Sound analysis, synthesis and processing
Homework #*3*

Group components:
Brusca Alfredo 10936149
Pomarico Riccardo 10661306

Mean square errors

Chorale

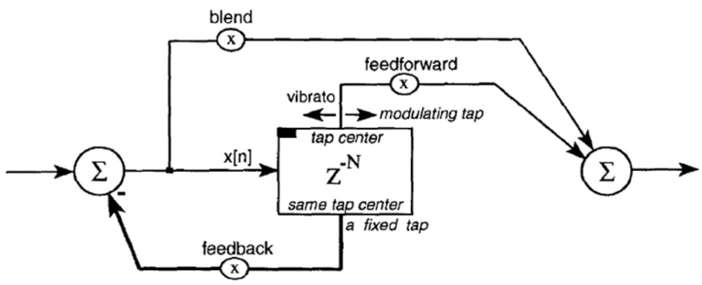
```
Command Window
  MSE: 3.113e-10
fx >>
```

Tremolo

```
Command Window
  MSE: 3.10989e-10
fx >>
```

In this homework, you implemented two SDFs to perform frequency modulation. The subtle frequency modulation achieved by this technique is similar to a vibrato effect. Please name another approach to implementing this effect and describe its computational scheme.

Vibrato can be implemented using the following signal processing path.



It is obtained by setting blend to 0, feedforward to 1, feedback to 0, delay to 0 ms, depth to 0-3 ms and mod to 0.1-5 Hz Sine.
In this way we obtain as output a delayed replica of the input signal, whose delay time is modulated in a sinusoidal fashion. Since blend is equal to 0, we have no direct signal.
To obtain a continuous pitch we need a fractional delay.

When you implemented the two SDFs, the modulator signals were predetermined. Is it possible to use any modulator signal for the chain of all-pass filters without affecting the stability of the filter? If the stability of the filter is an issue, please specify a valid condition for the modulator signal to ensure stability.

The impulse response for the SDF is:

$$H_N(z) = \left(\frac{a_1 + z^{-1}}{1 + a_1 z^{-1}} \right)^N$$

For the filter to be stable, all the poles must be inside the unitary circle. In this particular case, the modulus of the coefficient of the all-pass filter a1, which in the impulse response is both b1 and a2, must be less than 1 to ensure stability.

Let us consider the first all-pass filter in the chain of an SDF. What can we say about its group delay? For a periodic signal with two spectral components u(n) = A_1 sin (ω_1 n / Fs) + A_2 sin (ω_2 n / Fs), with $\omega_1 \neq \omega_2$, as input signal to the all-pass filter stage, is the delay of both frequency components equal? Under what condition?

The group delay for a given frequency is:

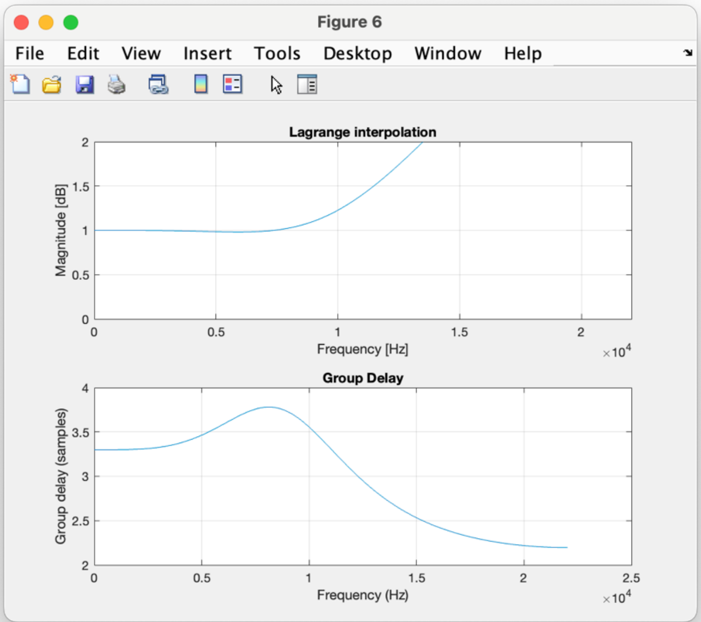
$$\tau_g(\omega) = -\frac{\partial \angle H(\omega)}{\partial \omega} = \frac{1 - a_1^2}{1 + 2a_1 \cos(\omega) + a_1^2}$$

Since this phase response is not linear, the delay of two different frequencies is not always equal. The delay is the same only when cos ω_1 = cos ω_2.

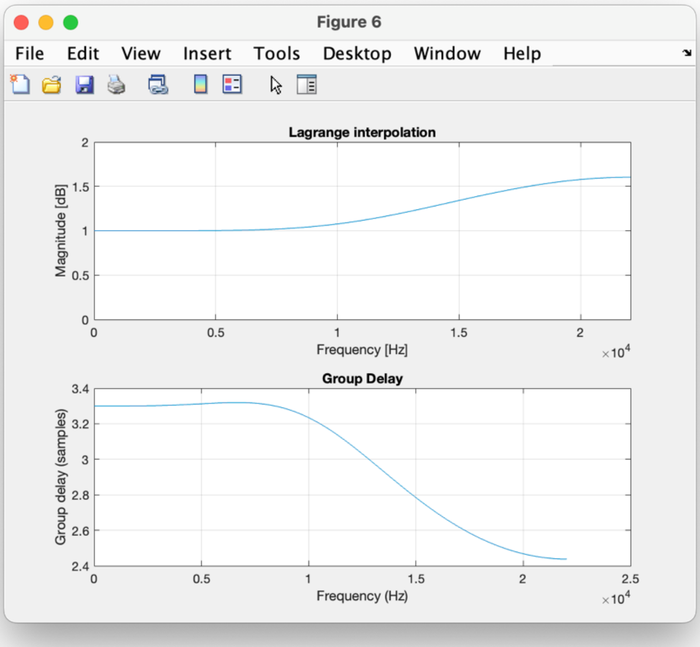
Suppose we implement a fractional delay using the Lagrange method. For a delay D = 3.3 samples, how does the filter order N affect the magnitude response and phase delay? Can we keep increasing the filter order to get better magnitude and phase delay responses? And why? Justify your answer providing useful MATLAB plots of the magnitude and phase delay responses.

In general N needs to satisfy the condition N > floor(D), which in our case means that N is greater or equal to 4. Then, for the filter taps to be optimal we need to satisfy `round(D) - N/2 = 0`, for N even; since round(D) in our case is an odd number we need to look for the other condition `floor(D) - (N-1)/2 = 0`. For D = 3.3 samples this means `3-(N-1)/2 = 0 → N = 7`. If we increase N, the sinc function that interpolates the samples will not be centered around D and thus the filter will not be optimal.

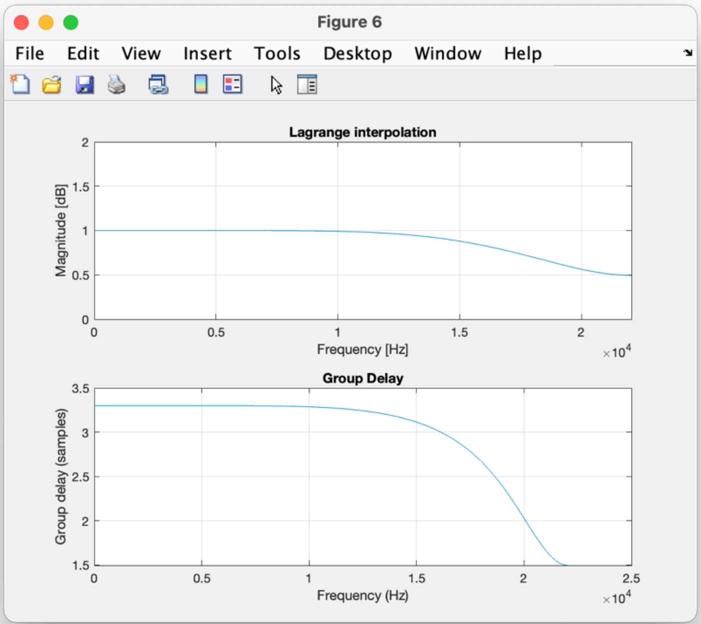
For N = 3 the phase response is correct until 5 kHz and then it has an unpredictable behavior.



For N = 4 the phase response is correct until 10 kHz and then starts to get lower whereas the magnitude starts to increase.



For N = 7 we have the maximally flat response.



We can clearly notice that the result is worse in the case of N > 7 (for example in the picture we set N = 20) and so by increasing the filter order we don't get a better magnitude and phase delay responses.