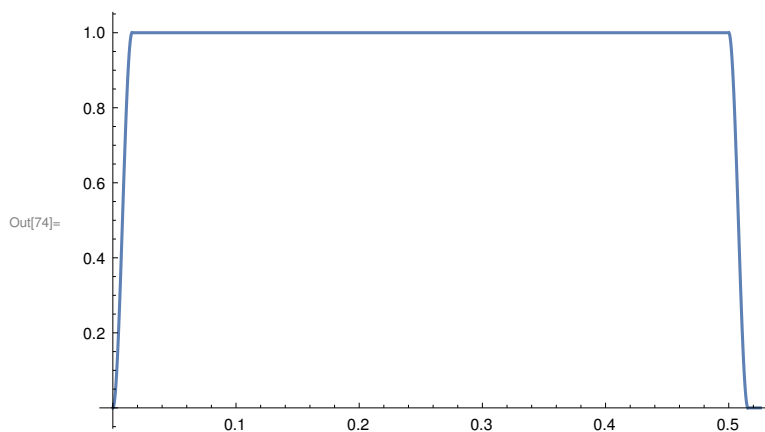


Bild 7 : Das verwendete Zeitfenster

```

In[75]:= specwindow[w_] = FourierTransform [window[t], t, w, FourierParameters -> {1, -1}]
(*= Ts Exp[I 2Pi Ts(1+alpha)/2]
   Sin[w Ts/2]/(w Ts/2)Cos[w alpha Ts/2]/(1-(alpha w/Pi Ts)^2);*)
Plot[Abs[specwindow[2 Pi s]], {s, 0, 9}, PlotRange -> All]

```

$$\begin{aligned}
 \text{Out[75]} = & -\frac{i e^{-\frac{i w}{64}}}{4(64\pi - w)} + \frac{i e^{-\frac{i w}{2}}}{4(64\pi - w)} + \frac{i e^{-\frac{33 i w}{64}}}{4(64\pi - w)} - \frac{i}{2w} - \frac{i e^{-\frac{i w}{64}}}{2w} + \\
 & \frac{i e^{-\frac{i w}{2}}}{2w} + \frac{i e^{-\frac{33 i w}{64}}}{2w} + \frac{i e^{-\frac{i w}{64}}}{4(64\pi + w)} - \frac{i e^{-\frac{i w}{2}}}{4(64\pi + w)} - \frac{i e^{-\frac{33 i w}{64}}}{4(64\pi + w)} - \frac{i w}{2(4096\pi^2 - w^2)}
 \end{aligned}$$

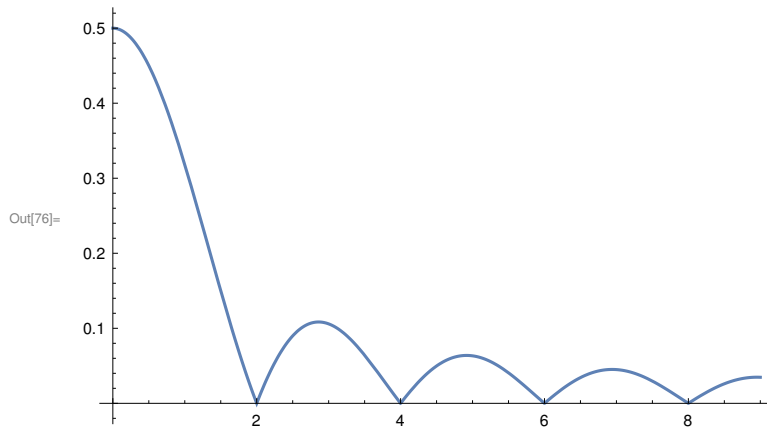


Bild 8: Das Betragsspektrum des neuen Fensters.

Es hat die gleichen Nullstellen wie das Spektrum des Rechteckfenster der Dauer T_s . Dies zeigt, dass die Interferenzfreiheit der Summanden im OFDM-Symbol mit Frequenzabständen $1/T_s$ mit dem neuen Zeitfenster zunächst nicht verloren geht. Dies wird aber der Fall sein, wenn das Fenster wegen Einfügen des CP verlängert wird. Außerdem stellt man fest, dass das Maximum mit der Zeitdauer des Fensters variiert. Insbesondere werden sich bei sehr kurzen Zeitfenstern (wie etwa bei WLAN mit $3.2 \cdot 10^{-6}$ s Kernsymboldauer) auch sehr kleine Spektralwerte dieser Größenordnung ergeben. Siehe Beispiel am Notebook-Ende.


```

In[87]:= coslist = Table[Cos[wc n Ts / NN], {n, 0, NWIN - 1}];
sinlist = Table[Sin[wc n Ts / NN], {n, 0, NWIN - 1}];
tpinput = Inphasewin coslist - Quadraturwin sinlist ;

In[90]:= controllwin2 [t_] =
  A Pi / B Sum[tpinput [[n + 1]] Sin[B (t - n Ts / NN)] / (Pi (t - n Ts / NN)), {n, 0, NWIN - 1}];
plot1win2 = Plot[ofdmchip[t], {t, 0, 3 Ts / 2}, PlotRange -> All,
  PlotStyle -> Directive[Blue, Thickness[0.005]]];
plot2win2 = Plot[controllwin2 [t], {t, 0, 3 Ts / 2}, PlotRange -> All,
  PlotStyle -> Directive[Red, Thickness[0.007]]];
Show[{plot1win2, plot2win2}]

```

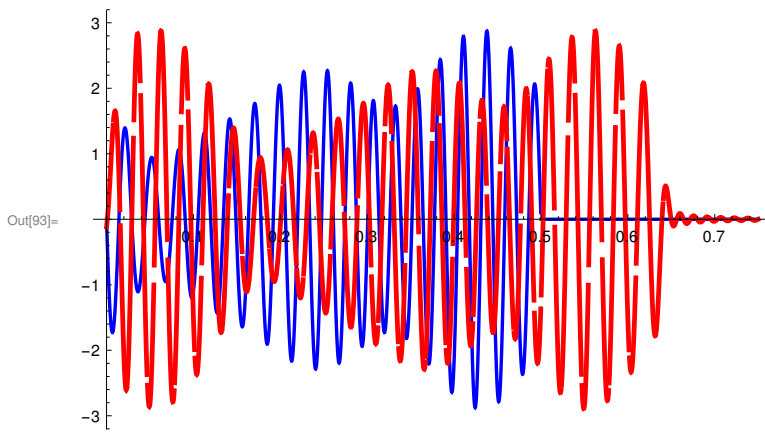


Bild 9 In blau das Signal mit dem Rechteck-Zeitfenster, in rot das zeitlich verlängerte Signal mit Prefix und Postfix, nach der Quadratur-Modulation, hier noch mit einem idealen Tiefpass der Bandbreite $B=64$ Hz als DAC-Simulation.

```

In[94]:= plot1win2 = Plot[ofdmchip[t], {t, 0, 3 Ts / 2},
  PlotRange -> All, PlotStyle -> Directive[Blue, Thickness[0.005]]];
plot2win2 = Plot[controllwin2 [t + Ts / 64 * 16] + 0.3, {t, 0, 3 Ts / 2},
  PlotRange -> All, PlotStyle -> Directive[Red, Thickness[0.007]]];
Show[{plot1win2, plot2win2}]

```

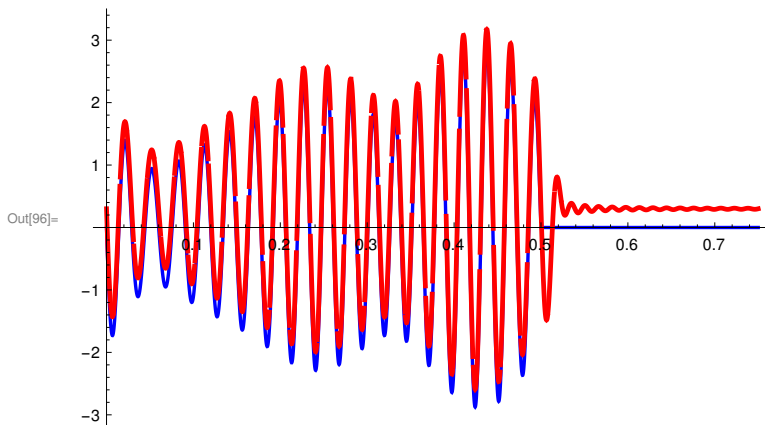


Bild 10 Da wir ja die Dauer des GI kennen, hier noch das um diese Zeit nach links verschobene neue Signal, das dann mit der blauen Kurve in $[0, Ts[$ gut

übereinstimmt (im Bild liegt die blaue Kurve fast direkt unter der roten, daher hier die rote mit Werte-Offset +0.3 gezeichnet).

Und nun der Vergleich der Amplitudenspektren der Signale mit dem Rechteckfenster (blau) und mit dem verwendeten "Raised Cosine Window", beide im Basisband und beide mit einem idealen Tiefpass bisher im Modell als DAC. Nachfolgend wird dieser noch durch ein realisierbares Tiefpass-Filter ersetzt.

Bemerkung: Eine alternative Möglichkeit zur Fensterung wäre auch ein diskretes Tiefpass-Filter, was aber auch eine Verlängerung der Zeitdauer mit sich brächte, da bei solchen Faltungen ja die Sample-Anzahl des Filters hinzukommt (wenn man das gefilterte Signal nicht abschneidet und damit den Filter-Effekt bei höheren Frequenzen zunichte macht).

(Man kann dies analog wie hier testen, indem man statt einer Fensterung ein passendes diskretes Tiefpassfilter verwendet (siehe andere Notebooks unter dieser URL). Ich überlasse dies interessierten Lesern zur eigenen Ausführung.

```
In[97]:= speccontrollwinbasis [w_] =
  A Pi / B Sum[(Inphasewin [[n + 1]] + I Quadraturwin [[n + 1]]) Exp[-I w n Ts / NN], {n, 0, NWIN - 1}];
pvgl = Plot[20 Log[10, abspecanalog [2 Pi f]], {f, -60, 60},
  PlotRange -> {-150, 3}, PlotStyle -> Directive [Blue, Thickness [0.004]]];
pwin1 = Plot[20 Log[10, Abs[speccontrollwinbasis [2 Pi s]]], {s, -60, 60}, PlotRange ->
  {-150, 3}, PlotStyle -> Directive [Red, Thickness [0.006]], PlotPoints -> 200];
Show[
  {pvgl,
  pwin1}]
```

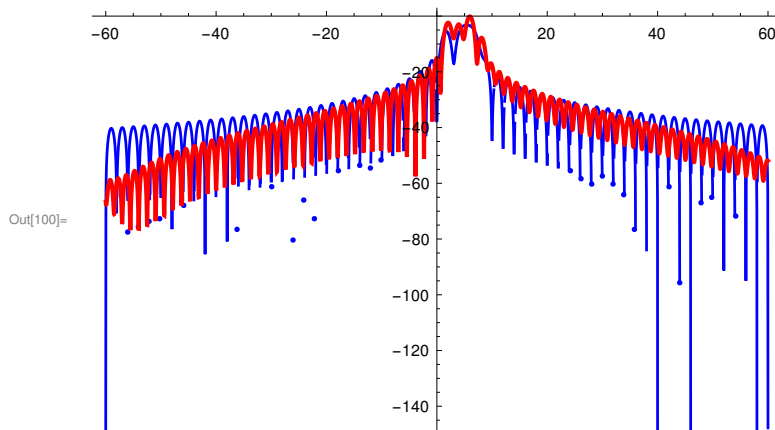


Bild 11 Amplitudenspektren in dB im Basisband, mit dem Rechteck-Zeitfenster in blau, mit dem "Raised-Cos-Fenster" in rot

An dieser Stelle stellt man fest, dass durch die verlängerte Zeitdauer des neuen Fensters ein Gleichanteil im Signal entstanden ist, also eine “*Intercarrier-Interferenz*” (*ICI*) -- analog auch bei den anderen Signalträgern. Erklärung: Dies ist ein sog. Leck-Effekt (engl. *Leakage*), der dadurch zustande kommt, dass die Samplingwerte mit einer IDFT gewonnen wurden, die wie das Analogsignal auf die Periode T_s ausgelegt war, während das neu gefensterete Signal mit CP nun eine andere Dauer hat, was dann diesen ICI-Effekt hervorruft (vgl. [1], Abschnitt 11.4).

```
In[101]:= 20 Log[10, Abs[speccontrollwinbasis [0]]]
(* Offenbart ICI durch die Einfügung des CP und
Verlängerung der Fensterdauer bei IDFT-
Sampling aber über die Zeitdauer Ts, hier
der Wert in dB beim unbesetzten "Gleichanteilsträger " *)

Out[101]:= -15.0025
```

Realisierbares Tiefpass-Filter als DAC

Ich konstruiere ein Butterworth-Tiefpass-Filter mit folgenden Daten (gefiltert wird das QM-Signal im Frequenzband bis $f_c + \text{Bandbreite}$ des OFDM-Symbols, d.h. bis 40 Hz im Beispiel. Damit das Filter hier noch möglichst linearen Phasengang hat, wähle ich 48 Hz als Passbandgrenze):

Passbandgrenze 48 Hz mit Verstärkung 0.9 an der Passbandecke
Stopbandgrenze 96 Hz mit maximaler Stopbandverstärkung 0.005

Das ergibt ein Butterworth-Tiefpass-Filter der Ordnung 9 mit einem **Delay von ca. 20 ms im Passband**. Zum Entwurf vgl. [1] oder mein Notebook unter der gleichen URL wie hier. Nachfolgend der Entwurf und die Impulsantwort.

```

In[102]:= K := 1;  $\omega_1 = 2 \pi 48$ ;  $\omega_2 = 2 \pi 96$ ; h1 := 0.9; h2 := 0.005; (*hier die Vorgaben*)
ord := N[Log[(K^2/h1^2 - 1)/(K^2/h2^2 - 1)]/(2 * Log[ $\omega_1/\omega_2$ ])];
n = Floor[ord] + 1; (* n = benötigte Filterordnung *)
 $\omega_g = \text{Sqrt}[\omega_1 \text{Exp}[-\text{Log}[K^2/h1^2 - 1]/(2 n)] \omega_2 \text{Exp}[-\text{Log}[K^2/h2^2 - 1]/(2 n)]$ ;
 $f_g = \omega_g / (2 \pi)$ ; (*  $2 \pi f_g = \omega_g = \text{Grenzkreisfrequenz im Frequenzgang } \hat{h}$  *)
z = Table[ $\omega_g \text{Exp}[i ((2 k - 1) \pi / (2 n) + \pi / 2)]$ , {k, 1, n}];
P[s_] := Product[Expand[(s - z[[k]]) (s - Conjugate[z[[k]])], {k, 1, n/2}];
If[EvenQ[n], H[s_] =  $K \omega_g^n / P[s]$ , H[s_] =  $K \omega_g^n / (P[s] (s - z[(n + 1) / 2]))$ ];
impulsantworttiefpass [t_] =
  Re[InverseFourierTransform [H[I w], w, t, FourierParameters -> {1, -1}]];
Plot[impulsantworttiefpass [t], {t, 0, 0.2}, PlotRange -> All, ImageSize -> Medium]

```

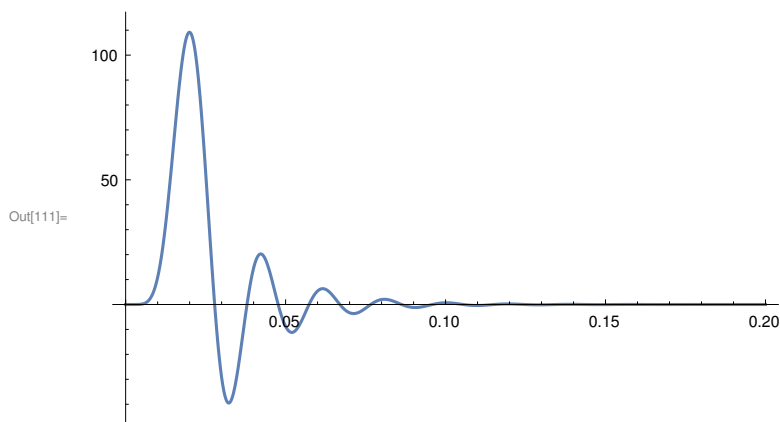


Bild 12a Impulsantwort des konstruierten Butterworth-Tiefpassfilters über Zeit t

```

In[112]:= Plot[20 Log[10, Abs[H[I 2 Pi f]]], {f, 0, 100}, PlotRange -> All, ImageSize -> Medium]

```

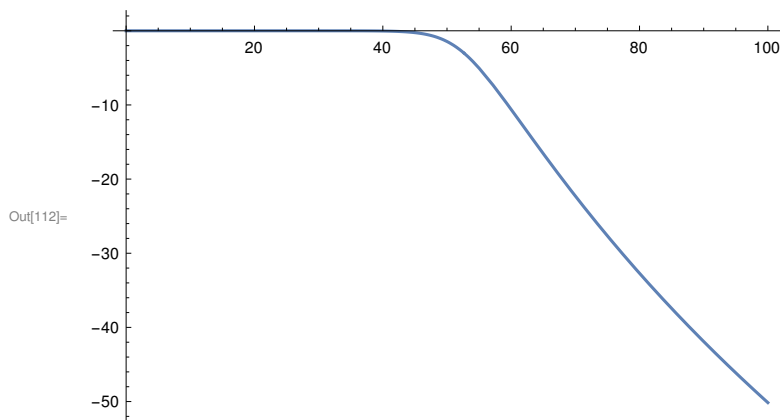


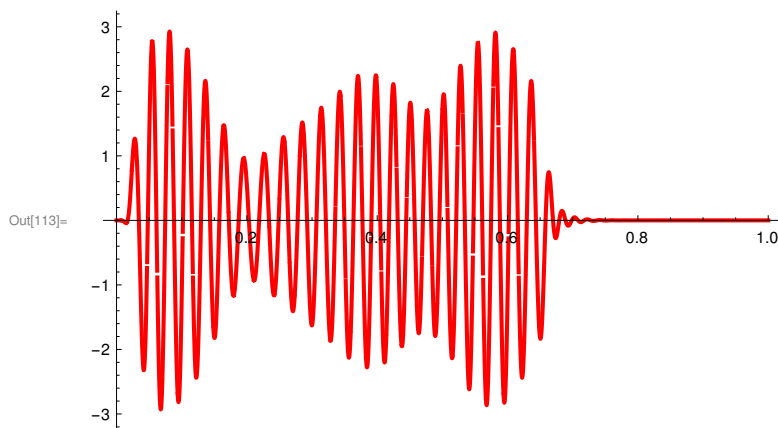
Bild 12b Amplitudengang des Butterworth-Tiefpassfilters über Frequenz f in dB

Nun die Filterung und ein Bild des Ergebnisses:

Anmerkung: 1) Zur Vergleichbarkeit muss hier der Faktor A wieder hinzugenommen werden.

2) Außerdem ist anzumerken, dass man wie im Shannonschen Abtasttheorem die Input-Samples mit dem Faktor π/B versehen muss (im math. Modell Dirac-Impulse, welche das π/B -fache der Samples als Impulsstärken haben). Anderenfalls erhält man einen falschen Output (math. Begründung: Der Faktor kommt beim idealen Tiefpassfilter ins Spiel, und dieses kann durch eine Folge von realisierbaren Tiefpässen mit Impulsantworten h_m approximiert werden -- in L_2 und im Distributionenraum D' . Da die Sample-Gruppe Y eine Distribution mit (zeit-)beschränktem Träger ist, konvergieren dann die Faltungen $Y * h_m$ gegen $Y * h$ mit der Impulsantwort h des idealen Tiefpasses. Der Faktor π/B bei den Impulsstärken ändert sich dabei nicht. (vgl. auch [1], Abschnitt 7.7, Konvergenz von Faltungen und Abtasttheorem in 11.1))

```
In[113]:= dacout[t_] = A  $\pi$  / B Sum[tpinput[[n + 1]] * impulsantworttiefpass [t - n Ts / NN], {n, 0, NWIN - 1}];
plotout = Quiet[Plot[dacout[t], {t, 0, 2 Ts}, PlotRange -> All,
  PlotStyle -> Directive[Red, Thickness[0.006]], PlotPoints -> 200]]
```



**Bild 13 Das generierte Zeitsignal am Ausgang unseres Butterworthfilters als DAC
Das Filter hat Verzögerung und Gruppenlaufzeit und von ca. 20 ms**

Da wir die Verzögerung des Filters kennen (ca. 20 ms, man rechne sie zur Übung nach) und ja auch die Dauer des Guard-Intervalls, können wir durch entsprechenden Zeitversatz die Übereinstimmung mit dem ursprünglichen OFDM-Chip testen.

```

In[114]:= plotout2 = Quiet[Plot[dacout[t+Ts/4+0.02]+0.3, {t, 0, 3 Ts/2}, PlotRange -> All,
      PlotStyle -> Directive[Red, Thickness[0.006]], PlotPoints -> 200]];
(* hier die letzte Kurve mit offset +0.3 in den Werten,
damit man überhaupt erkennt,
dass es 2 Kurven sind *)
Show[{plotout2, plot1win2}]

```

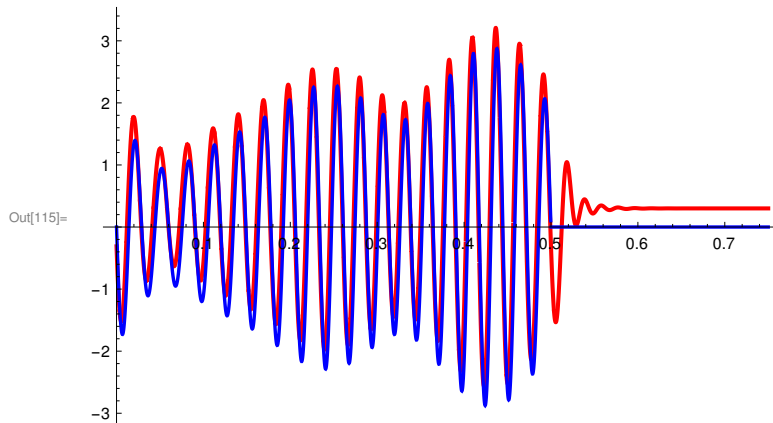


Bild 14 Zum Vergleich der ofdmchip vom Beginn mit dem Rechteck-Zeitfenster (blau) und der um $T_s/4+0.02$ nach links zeitversetzte Ausschnitt der Dauer T_s des gefilterten Signals (rot), gezeichnet mit Werte-Offset +0.3

Hier noch einmal das Amplitudenspektrum des Butterworth-gefilterten Signals (rot) im Basisband im Vergleich mit dem des Rechteckfensters wie oben (blau). Ich verwende dazu die Inphase- und die Quadratur-Samples vor der Quadraturmodulation und rechne mit dem Frequenzgang $H(f)$ des Butterworth-Filters.

```

In[116]:= specfiltbasis [w_] = A  $\pi$  / B
          Sum[(Inphasewin [[n + 1]] + I Quadraturwin [[n + 1]]) Exp[-I w n Ts / NN] H[I w], {n, 0, NWIN - 1}];
          pvg1 = Plot[20 Log[10, absSpecanalog [2 Pi f]], {f, -60, 60},
          PlotRange  $\rightarrow$  {-100, 3}, PlotStyle  $\rightarrow$  Directive [Blue, Thickness [0.004]]];
          pwinfilt = Plot[20 Log[10, Abs[specfiltbasis [2 Pi s]]], {s, -60, 60}, PlotRange  $\rightarrow$  {-100, 3},
          PlotStyle  $\rightarrow$  Directive [Red, Thickness [0.006]], PlotPoints  $\rightarrow$  200];
          Show[{pvg1, pwinfilt}]

```

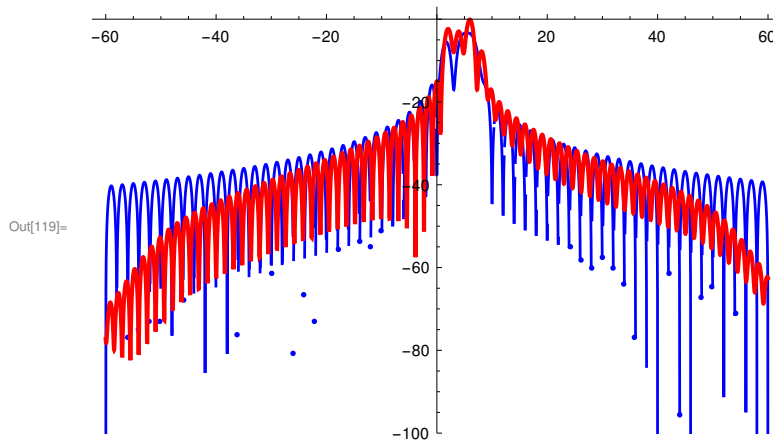


Bild 15 Amplitudenspektren in dB, Butterworth-geschnitten rot, Rechteck-Zeitfenster mit idealem Tiefpass blau.

Rückgewinnung der komplexen Amplituden des OFDM-Symbols aus dem Signal mit dem Raised-Cosine-Window, Entfernung des Cyclic Prefix: Entnahme eines Zeitausschnitts der Dauer T_s mit mäßig asynchroner Abtastung

Wir nehmen nun einen **Ausschnitt der Dauer T_s ab der Zeit $tr + \Delta t$** , also **nicht ganz synchron**, erreichen damit **Entfernung von Prefix und Postfix**, wodurch paarweise Orthogonalität zwischen den Trägern wieder hergestellt wird. Dann tasten wir eine Periode T_s ab (das Signal war T_s -periodisch fortgesetzt in Prefix und Postfix hinein), dann DFT zur Rekonstruktion der 16QAM-Amplituden. Hierzu genügen $M=6$ Abtastwerte für die gesuchten M Amplituden ($M=5$). **Das Ergebnis zeigt zunächst Phasenverschiebungen, da wir das Signal nicht genau bei $T_s + T_s/4$ beginnend abtasten, sondern mit einem gewissen Zeitversatz beginnend im Cyclic Prefix.** Da dieses aber eine periodische Fortsetzung des Signals nach links ist, können wir den **Phasenversatz korrigieren**, wenn wir z.B. die erste 16QAM-Amplitude ($1+3i$) als beim Empfänger bekannten **“Pilot-Wert”** bei der Frequenz $1/T_s$ verwenden. Ich generiere ohne

Umkehrung der Quadratur-Modulation die Inphase- und Quadratur-Komponenten hier (zur Rechensparnis) direkt aus ihren oben vorliegenden Samples.

Wir erhalten dann das **Ergebnis mit leichten Fehlern** aus den Rechnungen (einseitige Grenzwertbildungen, DFT usw.) und durch den "Leakage-Effekt" aufgrund Fensterung mit CP und vergleichen mit den verwendeten Amplituden (in beiden Fällen wieder **ohne den Faktor A** zur besseren Vergleichbarkeit) .

```
In[120]:= Inphasechipwinfilt [t_] =
  π / B Sum[Inphasewin [[n + 1]] * impulsantworttiefpass [t - n Ts / NN], {n, 0, NWIN - 1}];
Quadraturchipwinfilt [t_] =
  π / B Sum[Quadraturwin [[n + 1]] * impulsantworttiefpass [t - n Ts / NN], {n, 0, NWIN - 1}];
delayfilt = 0.021
Δt = 0.08
(*datawin=N[Table[
  Inphasechipwinfilt [n Ts/NN+Δt]+I Quadraturchipwinfilt [n Ts/NN+Δt], {n,0,M-1}]]];*)
datawin = N[Table[Limit[(Inphasechipwinfilt [t + delayfilt + Ts / 4 + Δt] +
  I Quadraturchipwinfilt [t + delayfilt + Ts / 4 + Δt]),
  t → n Ts / M, Direction → "FromAbove"], {n, 0, M - 1}]];
(* Abtastung, hier mit rechtsseitigen Grenzwerten
für
die
Abtaststellen,
damit Mathematica nicht 1/0 als
undefined erzeugt *)
result = Fourier[datawin, FourierParameters → {-1, -1}]
```

Out[122]= 0.021

Out[123]= 0.08

Out[125]= {-0.0013644 + 0.00095966 i, -2.10649 + 2.35922 i,
-2.3787 + 2.08454 i, 3.02492 - 2.97252 i, -0.378811 + 1.36073 i}

Nun noch die Phasenkorrektur wegen asynchroner Abtastung mit amplituden[[2]]=(1+3 i) als beim Empfänger bekannt vorausgesetzte Pilot-Amplitude und Vergleich mit den komplexen Amplituden des OFDM-Symbols. Trotz Rundungsfehlern bei den Rechnungen und "Fenster-Leakage" ist das Ergebnis m.E. zufriedenstellend.

```
In[126]:= z = result [[2]] / Abs[result [[2]]] / (amplituden [[2]] / Sqrt[10]);
endergebnis = Table[result [[k + 1]] z ^ (-k), {k, 0, 4}]
amplituden
```

Out[127]= {-0.0013644 + 0.00095966 i, 1.00016 + 3.00049 i,
3.0015 + 0.997246 i, -2.99428 + 3.00339 i, -1.00272 - 0.994806 i}

Out[128]= {0, 1 + 3 i, 3 + i, -3 + 3 i, -1 - i}

In[129]= 20 Log[10, Abs[A endergebnis [[1]]]] (* Rundungsfehler in dB
 beim Gleichanteil nach Entfernung von Prefix und Postfix *)
 Out[129]= -65.5556

Endergebnis zum Vergleich: a) rekonstruierte Amplituden, b) übertragene Amplituden (alles ohne Kanalverzerrungen und Rauschen bei identischem Empfang des Sende-Symbols), c) Leakage und Rundungsfehler bei 0 Hz in dB

Schlussbemerkungen:

- 1) Alles, was ich hier vorgestellt habe, hat nur modellhaften Charakter. In der realen Umsetzung des Prinzips sind insbesondere folgende Probleme zu lösen:
 - a) Durch die Superpositionen im Sendesignal können Amplitudenspitzenwerte entstehen, die eine Senderanlage de facto zerstören könnten, jedenfalls aber Signalverzerrungen und Außerbandstrahlung liefern, wenn der Sendeverstärker in nichtlineare Arbeitsbereiche gerät. Die Spitzenwerte korrespondieren zur Anzahl der verwendeten Träger: je mehr Träger, desto höher die Spitzenwerte. Hier muss der Nachrichtentechniker geeignete Lösungen finden (**Stichworte: Spitzenwertreduktion/ engl. Crest Factor reduction, PAPR-Reduction, Clipping, Transmit Filter, Pulse Shaping**) Man vgl. das Beispiel im Anhang unten.
 - d) Synchronisierung und Entstörung des empfangenen Signals beim Empfänger. Dazu verwendet man **Präambeln** zur Schätzung der Kanalstörung und **Pilot-Signale** in den Trägern der OFDM-Signale zur Synchronisierung (vgl. letztes Beispiel) und Schätzung von Kanal-Impulsantwort, Rauschen und Frequenz-Dispersion. Typische Störungen sind etwa: Der **Dopplereffekt bei mobilen Empfängern**, "**Fading**" und **Reflexionen mit Phasenversatz durch Mehrwegeausbreitung** u.a.m.
 - e) Entscheidung, welchen Elementen des 16QAM-Alphabets ein Einzel-Ergebnis zugeordnet wird: Zu jeder komplexen 16QAM-Amplitude gibt es eine Umgebung, so dass ein empfangenes Zeichen derjenigen Amplitude zugeordnet wird, in deren Umgebung es liegt.
 - f) "Demapping" der 16QAM-Amplituden, P/S-Umsetzung und Decodierung in die ursprüngliche 0-1-Bitfolge, die das Nutzsignal repräsentiert.

- 2) Beispieldaten aus [3] etwa zum DVB-T mit 16QAM im 8k-Mode und 8 MHz-Betrieb sind:

Subträger-Anzahl bei 8 MHz Bandbreite:	6817
Kernsymboldauer	0,896 ms
Guardintervall	224 µs
Subträgerabstand	1,116 kHz
FFT	8 k

3) Hier zum Schluss noch ein **Beispiel eines WLAN-Spektrums** mit den zugehörigen Rahmendaten von IEEE802.11a. Ich generiere ein einziges Symbol, das einfach die 16QAM-Amplitudenliste camp vom Beginn des Notebooks zyklisch auf die verwendeten Träger verteilt und ersetze in den Kanälen 7, 21, 43, 57 als Pilotwerte -10 auf Kanal 21, +10 sonst (hier zur Sichtbarkeit im nachfolgenden Amplitudenspektrum; in Wahrheit sind bei WLAN diese Werte -1 statt -10, +1 statt +10). Ich wähle als Frequenzen im Basisband -10^7 Hz bis $+9.6875 \cdot 10^6$ Hz, d.h. mit Null als Mittenfrequenz. Der Trägerabstand ist 312.5 kHz, die Bandbreite 20 MHz. 11 Randträger bleiben unbesetzt mit Amplitudenwerten Null. Kernsymboldauer ist 3.2×10^{-6} s ohne Prefix/Postfix. Bei Hinzurechnung der Zeit für andere Daten im WLAN-Rahmen ergibt sich mit Coderate 1/2 eine Bitrate von 24 Mbit/s mit 16QAM (vgl. [3]).

In[130]:=

```
wlana = Table[If[m ≤ 16, camp[[m]], camp[[m - 16]]], {m, 1, 32}];
wlanb = Flatten[Append[wlana, wlana]];
Length[wlanb];
wlanb[[28 ; 38]] = 0; (* Randträger mit Amplitudenwerten Null versehen *)
wlanb[[1]] = 0;
wlanb[[8]] = 10;
wlanb[[22]] = -10;
wlanb[[44]] = 10;
wlanb[[58]] = 10;
wlansymbol = wlanb;
wlanb
Length[wlanb]
Ts = 3.2 * 10^(-6); (* Zeitdauer Kernsymbol *)
TrAbstand = 1/Ts ; (* Trägerabstand *)
M = Length[wlansymbol];
fr = Flatten[
  Append[Table[k TrAbstand, {k, 0, M/2 - 1}], Table[-k TrAbstand, {k, M/2, 1, -1}]];
tr = Ts/4;
windowduration = N[Ts + tr + tr/4] (* Verlängerung der Zeitdauer des OFDM-
  Symbols von Ts auf Ts + 5 tr/4 für Prefix und Postfix wie oben*)
window[t_] = (0.5 (1 - Cos[Pi t/(tr/4)]) (UnitStep[t] - UnitStep[t - tr/4]) +
  (UnitStep[t - tr/4] - UnitStep[t - (Ts + tr)]) +
  0.5 (1 + Cos[Pi (t - (Ts + tr))/(tr/4)]) (UnitStep[t - (Ts + tr)] - UnitStep[t - (Ts + 5 tr/4)]));
Plot[window[t], {t, 0, 2 Ts}] (* Plot des Fensters *)
```

$$\text{Out[136]= } \left\{ 0, \frac{1+3i}{\sqrt{10}}, \frac{3+3i}{\sqrt{10}}, \frac{3+i}{\sqrt{10}}, \frac{1-i}{\sqrt{10}}, \frac{1-3i}{\sqrt{10}}, \frac{3-i}{\sqrt{10}}, 10, -\frac{1-i}{\sqrt{10}}, -\frac{1-3i}{\sqrt{10}}, -\frac{3-3i}{\sqrt{10}}, -\frac{3-i}{\sqrt{10}}, \right. \\ \left. -\frac{1+i}{\sqrt{10}}, -\frac{1+3i}{\sqrt{10}}, -\frac{3+i}{\sqrt{10}}, -\frac{3+3i}{\sqrt{10}}, \frac{1+i}{\sqrt{10}}, \frac{1+3i}{\sqrt{10}}, \frac{3+3i}{\sqrt{10}}, \frac{3+i}{\sqrt{10}}, \frac{1-i}{\sqrt{10}}, -10, \frac{3-i}{\sqrt{10}}, \frac{3-3i}{\sqrt{10}}, \right. \\ \left. -\frac{1-i}{\sqrt{10}}, -\frac{1-3i}{\sqrt{10}}, -\frac{3-3i}{\sqrt{10}}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \frac{3-i}{\sqrt{10}}, \frac{3-3i}{\sqrt{10}}, -\frac{1-i}{\sqrt{10}}, -\frac{1-3i}{\sqrt{10}}, \right. \\ \left. -\frac{3-3i}{\sqrt{10}}, 10, -\frac{1+i}{\sqrt{10}}, -\frac{1+3i}{\sqrt{10}}, -\frac{3+i}{\sqrt{10}}, -\frac{3+3i}{\sqrt{10}}, \frac{1+i}{\sqrt{10}}, \frac{1+3i}{\sqrt{10}}, \frac{3+3i}{\sqrt{10}}, \frac{3+i}{\sqrt{10}}, \frac{1-i}{\sqrt{10}}, \right. \\ \left. \frac{1-3i}{\sqrt{10}}, \frac{3-i}{\sqrt{10}}, \frac{3-3i}{\sqrt{10}}, -\frac{1-i}{\sqrt{10}}, 10, -\frac{3-3i}{\sqrt{10}}, -\frac{3-i}{\sqrt{10}}, -\frac{1+i}{\sqrt{10}}, -\frac{1+3i}{\sqrt{10}}, -\frac{3+i}{\sqrt{10}}, -\frac{3+3i}{\sqrt{10}} \right\}$$

Out[137]= 64

Out[143]= 4.2×10^{-6}

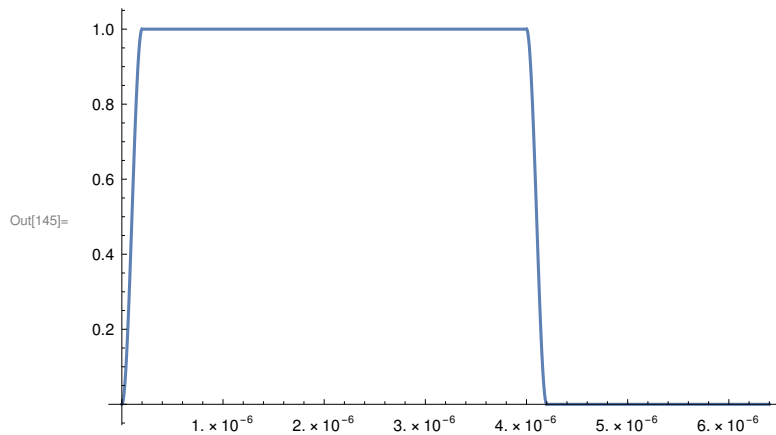


Bild 16 Zeitfenster der Dauer 4.2×10^{-6} s

Im folgenden Bild der Betrag der Fouriertransformierten des Zeitfensters mit sehr kleinem Maximum. Ich habe diesmal zur Abwechslung die **Laplace-Transformation** als Alternative zur Fouriertransformation verwendet. Die Abszisse ist auf Kanalnummern normiert. Man erkennt den oben schon angesprochenen Leakage-Effekt durch Prefix und Postfix.

```
In[146]:= ltwindow[s_] = LaplaceTransform [window[t], t, s];
```

In[147]:= Plot[Abs[Ltwindow[I 2 Pi s / Ts]], {s, 0, 5}, PlotRange -> All]

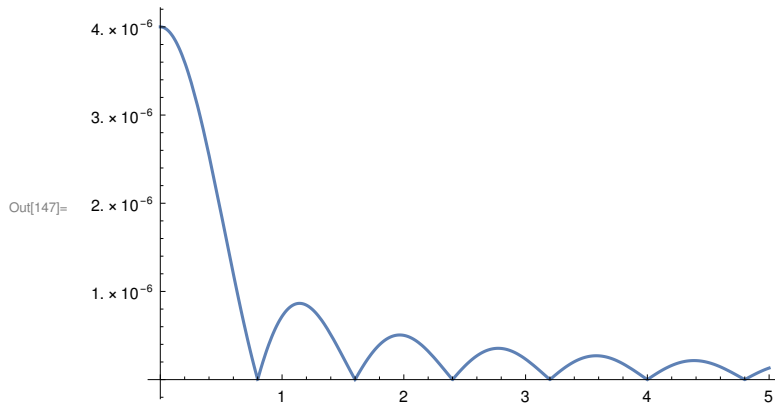


Bild 17 Betrag der FT des Fensters, halbseitig

Und schließlich ein Bild des **normalisierten WLAN-Energiedichte-Spektrums** (quadrierter Betrag der Fouriertransformierten) in dB für dieses OFDM-Symbol im Basisband (engl. *normalized energy density spectrum*).

Bemerkung: Das Signal ist ein Energiesignal. Als solches besitzt es **kein normalisiertes Leistungsdichte-Spektrum** (engl. *power density spectrum*) $P(f)$ (f für Frequenz, $s(t)$ das Signal) im üblichen Sinn,

$$P(f) = \lim_{T \rightarrow \infty} \frac{1}{T} \left| \int_{-\pi/2}^{\pi/2} s(t) e^{-i2\pi f t} dt \right|^2 = 0.$$

In rot das erzeugte **normalisierte Energiedichte-Spektrum** mit dem obigen Fenster (hier ohne Cyclic Prefix oder Postfix nur für einen Eindruck), in blau nochmal analog mit dem Rechteckfenster zum Vergleich. Hat das Eingangssignal V als physikalische Einheit, dann hat das **normalisierte Energiedichte-Spektrum** (mit normierendem Wirkwiderstand $R=1 \Omega$) die Einheit $V^2 s^2$.

```

In[148]:= specwLansymbol [w_] = Sum[wLansymbol [[n + 1]] * ltwindow[I (w - 2 Pi fr[[n + 1]])], {n, 0, M - 1}];
plotwlanwinenergydensitiespectrum = Plot[20 Log[10, Abs[specwLansymbol [2 Pi s / Ts]],
    {s, -60, 60}, PlotRange -> {-170, -80}, PlotStyle -> Directive [Red, Thickness [0.006]]];
ltrechteck [s_] = LaplaceTransform [UnitStep[t] - UnitStep[t - Ts], t, s];
specwLansymbolr [w_] =
    Sum[wLansymbol [[n + 1]] * ltrechteck [I (w - 2 Pi fr[[n + 1]])], {n, 0, M - 1}];
plotwlanrechteckenergydensitiespectrum =
    Plot[20 Log[10, Abs[specwLansymbolr [2 Pi s / Ts]], {s, -60, 60},
    PlotRange -> {-170, -80}, PlotStyle -> Directive [Blue, Thickness [0.003]]];
Show[{plotwlanwinenergydensitiespectrum, plotwlanrechteckenergydensitiespectrum}]

```

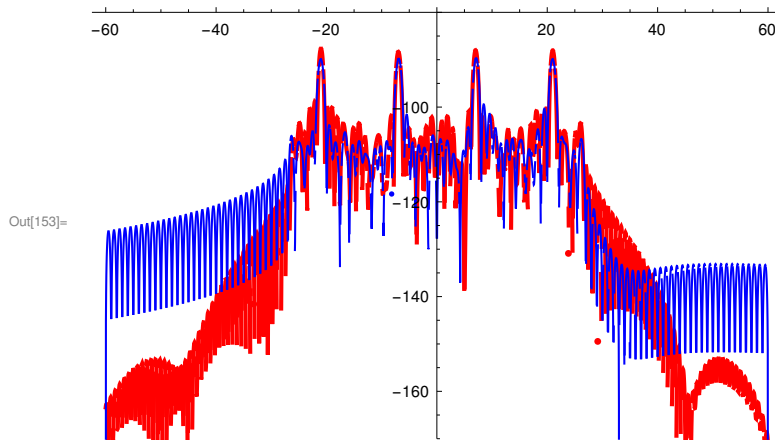


Bild 18 dB-Energiedichte-Spektrum des generierten WLAN-OFDM-Symbols im Basisband. Die Abszisse zeigt für ganzzahlige Werte die Kanalnummer. Man sieht die erhöhten Amplituden der 4 Pilotträger klar hervorragen.

Zu Spektralmasken in den WLAN-Standards.

In IEEE-Standards von WLAN werden Spektralmasken angegeben, die beim Sender einzuhalten sind. In vielen Büchern zur Nachrichten-Übertragung werden auch sog. "Leistungsdichte-Spektren" oder "Leistungsspektren" angegeben und grafisch dargestellt. Das macht Sinn (wir haben ja eigentlich ein **Energie-Signal**), wenn man voraussetzt, dass dabei das *Signal mit der Fensterdauer T_w periodisiert* wird, oder aber man betrachtet das *OFDM-Signal als (cyclo-stationären) stochastischen Prozess und berechnet dessen mittlere spektrale Leistungsdichte resp. sein "zyklisches Spektrum"*. Im ersten Fall erhält man für das gerade betrachtete eine OFDM-Symbol als **diskretes Spektrum** die Fourierkoeffizienten der zugehörigen Fourierreihe. Diese sind mit der Fensterdauer T_w **Abtastwerte an den Stellen $(2 \pi k/T_w)$ des oben errechneten kontinuierlichen Spektrums versehen mit dem Faktor $1/T_w$** (vgl [1], Abschnitt 10.5 zur Poissonsche Summenformel). Für $f_0=1/T_w$ ist das mit Wirkwiderstand $R=1 \Omega$ **normalisierte Leistungsdichte-Spektrum $p(f)$ des T_w -periodisch fortgesetzten Signals eine Impulsfolge $\delta(f-nf_0)$ mit den quadrierten Beträgen der**

Fourier koeffizienten als zugehörigen Impulsstärken . Die normalisierte Leistung des periodisierten Signals ist entsprechend die Reihe mit den Betragsquadraten dieser Fourierkoeffizienten, was wir bei Fourierreihen als **Parsevaltheorem** kennen gelernt haben (vgl. [1]).

Zum Nachweis der Einhaltung von Spektralmasken - zum Test von entwickelter Übertragungs-Hardware etwa - werden in der Praxis viele (Größenordnung einige tausend oder mehr) OFDM-Symbole - mit einer Zufalls-Bitfolge - erzeugt und die wie oben berechneten Spektren (meist näherungsweise mit Hilfe der FFT) gemittelt (*averaged periodogram technique* ; es sind verschiedene Mittelungsverfahren in Gebrauch, etwa *Bartlett's method*, *Welch's method* u.a.) Das Ergebnis dient als Näherung für eine *mittlere spektrale Leistungsdichte eines stochastischen Prozesses (s.o.) mit zufälligen Trägeramplituden aus dem 16QAM-Alphabet* (oder entsprechenden Amplituden bei anderen Modulationen wie 64QAM, BPSK etc.). Es gibt Formeln für die Autokorrelationsfunktion und ihre Fouriertransformierte unter variierenden Voraussetzungen an den modellierten stochastischen Prozess (siehe [11] und dort genannte weitere Referenzen). Es ist intuitiv klar, dass - nach Konstellation und Normierung (Faktor A) des OFDM-Alphabets - *diese spektrale Leistungsdichte des "stochastischen OFDM-Prozesses" durch die Fouriertransformierte des Zeitfensters und den Frequenzgang des interpolierenden Tiefpassfilters (DAC) bestimmt ist.*

Hier noch der zugehörige dB-Plot der mit dem Faktor (1/Fensterdauer) multiplizierten Spektren im Basisband. Sie gelten **für das eine betrachtete OFDM-Symbol** . Die Abtastwerte dieser Spektren sind die Fourierkoeffizienten des jeweils periodisierten Signals, mit normierter Frequenz ($f T_s$) dargestellt wie oben. Man erkennt auch daran, dass mit dem "Raised Cosine Window" die Außerbandstrahlung in weit höherem Maß gedämpft wird als dies beim Rechteckfenster der Fall ist. Erst damit wird die für das WLAN-Beispiel vorgegebene Spektralmaske im IEEE-Standard erreicht.

In[154]:=

```

Tw = windowduration ;
plotwlanwinspectrumfakt =
  Plot[20 Log[10, Abs[specwlan[symbol [2 Pi s / Ts]] / Tw], {s, -80, 80}, PlotRange -> {-120, 20},
  PlotStyle -> Directive[Red, Thickness[0.006]], ImageSize -> Large];
plotwlanrechteckspectrumfakt = Plot[20 Log[10, Abs[specwlan[symbolr [2 Pi s / Ts] / Ts]],
  {s, -80, 80}, PlotRange -> {-120, 20},
  PlotStyle -> Directive[Blue, Thickness[0.003]], ImageSize -> Large];
Show[{plotwlanwinspectrumfakt , plotwlanrechteckspectrumfakt }]

```

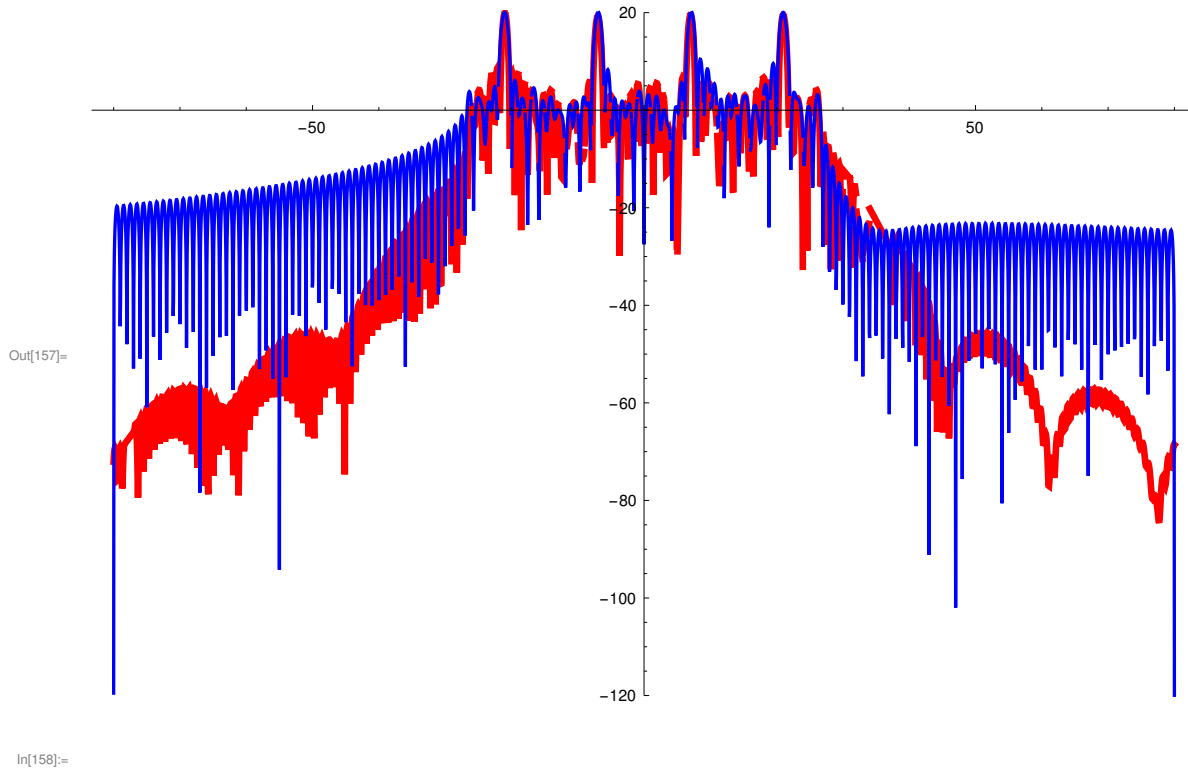


Bild 19 Abtastwerte ergeben diskrete dB-Leistungsdichte-Spektren der periodisierten generierten WLAN-OFDM-Symbole im Basisband. Die Abszisse zeigt für ganzzahlige Werte die Kanalnummer.

Es ist eine enorme Leistung von Nachrichtentechnikern, dass DSL, WLAN, DVB-T und DAB, DRM heute mit vielen Feinissen so weit entwickelt sind, dass diese Technik für vergleichsweise sehr niedrige Preise in einer Kombination aus Software und leistungsfähiger Hardware verfügbar ist, in denen die gesamte Übertragungstechnik steckt für den TV-Empfang auf einem Notebook, in Mobilfunkgeräten, in billigen WLAN-Sticks + Software usw. Das alles, damit Menschen -- als heute wohl eine vielfache "Nutzung" -- zweifelhafte Websites besuchen, ihr Mittagessen oder Selfies fotografieren und durch die Welt "posten" können, aber gelegentlich ja auch für Sinnvolleres. Allen Lesern, die das Thema spannend finden, sei unten eine nur kleine Auswahl von Referenzen genannt. In jeder Hochschul-Bibliothek findet man schnell viele weitere.

Anhang 1: Noch einmal zur Quadraturamplitudenmodulation und zur Wahrnehmung von Amplitudenmodulationen im Hörbereich

Die Fourier-Analyse beschreibt vielfache und auch ganz unterschiedliche Phänomene durch “*die wunderbare Welt der Schwingungen*”. Ein Vorteil dieser Beschreibungsmöglichkeit - es gibt ja auch ganz andere Möglichkeiten, mathematisch grob gesagt mit anderen Funktionensystemen als “Basisfunktionen” wie Wavelets u.v.a.m. - ist, dass Sachverhalte, die Schwingungen betreffen, oft ganz unabhängig von der Frequenz sind, d.h. in unterschiedlichen Frequenzbereichen erklärt, veranschaulicht oder erfahrbar gemacht werden können.

Es bietet sich an, eine Quadraturamplitudenmodulation wie oben im Hörbereich erfahrbar zu machen und nicht nur abstrakt mathematisch zu verstehen.

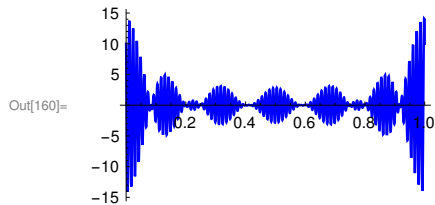
Ein einfaches Beispiel einer 16QAM-Modulation mit mehreren Trägern kann man im Hörbereich als An- und Abschwellen der Lautstärke/des Schalldrucks bei gleichbleibenden Tonhöhen im Akkord hörbar machen. *Ganz ähnlich (vgl. Bild unten) - nur außerhalb des Hörbereichs und wesentlich schneller im Zeitablauf und nur eine Periodendauer der Grundschwingung lang - kann man sich ein OFDM-Signal im HF-Bereich vorstellen.* Ich verwende nachfolgend für mein Beispiel folgende Daten:

Ich wähle die Symboldauer $T_s=1$ s, zehnmal die komplexe Amplitude $c = 1+1i$ (ohne den Faktor $1/\sqrt{10}$) und plote die wie in (I) entstehende 16QAM-Modulation über die Symboldauer. Inphase- und Quadratur-Komponente sind bis auf eine additive Konstante **Dirichlet-Kerne mit Periode $T=1$** , deren zentrale Rolle in der Theorie der Fourierreihen wohl bekannt ist.

Man erkennt an diesem Beispiel sofort die **Spitzenwert-Problematik**: Stellen Sie sich vor, dass tausende Träger wie beim DVB-T die gleiche komplexe Amplitude $c=1+i$ im gleichen Zeitabschnitt übermitteln sollen (vgl. Werte der Dirichlet-Kerne bei Null und Eins). Die dabei entstehenden Spitzenwerte könnten ohne nachrichtentechnische Gegenmaßnahmen nicht übertragen werden.

```
In[159]:= testsignal [t_] =
  ComplexExpand [Re[Exp[I 2 Pi 80 t] (Sum[(1 + I) Exp[I 2 Pi k t], {k, -5, 5}] - (1 + I))]
Plot[testsignal [t], {t, 0, 1}, PlotRange -> All,
  PlotStyle -> Directive[Blue, Thickness[0.008]], ImageSize -> Small]
```

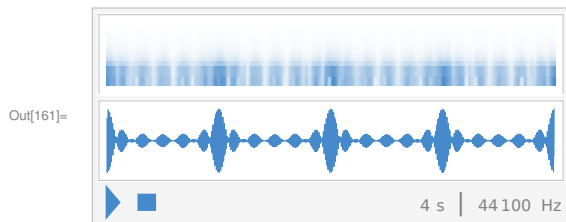
```
Out[159]= 2 Cos[2 π t] Cos[160 π t] + 2 Cos[4 π t] Cos[160 π t] +
  2 Cos[6 π t] Cos[160 π t] + 2 Cos[8 π t] Cos[160 π t] +
  2 Cos[10 π t] Cos[160 π t] - 2 Cos[2 π t] Sin[160 π t] - 2 Cos[4 π t] Sin[160 π t] -
  2 Cos[6 π t] Sin[160 π t] - 2 Cos[8 π t] Sin[160 π t] - 2 Cos[10 π t] Sin[160 π t]
```



Wir sehen im Bild oben den zeitlich schwankenden Amplitudenverlauf. Das Ganze durch die Mittenfrequenz $f_c = 80$ Hz im Hörbereich (das ist in etwa die Note E in der sog. Großen Octave).

Die 16QAM - 4-fach wiederholt - ist nun umgangssprachlich wie ein "Blubbern" deutlich zu hören.

```
In[161]:= Sound[SampledSoundList [Table[testsignal [t], {t, 0, 4, 1. / 44 100}], 44 100]]
```



```
In[162]:= Export["qamsound .wav",
  Sound[SampledSoundList [Table[testsignal [t], {t, 0, 4, 1. / 44 100}], 44 100]], "Sound"]
```

```
Out[162]= qamsound .wav
```

Mit Lautsprechern am Notebook über eine Soundkarte ist das gut zu hören. Eine einzige Periode würde als OFDM-Signal die Information über 40 Bit des Nutzsymbols transportieren.

Empfehlenswerte Referenzen:

- [2] S.B. Weinstein, P.M. Ebert Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform
IEEE Transactions on Communication Technology, Vol. COM-19, No. 5, October 1971
- [3] K.-D. Kammeyer, A. Dekorsy Nachrichtenübertragung
Vieweg+Teubner, 6. Auflage 2018
- [4] O. Mildenberger Übertragungstechnik, Vieweg1997
- [5] M. Werner Nachrichten-Übertragungstechnik, Vieweg 2006
- [6] H. Schulze, C. Lüders Theory And Applications Of OFDM And CDMA
Wiley 2005
- [7] European Telecommunications Standards Institute Digital Radio Mondiale (DRM); System Specification
ETSI ES 201 980 V4.1.1 (2014-01)
- [8] Wikipedia, Google et al. Suchen Sie unter Stichworten wie 16QAM, OFDM, OFDM Pulse Shaping u.a.
- [9] Analog Devices DAC mit DDS: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9857.pdf>
- [10] BROADCOM, L. Montreuil et al. OFDM TX Symbol Shaping
www.ieee802.org/3/bn/public/jan13/montreuil_01a_0113.pdf
- [11] van Waterschoot et. al Analytical expressions for the power spectral density of CP-OFDM and ZP-OFDM signals,
IEEE Signal Processing Letters, 17(4), 2010, 371-374 oder Anonymos ftp from ftp.esat.kuleuven.be in the directory pub/sista/vanwaterschoot/reports/09-148.pdf