JOHANNES KEPLER
UNIVERSITÄT LINZ

Stephanie Vidovic
k01505694,
Simon Grundner
k12136610
Institute of
Signal Processing

k01505694@students.jku.at
k12136610@students.jku.at
https://jku.at/isp

14. Mai 2025

UE03 Signalverarbeitung

Protokoll

Gruppe 13 - SoSe 2025

Schlagwörter DFT, FFT, STFT

INSTITUTE OF SIGNAL PROCESSING

JOHANNES KEPLER
UNIVERSITÄT LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Inhaltsverzeichnis

1. Aufgabe

DFT vs. FFT (10 Punkte)

In diesem Problem werden Sie die Ausführungszeiten von DFT und FFT Implementierungen vergleichen. Dazu sollen Sie zunächst in Matlab / Octave Funktionen schreiben, welche die DFT sowie die inverse DFT (IDFT) implementieren, indem Sie die jeweiligen Definitionsgleichungen umsetzen.

Schreiben Sie dann eine Funktion `eval_dft_vs_fft(N)`, welche zunächst ein komplexwertiges Zufallssignal der Länge N erstellt. Vergleichen Sie anschließend die Ausführungszeiten ihrer Implementierungen mit den Matlab built-in Funktionen (FFT + IFFT), indem Sie das komplexwertige Signal zunächst transformieren (DFT bzw. FFT) und dann wieder rücktransformieren (IDFT bzw. IFFT). Das heißt Ausgangssignal sollte gleich dem Eingangssignal sein.

Um die Laufzeiten zu vergleichen verwenden Sie die Matlab Funktionen `tic` und `toc`. Zusätzlich sollen Sie den mittleren quadratischen Fehler (RMS error) zwischen den jeweiligen Eingangs- und Ausgangssignalen messen und vergleichen.

Die Definitionsgleichungen der DFT/IDFT lauten

$$F_D[k] := \sum_{n=0}^{N-1} f_D[n] W_N^{kn}, \qquad f_D[n] := \frac{1}{N} \sum_{k=0}^{N-1} F_D[k] W_N^{-kn} \qquad W_N := e^{-j2\pi/N}$$

und werden jeweils mittels einer doppelten `for`-Schleife implementiert, woher auch die schlechte Zeitkomplexität $\mathcal{O}(n^2)$ stammt.

Das Matlab-Skript ist im Anbei. Die Ergebnisse der Evaluierung einer 1024 Punkt Transformation sind

```
Results:
DFT: t=1.163559, err=4.22e-13
FFT: t=0.003367, err=3.19e-16
Forward FT: err=2.84e-10
Inverse FT: err=4.22e-13
```

mit der Fehlerrechnung

```
errDFT = rms(xInit - xIDFT); % Fehler zwischen echtem Signal und DFT rekonstruiertem
errFFT = rms(xInit - xIFFT); % Fehler zwischen echtem Signal und FFT rekonstruiertem
errFT  = rms(xFFT - xDFT);   % Fehler zwischen FFT und DFT
errIFT = rms(xIFFT - xIDFT); % Fehler zwischen IFFT und IDFT
```

2. Aufgabe

Fenstereffekt der DFT I (10 Punkte)

Es gibt zwei Fälle bei denen ein Zeitsignal mit der DFT exakt verarbeitet werden kann:

- das Signal ist zeit begrenzt
- das Signal ist periodisch und die DFT-Länge ist gleich (oder ein ganzzahliges Vielfaches)

der Periodendauer. Ist das nicht der Fall, dann treten bei der Verarbeitung mit der DFT Abweichungen auf. Diese Abweichungen werden als Fenstereffekte bezeichnet. In dieser Aufgabe sehen wir uns den so- genannten Leakage-Effekt genauer an. Wir betrachten als Signal die komplexe Exponentialschwingung

$$x[n] = e^{j\frac{2\pi}{M}n}$$

(a) Setzen sie die Definition von $x[n]$ in die Gleichung für die DFT ein und Vereinfachen Sie den Ausdruck so gut es geht. (Tipp: Benützen Sie die Formel für die endliche Geometrische Reihe

(b) Wir setzen nun $M = N$. Wie ist diese Wahl im Hinblick auf die oben genannten Fälle zu interpretieren?

(c) Zeigen Sie, dass für $k \neq 1$ $X[k] = 0$ ist. (Ausgangspunkt: Ergebnis von (b))

(d) Zeigen Sie mit der Regel von de L'Hospital, dass $X[k] = N$ für $k = 1$ ist. (Ausgangspunkt: Ergebnis von (b))

(e) Das Ergebnis lautet also $X[k] = N\delta(k - 1)$. Interpretieren Sie dieses Ergebnis

Aufgabe 2.

a.) $x[n] = e^{j\frac{2\pi}{M}\cdot n}$

$$X[k] = \sum_{n=0}^{N-1} x[n]\cdot e^{-j2\pi k\cdot\frac{n}{N}}$$

$$\sum_{n=0}^{N-1} e^{j\cdot\frac{2\pi}{M}\cdot n}\cdot e^{-j\cdot 2\pi M\cdot\frac{n}{N}}$$

$$\sum_{n=0}^{N-1} e^{j\cdot\frac{2\pi}{M}\cdot n - j2\pi M\cdot\frac{n}{N}}$$

$$\sum_{n=0}^{N-1} e^{j\cdot\left(\frac{2\pi}{M}\cdot n - M\cdot\frac{2\pi}{N}\cdot n\right)}$$

$$\sum_{n=0}^{N-1} e^{j2\pi n\cdot\left(\frac{1}{M}-\frac{k}{N}\right)}$$

Endliche geometrische Reihe

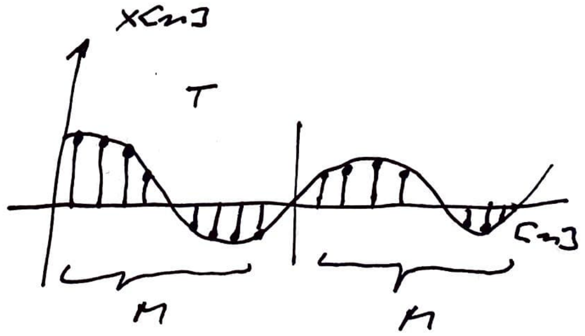
$$\sum_{n=0}^{p} q^n = \frac{1-q^{p+1}}{1-q}$$

$$\sum_{n=0}^{N-1} \Big(\underbrace{e^{j2\pi\left(\frac{1}{M}-\frac{k}{N}\right)}}_{q}\Big)^n$$

$$= \frac{1-e^{j2\pi\left(\frac{1}{M}-\frac{k}{N}\right)\cdot N}}{1-e^{j2\pi\left(\frac{1}{M}-\frac{k}{N}\right)}}$$

b.) N... Länge DFT

M... Signallänge



Betrachtung über eine Periode

Signal ist periodisch und die DFT-Länge ist gleich.

c.)

$$X[k] = \frac{1-e^{j2\pi(\frac{1}{M}-\frac{k}{N})\cdot N}}{1-e^{j2\pi(\frac{1}{M}-\frac{k}{N})}}$$

$$M=N \Rightarrow \frac{1-e^{j2\pi(1-k)}}{1-e^{j2\pi\frac{1}{N}\cdot(1-k)}}$$

$$e^{jx} = \cos(x) + j\cdot\sin(x) \rightarrow \text{Euler}$$

$$e^{j2\pi(1-k)} = \underbrace{\cos(2\pi(1-k))}_{1} + j\cdot\underbrace{\sin(2\pi(1-k))}_{0}$$

$k \ldots 1, 2, 3, 4 \ldots$ ganzzahlig $\rightarrow$ immer 1!

$$\rightarrow X[k] = \frac{1-e^{j2\pi(1-k)}}{1-e^{j2\pi\frac{1}{N}\cdot(1-k)}} \quad \begin{matrix}\rightarrow = 1\\ \rightarrow \neq 1\end{matrix}$$

d.)

$$X[k] = \frac{1-e^{j2\pi(1-k)}}{1-e^{j2\pi\frac{1}{N}\cdot(1-k)}}$$

$$\lim_{k\to 1} \frac{1-e^{j2\pi(1-k)}}{1-e^{j2\pi\frac{1}{N}\cdot(1-k)}} ; \frac{0}{0}$$

d'Hospital

$$\lim_{k\to 1} \frac{j2\pi\cdot e^{j2\pi(1-k)}}{\frac{1}{N}\cdot j2\pi\cdot e^{j2\pi\frac{1}{N}(1-k)}} = \frac{j2\pi}{\frac{1}{N}\cdot j2\pi} = N$$

e.)

$$\left.\begin{matrix} X[k] = 0 & \text{bei } k \neq 1 \\ X[k] = 1 & \text{bei } k = 1 \end{matrix}\right\} \delta[k-1]$$

bzw.

$$\left.\begin{matrix} X[k] = 0 & \text{bei } k \neq 0 \\ X[k] = 1 & \text{bei } k = 0 \end{matrix}\right\} \delta[k]$$

$X[k] = N \cdot \delta[k-1]$

- Spektrum hat 1-NICHT-NULL-WERT

Wert reicht aus um das Signal vollständig zu beschreiben.

$$x[n] = e^{j\frac{2\pi}{N}\cdot n} = \underbrace{e^{j\frac{2\pi}{N}\cdot n}}_{\substack{\text{unter der} \\ \text{Annahme } M=N}}$$

- Spektrum $X[K=1] = N$ sonst 0

$\Rightarrow f = \frac{1}{N}$

bei $X[K=2] = N$ sonst 0

$f = \frac{2}{N}$ ...

vocadijalskdfjkfjkdj

3. Aufgabe

Fenstereffekt der DFT II (10 Punkte)

In dieser Aufgabe überprüfen wir in Matlab was passiert, wenn keiner der in Aufgabe 2 genannten Fälle zutrifft (M und N teilerfremd). Dazu untersuchen wir ein Signal welches aus zwei benachbarten Kosinusschwingungen besteht. Dazu wird das eigentlich unendliche Signal einmal mit einem Rechteckfenster und einmal mit einem Hamming-Fenster zeit begrenzt. Wählen Sie:

```
N = 128;
n = 0:N-1;
w1 = 2*pi*0.1;
w2 = 2*pi*0.15;
x = cos(w1*n) + cos(w2*n);
```

Hier wurde `x` bereits (indirekt) mit einem Rechteckfenster gefenstert.

(a) Berechnen Sie das Spektrum von `x` und stellen Sie dessen Betrag grafisch dar (Befehle: `fft`, `abs`, `stem`). Vergessen Sie die Achsenbeschriftung nicht.

(b) Erzeugen Sie ein Hamming-Fenster mit `w = hamming(N).';` Generieren Sie damit das gefensterte Signal durch elementweises Multiplizieren von w mit `x` (Befehl: `w.*x`). Berechnen Sie davon das Spektrum und stellen Sie dessen Betrag grafisch dar.

(c) Interpretieren Sie die Ergebnisse von (a) und (b).

(d) Experimentieren Sie mit den Parametern w_1 und w_2 und finden Sie Einstellungen bei denen die DFT das exakte Ergebnis liefert. Erklären Sie auch, wieso mit den gewählten Einstellungen das DFT Ergebnis exakt ist.

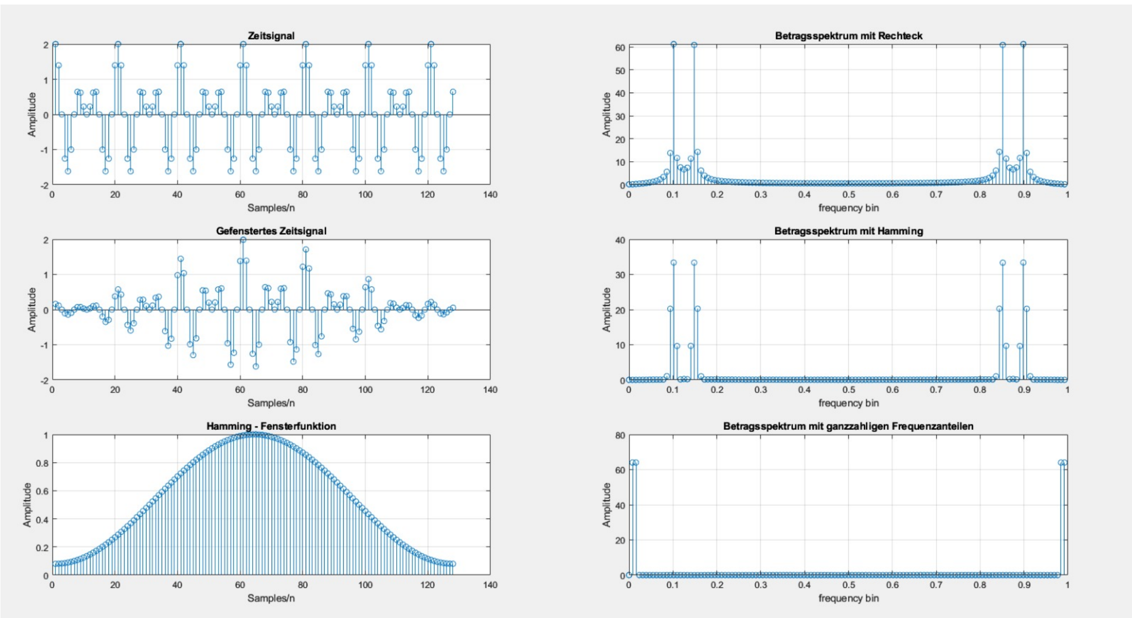
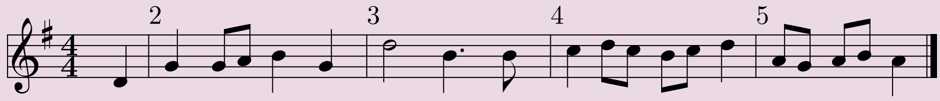


Abbildung 1: Vergleich mit gefenstertem Spektrum

4. Aufgabe

Audiosignal-Synthese (10 Punkte)

In diesem Versuch soll die Verbindung zwischen den digitalen Signalen am PC und der *realen* Welt mit Hilfe von Audiosignalen hörbar gemacht werden. Dabei lernen Sie auch ein Beispiel für ein etwas komplizierteres MATLAB-Programm kennen. Es soll ein Musikstück vertont werden. Grundlage ist die Zeit-Frequenz-Darstellung im untenstehenden Notenblatt. Horizontal ist der zeitliche Verlauf und vertikal die Frequenzlage angegeben.



Die daraus resultierende Abfolge der Töne mit den zugeordneten Zeitdauern bezogen auf ein Grundintervall ist in der Tabelle unten zusammengestellt.

Note	d	g	g	a	h	g	d’	h	h	c’
Dauer	1/4	1/4	1/8	1/8	1/4	1/4	1/2	3/8	1/8	1/4
Note	d’	c’	h	c’	d’	a	g	a	h	a
Dauer	1/8	1/8	1/8	1/8	1/4	1/8	1/8	1/8	1/8	1/4

Der Zusammenhang zwischen den Noten und der physikalischen Signaldarstellung, d.h. die Frequenzlage, erschließt sich aus den in der Musik bekannten Beziehungen:

- der Kammerton **a’** entspricht einem Sinuston mit 440Hz, d.h. **a** entspricht 220Hz
- eine Oktave, z.B. der Übergang von **a** zu **a’**, umfasst eine Frequenzverdopplung
- in einer Oktave gibt es 12 Halbtonschritte

Daraus ergibt sich die Frequenzzuordnung der *C-Dur*-Tonleiter ($\mathsf{F} \cdot 220$Hz)

Note	c	d	e	f	g	a	h/b	c’
Frequenzfaktor F	$2^{-\frac{9}{12}}$	$2^{-\frac{7}{12}}$	$2^{-\frac{5}{12}}$	$2^{-\frac{4}{12}}$	$2^{-\frac{2}{12}}$	1	$2^{\frac{2}{12}}$	$2^{\frac{3}{12}}$

Mit diesen Festlegungen kann nun jeder Note ein Sinuston entsprechender Frequenz und Dauer zugeordnet werden. (Im Englischen wird für die deutsche Note **h** der Buchstabe **b** verwendet und die Tonhöhe **pitch** genannt. Durch das Vorzeichen # auf der 5. Linie von unten wird der Ton **f** um einen Halbton zum **fis** erhöht.)

- Machen Sie sich mit dem Programm `audiosynth.m` vertraut und starten Sie dieses.
- Das mit dem Programm erzeugte Audiosignal klingt unnatürlich, da es nur aus jeweils ein- und ausgeschalteten Sinustönen besteht. Ein besserer Höreindruck lässt sich mit Hilfe einer Hüllkurvenbewertung erzielen. In der Audiotechnik wird hierfür oft das sogenannte ADSR-Profil (siehe Abbildung 4) verwendet. Es besteht aus vier Geradenstücken, die die vier Phasen **Attack** (A), **Decay** (D), **Sustain** (S) und **Release** (R) repräsentieren.

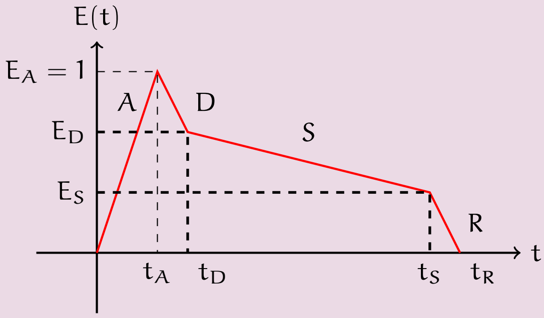


Abbildung 4: ADSR-Hüllkurve

(a) Machen Sie sich mit der Funktion `adsr_profile.m` vertraut und modifizieren Sie in der Folge das Programm `audiosynth.m`, indem Sie die Sinustöne mit dem ADSR-Profil bewerten (mögliche Parameterwahl: $t_A = 0.15$, $t_D = 0.25$, $t_S = 0.9$, $E_D = 0.9$, $E_S = 0.7$).

(b) Erweitern Sie das Programm um die grafische Ausgabe des Musikstücks im Zeitbereich.

(c) Probieren Sie verschiedene Einstellungen für die Abtastfrequenz f_s, die Zeitskalierung T sowie für die Parameter des ADSR-Profils aus. Kommentieren Sie Ihre Versuche kurz im Protokoll.

(d) Der Höreindruck lässt sich noch weiter verbessern, wenn Sie zusätzliche harmonische Anteile, z.B. jeweils bei der doppelten, dreifachen, etc... Frequenz, hinzunehmen. Dabei können Sie die Amplituden der Frequenzkomponenten unterschiedlich gewichten. Experimentieren Sie mit verschiedenen Einstellungen bis Sie einen zufriedenstellenden Höreindruck erhalten. Kommentieren Sie Ihre Versuche kurz im Protokoll.

(e) Erzeugen Sie eine "WAVE"-Datei aus Ihrem finalen Musikstück. Verwenden Sie dazu den MATLAB-Befehl `audiowrite`. Diese Datei soll ebenfalls abgegeben werden.

(f) Mit dem Programm `kurzanalyse.m` können Sie eine Kurzzeitspektralanalyse einer Wave-Datei durchführen. Das Programm öffnet zunächst eine Audio-Datei im `wav`-Format und führt eine FFT über *die gesamte Signallänge* durch. Signal und Spektrum werden dargestellt. Im zweiten Teil wird das Signal in überlappende Blöcke eingeteilt und jeder Block für sich transformiert.

   (i) Rufen Sie das Programm auf und lesen Sie die Musikdatei ein, die Sie im vorigen Punkt erzeugt haben. Wählen Sie die FFT-Länge zu $M = 4096$ und den Parameter Overlap $OL = 2$. Das FFT-Fenster wird dann jeweils um $4096/OL$ Folgenelemente weitergeschoben. Die Beträge der resultierenden Kurzzeitspektren werden zusammen als Spektrogramm angezeigt.

   (ii) Rotieren Sie das Spektrogramm, so dass Sie den Frequenzinhalt über der Zeit schön beobachten können. Vergleichen Sie das Bild mit der Notenzeile. Was erkennen Sie?

   (iii) Variieren Sie den Parameter der FFT-Länge (z.B. 256, 512, ...) und erklären Sie, was Sie in Bezug auf die Frequenz- und Zeitauflösung feststellen können.

(a) Modifikation des Programms

Nach der Berechnung des Sinustons wird dieser in der `adsr_profile` Funktion verarbeitet und an das Musikstück angehängt.

```
% ...
s = sin(Omega*n);           % tone
sAdsr = adsr_profile(s);
music = [music sAdsr];      % concatenated audio signal
% ...
```

(b) Grafische Ausgabe

Für die Grafische Ausgabe wurden, um das Signal besser ablesen zu können, die Segmente je nach Frequenz unterschiedlich eingefärbt.

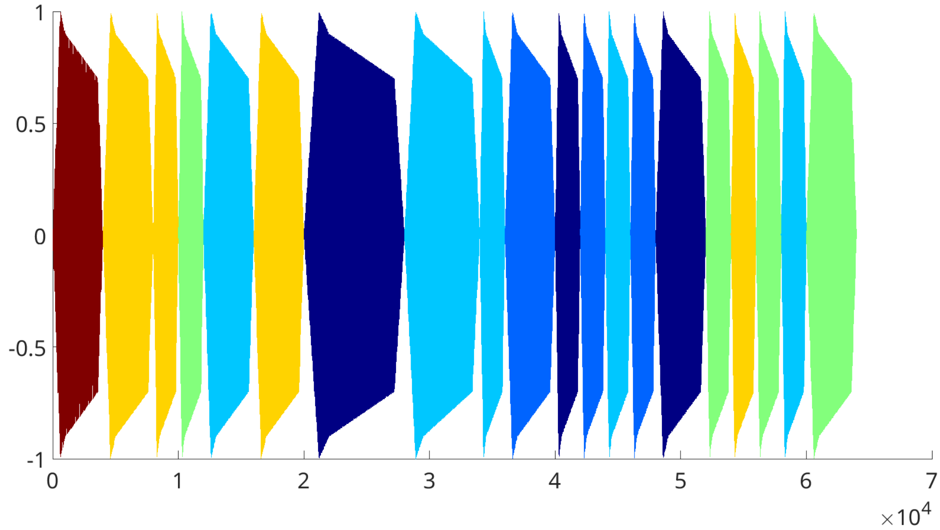


Abbildung 2: Hüllkurven bewertetes Musiksignal

(c) Sampling Parameter

Die Abtastfrequenz und die Tondauer stehen bei der Verarbeitung des Signals im Verhältnis zur Anzahl der Samples für jeden Ton. Die Gesamtlänge des Signals wird damit länger, wenn

- die Sample-Rate vermindert, oder
- die Tondauer verlängert wird.

Die Zeiten der ADSR-Hüllkurve sind relativ zur Gesamtlänge angegeben. Ein Veränderung dieser Werte hat zur folge, dass die Einhüllende von jedem Ton eine andere Form hat, wie in Abbildung 4 gezeigt. Dabei ist zu beachten, dass $t_A + t_D + t_S < 1$. Die Release-Zeit ist dann die verbleibende Zeit nach dem der Sustain abgelaufen ist.

(d) Harmonische Anteile

Als Ziel wurde hier gewählt, den Ton einem Sägezahn anzunähern. Das kann erreicht werden, in dem weitere Sinusschwingungen mit einer ganzzahligen vielfachen Frequenz der Grundschwingung aufaddiert werden. Der Dämpfungsfaktor $1/i$ sorgt dafür, dass höhere Harmonische das Signal nicht verwüsten.

```
nHarmonics = 12;

for k=1:length(pitch)
    % ...
    s = 0;
    for i = 1:nHarmonics
        s = 1/i * sin(i*Omega*n) + s;
    end
    % ...
end
```

An einem stark vergrößertem Bild des Signals kann man das Sägezahn erkennen.

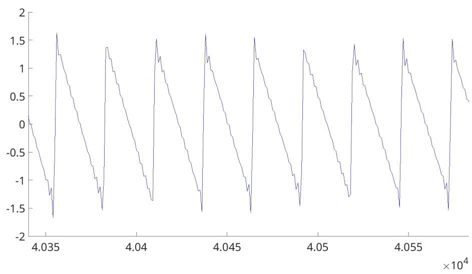


Abbildung 3: Sägezahn Signal

(f) Short-Time-Fourier-Transform

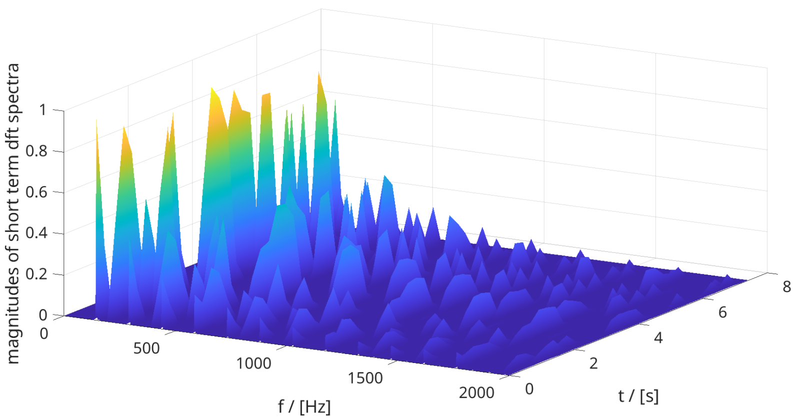


Abbildung 4: 3D STFT

(ii) Vergleich Notenblatt

Speziell im Frequenzbereich der Grundschwingungen kann die Notenfolge gut erkannt werden.

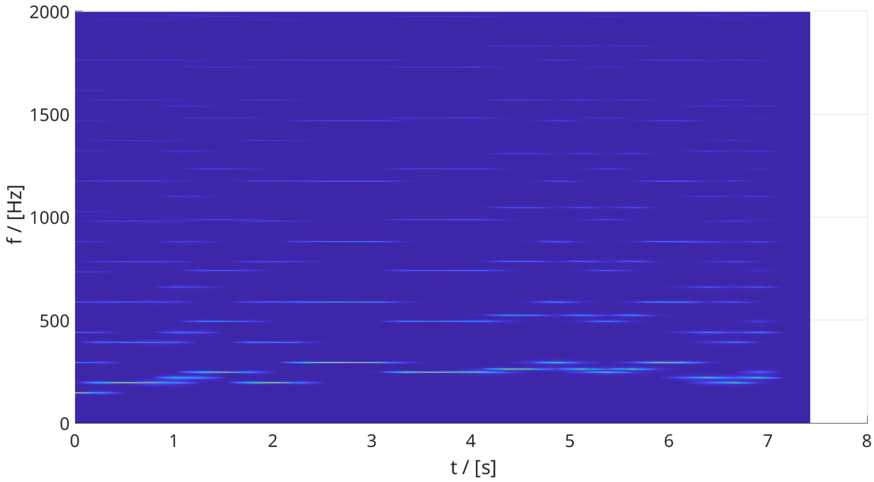


Abbildung 5: STFT von Unten

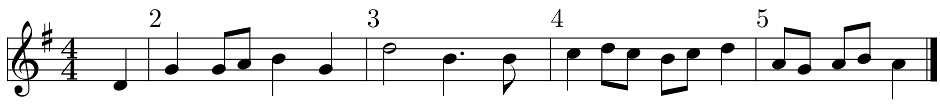


Abbildung 6: Notenblatt zum Vergleich

(iii) FFT Länge reduzieren

Mit einer kleineren FFT-Länge wird es zunehmen schwieriger, die genaue Frequenz des Signals zu ermitteln - die Amplitude verschmiert. Im Gegenzug werden jedoch die zeitlichen Übergänge der Töne schärfer.

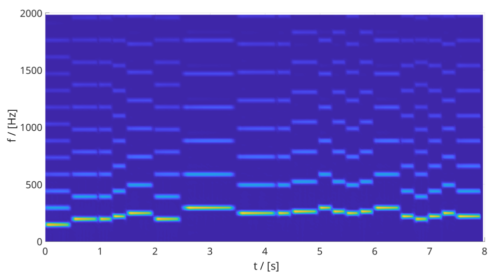


Abbildung 7: STFT 512

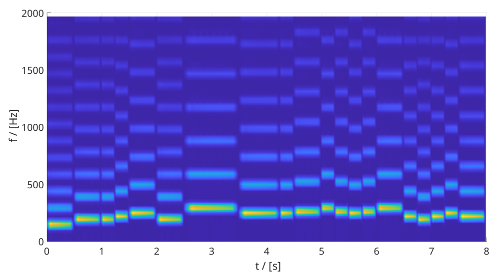


Abbildung 8: STFT 256

Dieser Zusammenhang ist durch das Zeit-Bandbreite-Produkt gegeben [1, p. 41]:

$$\Delta t \cdot \Delta f = \text{const}, \quad \text{const} \geq \frac{1}{4\pi}$$

Anhang

Literatur

[1] Daniel Ch. von Grünigen. *Digitale Signalverarbeitung : mit einer Einführung in die kontinuierlichen Signale und Systeme ; mit 91 Beispielen, 80 Aufgaben sowie einer CD-ROM mit Lösungen sowie Entwurfs- und Simulationsprogrammen.* Fachbuchverl. Leipzig im Carl-Hanser-Verl, 4. aufl. edition, 2008.

Feedback

1 Bitte Funktion wie in Angabe nennen (`eval_dft_vs_fft`)

3.a,b auf der x-Achse sollte k von 0-128 gehen (entsprechend der Beschriftung 'frequency bin') -0,5P;

3.c fehlt -3P;

3.d Erklärung fehlt -1P;

4 unklare Erklärung der Abtastfrequenz, Zeitskalierung (vermindern der Sample Rate erhöht nicht die Gesamtdauer des Signals - bei zu niedriger Abtastfrequenz wird das Abtasttheorem verletzt.) -0,5P;