Interference Transform: Estimating the frequency and phase of low resolution samples

Carlos Tarjano Valdecy Pereira

May 18, 2020

Abstract

The Fourier Transform, naturally due to the context and time in which it was idealized, was formulated without regard to discrete waves or computational complexity. Some developments, such as the Discrete Fourier Transform and the Fast Fourier Transform algorithm updated the theory to the digital, discrete era. Some shortcomings were introduced with those methods, such as diminishing resolution in dealing with a small number of os samples. In this paper we propose a transform specifically designed for the discrete case, optimizing the inference of frequency and amplitude from whatever sampled points are available. The algorithm is based on the representation of the set of samples in a frequency versus phase space and the investigation of the interference pattern thus generated. Each sample defines a sinusoidal planar wave in this space, with a particular amplitude, direction and frequency. The determination of the point in which maximum constructive interference occurs translates to the determination of the most accurate sinusoidal description of the whole set of samples. TODO: improvement of the naïve algorithm. better accuracy TODO: an alternative to wavelet and short-time Fourier transform

Keywords: Fast Fourier Transform, Digital Signal Processing, Wavelet Transform

1 Introduction

2 Literature Review

3 Methodology

3.1 Theory

One can represent the discrete wave as an an array $W[i]$ of $n-1$ real numbers. Such wave is related, as illustrated in figure 1, to an hypothetical continuous wave $S(t)$ of which its components are sampled in uniform time intervals at a rate of fps frames per second.

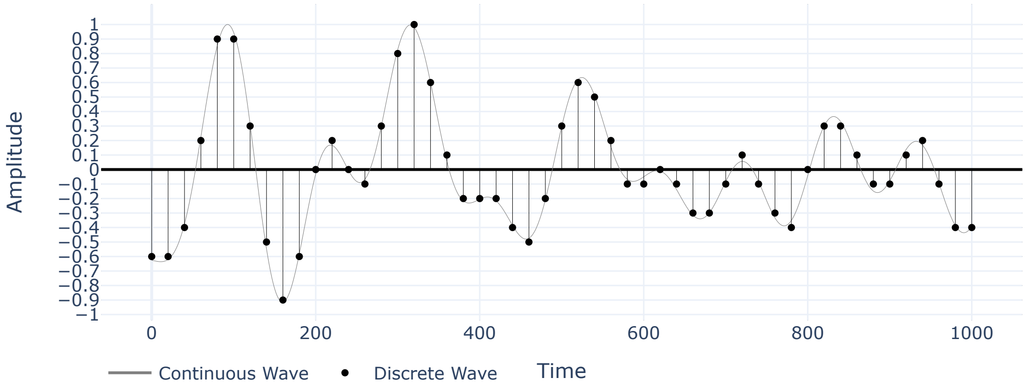

Figure 1: Discrete samples (black points) from a continuous wave (grey line). Note that the amplitude is also truncated

Thus, we have: $W[i] \in \mathbb{R} \forall i \in \mathbb{N}, 0 \leq i < n$, $S(t) \in \mathbb{R} \forall t \in \mathbb{R}, 0 \leq t < t_f$ where $t_f = n/\text{fps}$

4 Results

5 Discussion + Conclusion

6 References