MUMT605 Assignment 2

Johnty Wang

14 Nov 2014

Part 1

1 Given,

$$\omega[k] = \frac{ArgX[t,k] - ArgX[s,k] + 2\pi p}{H}$$

(Intuitive meaning: frequency is given by the phase difference at two time points, plus a potential multiple of 2π, divided by time between the two points in which this phase difference occurred)

When $t - s = 1 = H$, we have:

$$\omega[k] = ArgX[t,k] - ArgX[s,k] + 2\pi p$$

Now, since we're dealing with a sampled signal, we must have

$\omega[k] < \pi$ to satisfy the Nyquist criterion and since $t - s = 1$, it is impossible for a signal below this limit to increment more than 2π in the timespan of **one single sample** and still be adequately represented without aliasing, so the only possible value of p is *zero*.

2 As described in the paper, we need a hop size $H \leq \frac{N}{2K}$ where K is the bandwidth (in samples) of the window and N is the window size (in samples). Putting in $t - s = H$ and $C_w = K$ and M for window size gives

$$H \leq \frac{M}{2C_w}$$

(note: I'm not totally certain the definition of C_w, I assume its the same as 'K' based on my interpretation of notes taken in class)

This condition can also be satisfied when $t-s$ is 2π multiples of a given frequency (for that frequency).

3 Since $H = u_r - u_{r-1}$ and is the amount of time that a particular frequency $\omega(u_r, k)$ will have evolve for since the previous phase $ArgY(u_r, k)$, we have simply:

$$\lambda(u_r, k) = ArgY(u_r, k) + H\omega(u_r, k)$$

4 We still need to maintain the bandwidth-related limit from 2.) above, and in the stretched case, we essentially have a larger gap in the hop given by α. Hence, we have the following:

$$\alpha t_r - s_r \leq \frac{M}{2C_w}$$

5

6 Looking at the original equation with $t_{r-1} - s_r \neq 1$ (in other words, $H \neq 1$:

$$\omega[k] = \frac{ArgX[t,k] - ArgX[s,k] + 2\pi p}{t_r - s_r}$$

7

1. For a given function

$$y(t) = \begin{cases} x(t) & t \geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

Then, based on the definition of $| \, |$:

$$y(|t|) = \begin{cases} x(t) & t \geq 0 \\ x(-t) & t < 0 \end{cases}$$

Therefore, for this particular example, we have:

$$x(t) = s(t) + s(-t)$$

from the property of the fourier transform:

$$\begin{aligned} \text{if } a(t) &\rightarrow A(f) \\ \text{then } a(-t) &\rightarrow A(-f) \\ \\ \text{so } x(t) &\rightarrow X(f) \\ \implies X(f) &= S(f) + S(-f) \\ \implies X(f) &= \frac{1}{\alpha - 2\pi j f} + \frac{1}{\alpha + 2\pi j f} \\ &= \frac{\alpha + 2\pi j f + \alpha - 2\pi j f}{(\alpha - 2\pi j f)(\alpha + 2\pi j f)} \\ &= \frac{2\alpha}{\alpha^2 + 4\pi f^2} \end{aligned}$$

2. When we sample the spectrum $X(f)$ of the time series signal $x(t)$ using a dirac comb, it creates a "periodized" version of the signal, which can be expressed as:

$$\begin{aligned} & X'(f) = X(f)\text{Ш}_{T_0}(f) \\ & \text{where} \\ \text{Ш}_{T_0}(f) &= \sum_{k=-\infty}^{\infty} \delta(f - kT_0) \\ \implies X'(f) &= \frac{2\alpha}{\alpha^2 + 4\pi f^2} \sum_{k=-\infty}^{\infty} \delta(f - kT_0) \end{aligned}$$

3. From the above, we can see that $X'(f)$ is nonzero where $f = kT_0$, which means the amplitude of the kth harmonic can be expressed as:

$$\frac{2\alpha}{\alpha^2 + 4\pi k^2 {T_0}^2}$$

4. One advantage of this method is that the calculation is very simple - by exploiting the additive and time reversal properties of the fourier transform, we did not need to evaluate the integral for the |t| case.

Part 2

Overview

For this part, I implemented a matlab function called `A2_func` in Matlab, with its help/description as follows:

```
% code here
% code here
```

The internal code comments provide explanation of the process, but the overall process is as follows:

- Compute discrete spectrum
- Generate frequency independent wavetable with correct duty cycle from spectrum, that is otherwise independent of sample rate
- Produce frequency correct wavetable given sample rate, target frequency and fill up entire buffer corresponding to required duration of the synthesized output

Putting it together

Below is a file listing of the submitted assignment:

- `A2_func`: the main time stretching function
- `runme.m`: the tester application that does the following:
  1. Loads sample waveform from file
  2. Calls the function with a few different values of time stretch
  3. Plays back original, and time stretched versions