



BEUTH HOCHSCHULE  
FÜR TECHNIK  
BERLIN  
University of Applied Sciences

---

# DTMF-Detektor mit Goertzel-Algorithmus

## Laborbericht

angefertigt von

Robby Kozok, Nic Frank Siebenborn, Pascal Kahlert

in dem Fachbereich VII – Elektrotechnik - Mechatronik - Optometrie –  
für das Modul Digitale Signalverarbeitung III  
der Beuth Hochschule für Technik Berlin im Studiengang  
**Elektrotechnik - Schwerpunkt Elektronische Systeme**

Datum 7. Februar 2016

### Lehrkraft

Prof. Dr.-Ing Marcus Purat    Beuth Hochschule für Technik

---

# Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitung</b>	<b>2</b>
<b>2</b>	<b>Goerzel Algorithmus</b>	<b>3</b>
2.1	Aufgabenstellung . . . . .	3
2.2	Durchführung . . . . .	3
2.3	Auswertung . . . . .	4

# Kapitel 1

## Vorbereitung

$$h_k(n) = \sigma(n) * (\cos(k(n+1)\frac{2 * \pi}{N}) + j \sin(k(n+1)\frac{2 * \pi}{N}) = h_k^{RE}(n) + j h_k^{IM}(n) \quad (1.1)$$

Ergebnis der rechtsseitigen z-Transformation.

Imaginärteil:

$$X_{R_{sin}}(z) = \frac{z^2 * \sin(\frac{\pi}{N} * k)}{z^2 - 2 * \cos(\frac{\pi}{N} * k) * z + 1} \quad (1.2)$$

Realteil:

$$X_{R_{cos}}(z) = \frac{z^2 * \cos(\frac{\pi}{N} * k) - z}{z^2 - 2 * \cos(\frac{\pi}{N} * k) * z + 1} \quad (1.3)$$

---

# Kapitel 2

## Goerzel Algorithmus

In dem folgenden Kapitel wird die FFT ohne Fensterung untersucht.

### 2.1 Aufgabenstellung

In dieser Aufgabe sollten der Goertzel und FFT Algorithmus verglichen werden. Im ersten Schritt sollten bei Ergebnisse für die 8 DTMF Frequenzen verglichen werden und im zweiten Schritt sollte betrachtet werden, welche der beiden Implementierungen einen höheren Rechenaufwand besitzt.

### 2.2 Durchführung

Bevor der Goertzel Algorithmus genutzt werden kann müssen seine Koeffizienten berechnet werden. Dies haben wir in einem Matlab Script (s. Code Listing 2.1) realisiert.

```
1 dtmf_697 = 59;
2 dtmf_770 = 66;
3 dtmf_852 = 73;
4 dtmf_941 = 80;
5
6 dtmf_1209 = 103;
7 dtmf_1336 = 114;
8 dtmf_1477 = 126;
9 dtmf_1633 = 139;
10
11 a(1) = cos((pi/2048)*dtmf_697) * 2^15;
12 b(1) = sin((pi/2048)*dtmf_697) * 2^15;
13
14 a(2) = cos((pi/2048)*dtmf_770) * 2^15;
15 b(2) = sin((pi/2048)*dtmf_770) * 2^15;
16
17 a(3) = cos((pi/2048)*dtmf_852) * 2^15;
18 b(3) = sin((pi/2048)*dtmf_852) * 2^15;
19
20 a(4) = cos((pi/2048)*dtmf_941) * 2^15;
21 b(4) = sin((pi/2048)*dtmf_941) * 2^15;
22
23 a(5) = cos((pi/2048)*dtmf_1209) * 2^15;
24 b(5) = sin((pi/2048)*dtmf_1209) * 2^15;
25
26 a(6) = cos((pi/2048)*dtmf_1336) * 2^15;
27 b(6) = sin((pi/2048)*dtmf_1336) * 2^15;
```

---

```

28
29 a(7) = cos((pi/2048)*dtmf_1477) * 2^15;
30 b(7) = sin((pi/2048)*dtmf_1477) * 2^15;
31
32 a(8) = cos((pi/2048)*dtmf_1633) * 2^15;
33 b(8) = sin((pi/2048)*dtmf_1633) * 2^15;

```

Listing 2.1: Berechnung der Goertzel Koeffizienten

In Zeile 1 bis 9 definieren wir die Positionen der DTMF Frequenzen. In den folgenden Zeilen berechnen wir für jeden Goertzel Filter die Koeffizienten. Im Source Code ergibt sich dadurch ein Array wie in Listing 2.2 dargestellt.

```

1      fract16 coeff[16] =
2      {
3          //ak      bk
4          32634, 2962,
5          32600, 3312,
6          32563, 3662,
7          32522, 4011,
8          32360, 5156,
9          32268, 5701,
10         32158, 6294,
11         32026, 6934
12     }

```

Listing 2.2: Array der Goertzel - Koeffizienten

Im Vergleich zur Übung 4 musste nun anstatt des FFT Algorithmus der Goertzel Algorithmus genutzt werden.

Der Rechenaufwand wurde wie in Übung 2 bestimmt.

## 2.3 Auswertung

Der Goertzel-Algorithmus lieferte uns nicht das erwartete Ergebnis. Wir erwarteten genau ein eindeutiges Maximum, dies erhielten wir aber nicht. Deswegen wird der DTMF Detektor im weiteren nicht mehr betrachtet.

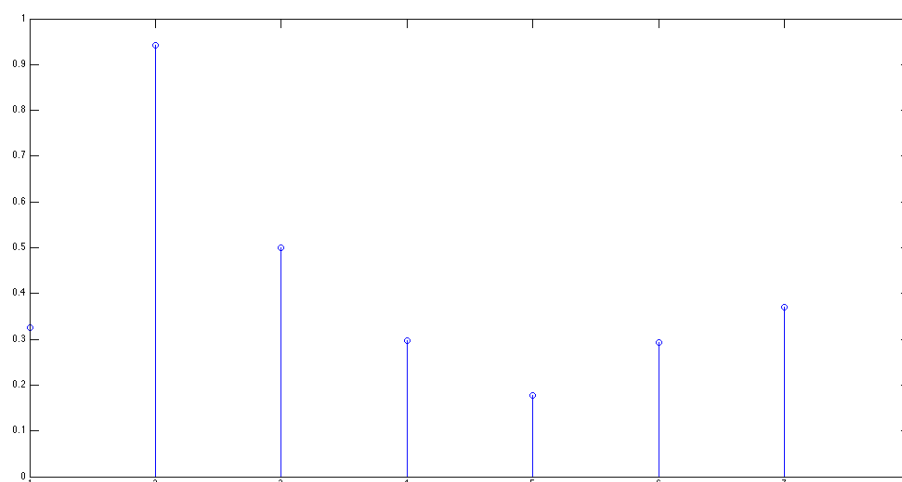


Abbildung 2.1: Absolute Beträge der Ergebnisse des Goertzel-Algorithmus



In Figure 2.1 ist zu sehen, dass jeweils der 2 und 8 Werte im Vergleich zu den anderen relativ hoch ist, da wir aber nur einen hohen Wert erwarteten (sehen wir Absprache mit Herrn Purat) davon aus, dass der Goertzel-Algorithmus nicht wie erwartet funktioniert. Dahingegen ist in Figure 2.2 zu sehen, dass es ein ganz klares Maximum gibt.

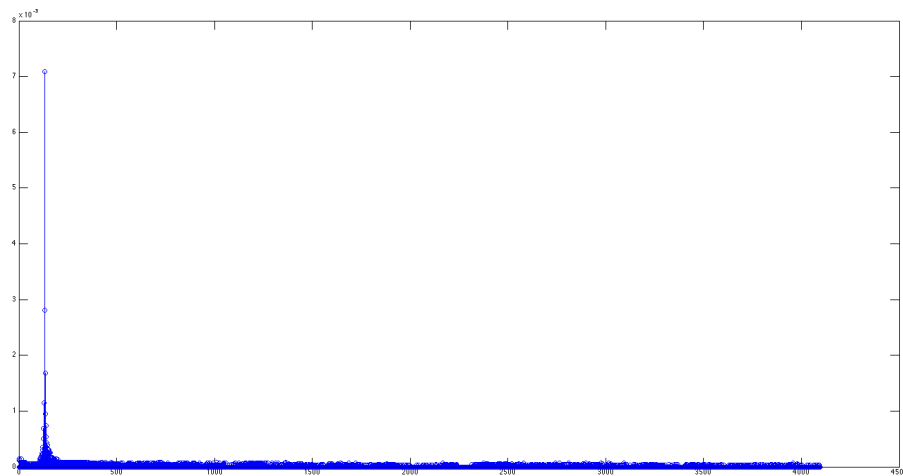


Abbildung 2.2: Absolute Beträge der Ergebnisse des FFT-Algorithmus

Im weiteren soll der Rechenaufwand beider Implementierungen verglichen werden.

	Goertzel-Algorithmus	FFT-Algorithmus
Rechenaufwand in Maschinenzyklen	1524110	3727089

Die Tabelle zeigt, dass der Goertzel-Algorithmus deutlich schneller ist. In unserem Beispiel um einen Faktor von 2,4. Dies liegt daran, dass der Goertzel Algorithmus nur noch die Spektralwerte an genau den Stellen der DTMF Frequenzen errechnet und nicht wie die FFT über einen gesamten Block von 4096 Werten.

# Abbildungsverzeichnis

2.1	Absolute Beträge der Ergebnisse des Goertzel-Algorithmus . . . . .	4
2.2	Absolute Beträge der Ergebnisse des FFT-Algorithmus . . . . .	5