

## HPC - Lab 01

Aubry Mangold  
ISCS, HEIG-VD  
aubry.mangold@heig-vd.ch

### Program

The DTMF Encoder/Decoder program is designed to encode and decode Dual-Tone Multi-Frequency (DTMF) signals, which are commonly used in telecommunication signaling. The program is implemented in C and relies on several external libraries, including `libfftw3` for Fast Fourier Transform (FFT) operations and `libsndfile` for handling audio file input and output.

See the program's README file for build and run instructions.

### Benchmarking

The program was benchmarked in both its version by measuring the decoding time for messages of size 10, 100, 1000 and 10000 characters using the Unix `time` command. The content of the messages was randomly generated using an alphabet.

The benchmarking was done on an AMD Ryzen 7 PRO 6850U (16 core @ 4.77 GHz) processor using 32 GB of RAM on Fedora Linux 41 using kernel 6.12.15. The program was compiled using GCC 14.2 and the `-O3` optimization flag.

The benchmark results are presented in tabular format in Table 1 and plotted in Figure 1.

Size	Goertzel Decoding Time (ms)	FFT Decoding Time (ms)
10	20	0
100	170	40
1000	1760	420
10000	17110	3900

Table 1: Benchmark results for DTMF decoding

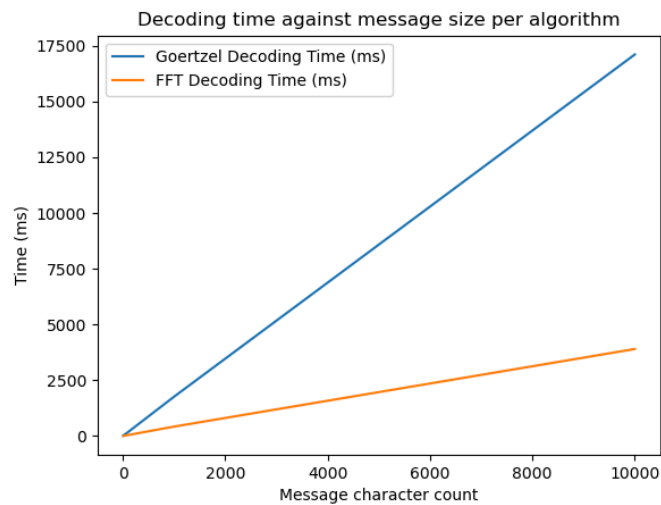


Figure 1: Plotted benchmark results for DTMF decoding

## Conclusion

The benchmark results show that the decoding time increases linearly with the size of the message, as expected. The Goertzel algorithm implementation is slower than the FFT-based implementation for all message sizes, indicating that the FFT-based implementation is more efficient.

The difference in performance between the two implementations is less pronounced for larger message sizes, indicating that the overhead of the Goertzel algorithm becomes less significant as the message size increases.

This result is in contrast to the theoretical expectation that the Goertzel algorithm should be faster than the FFT-based implementation for small message sizes. The discrepancy may be due to the specific implementation details of the Goertzel algorithm as it is entirely implemented by hand in C, whereas the FFT-based implementation relies on the highly optimized `libfftw3` library.

## Further work

- Improve the Goertzel algorithm implementation
- Improve the overall clarity of the code
- Average the benchmark results over multiple runs