

MUMT605 Assignment 2

Johnty Wang

original due date: 14 Nov 2014
submitted 12 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* for hop size 1.

2 As described in the paper, we need a hop size $H \leq \frac{M}{2K}$ 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 the maximum value of

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

(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:

$$\lambda(u_r, k) = ArgY(u_{r-1}, k) + H\omega(u_r, k)$$

- 4 We need to ensure that any frequency contribution within a particular window's main bandwidth does not get leaked into the next frame when overlapped and added, so the amount that successful windowed versions can be copied back must not be closer than $\frac{K}{M}$ where K is the bandwidth of the window and M the window size.

5 $\alpha > \frac{1}{2}H$

- 6 I wasn't 100% sure how to do this, but from my understanding there is an integer relationship between H , and $t_r - s_r$

- 7 As shown in the paper, the angle of each bin k in the output Y is given by:

$$ArgY[u_i, k] = ArgY[u_{i-1}, k] + ArgX[t_i, k] - ArgX[s_i, k]$$

For a complex number $C = a + ib$, $ArgC = \arctan \frac{b}{a}$, and from the trig identity

$$\arctan(\alpha) - \arctan(\beta) = \arctan\left(\frac{\alpha - \beta}{1 + \alpha\beta}\right)$$

We can see that for two complex numbers $C1 = a + ib$ and $C2 = c + id$, the difference of their Args are, after some simple gymnastics involving fractions:

$$\arctan(b/a) - \arctan(d/c) = \arctan\left(\frac{bc - ad}{ac + bd}\right)$$

And when you divide two complex numbers $C1 = a + ib$ and $C2 = c + id$, you have:

$$\frac{a + ib}{c + id} = \frac{(a + ib)(c - id)}{(c + id)(c - id)} = \frac{ac + bd + i(bc - ad)}{c^2 + d^2}$$

The angle of the result, is also $\arctan\left(\frac{bc - ad}{ac + bd}\right)$; which implies that the angle of the result of subtraction between two complex numbers is the same as the angle of the division of those two numbers. Similarly, addition would imply multiplication.

Finally, given that the magnitude is simply taken from the current $X[t_i, k]$, we can see the result at Y is given by X at t_i , but with a angle increment (multiply) by the difference of angles between $Y[u_{i-1}]$ and $X[s_i]$ (division). The division by the absolute value of $Y[u_{i-1}]$ and $X[s_i]$ normalizes the operation and insures no modification of the magnitude:

$$Y[u_i, k] = X[t_i, k] \left(\frac{Y[u_{i-1}, k]}{X[s_{i-1}, k]} \right) \left| \frac{Y[u_{i-1}, k]}{X[s_{i-1}, k]} \right|^{-1}$$

Part 2

Overview

For this part, I implemented the following files:

- `A2_func`: the main time stretching function
- `runme.m`: the tester application that does the following:
 1. Generate sine wave samples, as well as a more complex waveform from file
 2. Calls the function with a few different input samples
 3. Plays back original, and time stretched versions

Some Notes

In the above method proposed by Puckette using a hop size of 1, it is possible to create time stretches where the fraction is composed of integer numerator and denominator values for the technique. However, to save computation it would also be possible to use hop sizes that are larger (but within the limit specified by the windowing), as in 6.) from part 1. This does bring in the constraint on the numerator value to that particular hop size. Also, the amount of input STFTs performed can be reduced by only computing the ones required at indices required for re-synthesis.

I also attempted implementing the manual method of computing the output spectrum by unwrapping and accumulating the phase at the time stretched positions for each consecutive input STFT, and the code is documented and included in `A2_func` as well. This method, while requiring a.) the phase unwrapping and b.) the recalculation of the complex values from phase and magnitude, does allow for higher resolutions of stretch at a relatively larger hop size. Of course, constraint by windowing still exists, and the actual stretch will be truncated by the nearest sample location where the time stretched version can be placed. Unfortunately, due to an implementation error that I spent a lot of time debugging, I have yet to find a functional solution, and perhaps spent way more time than I should have on it ;). Thanks for your patience!