

Signal Analyse

Hausaufgabe 3

Hannes Reindl
01532129

March 6, 2020

Beispiel 2: STFT Analysis

Task 1: Manuelles decodieren einer Nachricht

In der ersten Aufgabenstellung soll mithilfe der Funktion `signalgenerator2(id)` eine codierte Nachricht erstellt und mithilfe des Matlab Befehls `spectrogram` analysiert werden.

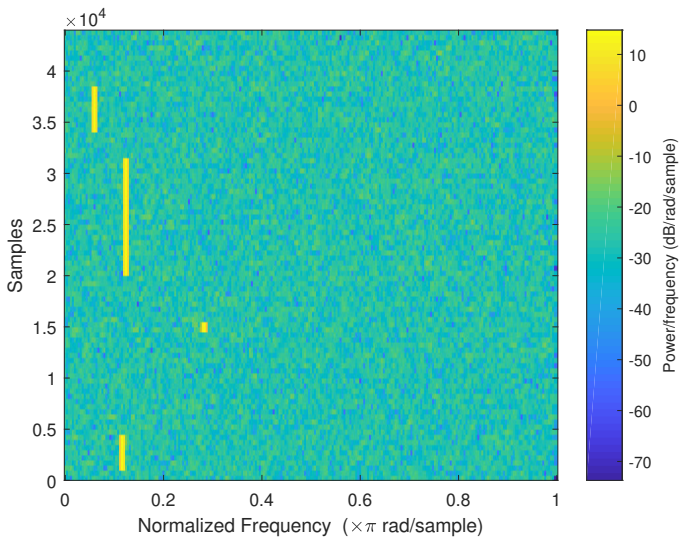
```
spectrogram(x>window,noverlap,nfft,fs)
```

Für die Fensterlänge wurde `window = 500` gewählt, da "Encryption Length/Pause" in der gegebenen Tabelle auch immer ein vielfaches von 500 Samples war. Dadurch sollte gewährleistet sein, dass kein Symbol mittendrin abgeschnitten wird und die Heisenberg Boxen entsprechen in der Zeit genau den Übergängen. Bei längeren Fensterlängen könnten kurze Symbole in der Zeit nicht mehr gut aufgelöst werden. Bei kleineren Fensterlängen verschlechtert sich die Frequenzauflösung.

`noverlap` wurde auf 0 gesetzt und `nfft` gleich der Fensterlänge. (Wäre `noverlap` ungleich null, stimmen die Übergängen mit den Heisenberg Boxen nicht mehr zusammen)

Task 1: Manuelles decodieren einer Nachricht

spectrogram(xn, 500, 0, 500)



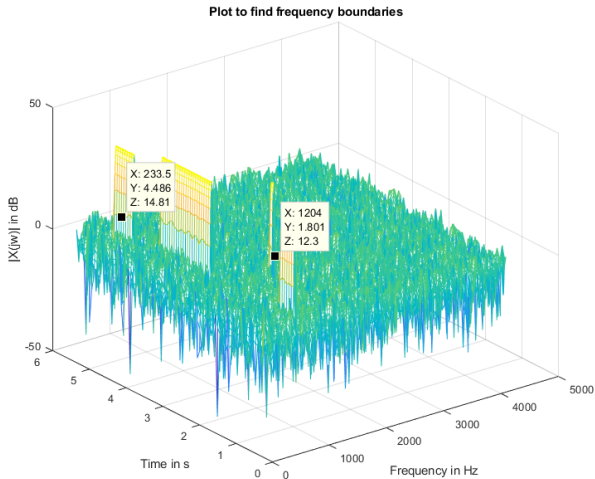
Task 1: Manuelles decodieren einer Nachricht

Mithilfe der Cursor Funktion wurde die Kennwerte ermittelt:

First symbol: $f = 0.12 \cdot f_S/2 = 491 \text{ Hz} \rightarrow$	" "
$t_0 = 4250 - 750 = 3500 \rightarrow$	"z"
$t_1 = 14250 - 4250 = 10000 \rightarrow$	"w"
Second symbol: $f = 0.288 \cdot f_S/2 = 1176 \text{ Hz} \rightarrow$	"e"
$t_0 = 1.525E4 - 1.425E4 = 1000 \rightarrow$	"i"
$t_1 = 1.975E4 - 1.525E4 = 4500 \rightarrow$	" "
Third symbol: $f = 0.128 \cdot f_S/2 = 524 \text{ Hz} \rightarrow$	"a"
$t_0 = 3.125E4 - 1.975E4 = 11500 \rightarrow$	"c"
$t_1 = 3.375E4 - 3.125E4 = 2500 \rightarrow$	"h"
Fourth symbol: $f = 0.064 \cdot f_S/2 = 262 \text{ Hz} \rightarrow$	"t"
$t_0 = 3.825E4 - 3.375E4 = 4500 \rightarrow$	" "
$t_1 = 4.375E4 - 3.825E4 - 1000 = 4500 \rightarrow$	" "

Task 2: Bandbreite

Aus Plot gelesen: Bandbreite = $1204 - 233.5 = 970.5$ Hz



Task 3: Encoder

Nun sollen die zwei gegebenen strings

```
ghtqhtljtlhtxtnyaayhtxtnxht ht ht htaaa
```

```
xmmjmmymmktnkkm mmatnokmjmmymmommztnzkmkmm tnakm
```

entsprechend der Anleitung im PDF codiert werden. Dazu musste zuerst die Tabelle vervollständigt werden. Mithilfe der Funktion `codingtest()` wurde für jedes Zeichen ein Symbol generiert und dann mit der `get_parameters()` die Parameter berechnet.

Task 3: Encoder

Näheres zur Funktion: `get_parameters()`

Diese berechnet für das übergebene Symbol das Spektrogramm zweimal. Einmal mit guter Zeitauflösung und einmal mit guter Frequenzauflösung. Zuerst werden die Zeitpunkte ermittelt, bei denen ein Symbol beginnt, endet und von HIGH auf LOW wechselt. Aus diesen wird dann die Zeit in Samples für t_0 und t_1 berechnet. Danach wird entsprechend den ermittelten Zeiten das Signal in Teilabschnitten betrachtet und für jeden Teilabschnitt die dominierten Frequenz ermittelt (über arithmetisches Mittel). Rückgabeparamter ist ein struct mit den Kinder `s0`, `s1` (da Samples und nicht Zeit anderer Variablen Name), sowie `freq` welche die Frequenz enthält. Dieses struct hat so viele Elemente wie Symbole im Signal sind.

Task 3: Encoder

Character	Frequency	Length	Character	Frequency	Length
-	Hz	Samples	-	Hz	Samples
a	524,3	5000	o	585,7	5500
b	1568,8	10500	p	348,2	3000
c	1974,3	11500	q	987,1	8000
d	2793,5	13500	r	2093,1	12000
e	1175,6	9000	s	1761,3	11000
f	2349,1	12500	t	262,1	1500
g	1046,5	8500	u	2637,8	13000
h	329,7	2500	v	1318,9	9500
i	245,8	1000	w	1396,7	10000
j	698,4	6500	x	782,3	7000
k	438,3	4000	y	659,5	6000
l	880,6	7500	z	393,2	3500
m	294,9	2000	(space)	495,6	4500
n	221,2	500			

Task 3: Encoder

Mithilfe dieser Tabelle wurde eine weitere Funktion geschrieben (`own_encoder`), welche die gegebenen Strings codiert und abspielt.

1 String: Helene Fischer - Atemlos

2 String: Udo Jürgens - Ich war noch niemals in New York

Task 4: Bandbreite

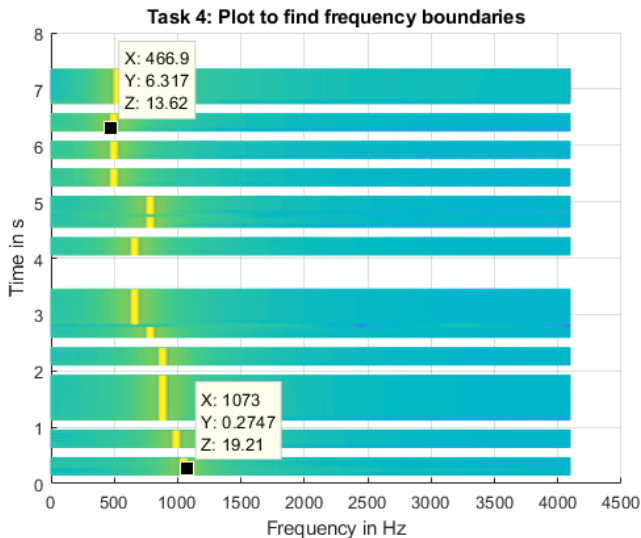
Bandbreite wieder über Plot ähnlich wie in (B) ermittelt.

soundstring1: Bandbreite = $1073 - 467 = 606 \text{ Hz}$

soundstring2: Bandbreite = $811 - 364.5 = 446.5 \text{ Hz}$

Task 2: Bandbreite

soundstring1



Task 2: Bandbreite

soundstring2

