**Vibrato Synwave**

Vibrato – Separate the input's fundamental sound from the filters, applies vibrato to the fundamental sound then the output is added its original signal filters.

**Syntax**

```
synWave=delay_mod(Input,FS,Width, lpc_filter_order);
```

**Description**

This is a modified all pass interpolation Vibrato function that has added a fractal noise to modulate the signal. Allowing the user to input more parameters into the vibrato.

![Fig1. All pass interpolation delay line](image)

For start, the input is separated into its fundamental noise and a series of filters whose number is determined by the user (the best results are in the range of 3 to 6). Then fractal noise is created for the modulation and the low-pass filter applied to limit the frequency content of the modulator all of this just to the fundamental sound. Finally the Vibrato applied signal is added to its original filters.

```
synWave=delay_mod(Input,FS,Width, lpc_filter_order);
```

produces a signal with a vibrato added only to its fundamental frequency as its output. The whole arguments in the function must be determined by the user.

*Width* determines width of the delay applied to the vibrato.

*lpc_filter_order* determines the number of filter coefficients into which the input signal will be split when divided into fundamental sound and filter(s). Values between 3-6 perform better since more filters tend to be more noticeable at the output signal.
Example

Create a human like vibrato effect to a violin input.
\[ \text{synWave} = \text{delay_mod} \left( \text{Input}, \text{FS}, .0002, 3 \right); \]

Fig. 2: Vibrato added modulation signal graph

Diagnostic

The result is indeed a more human like vibrato, however you can notice the presence of the filter at the end as the sound is fading. If the parameters are kept under the values mentioned the output is quite realistic.

References