Project 4-Chorus

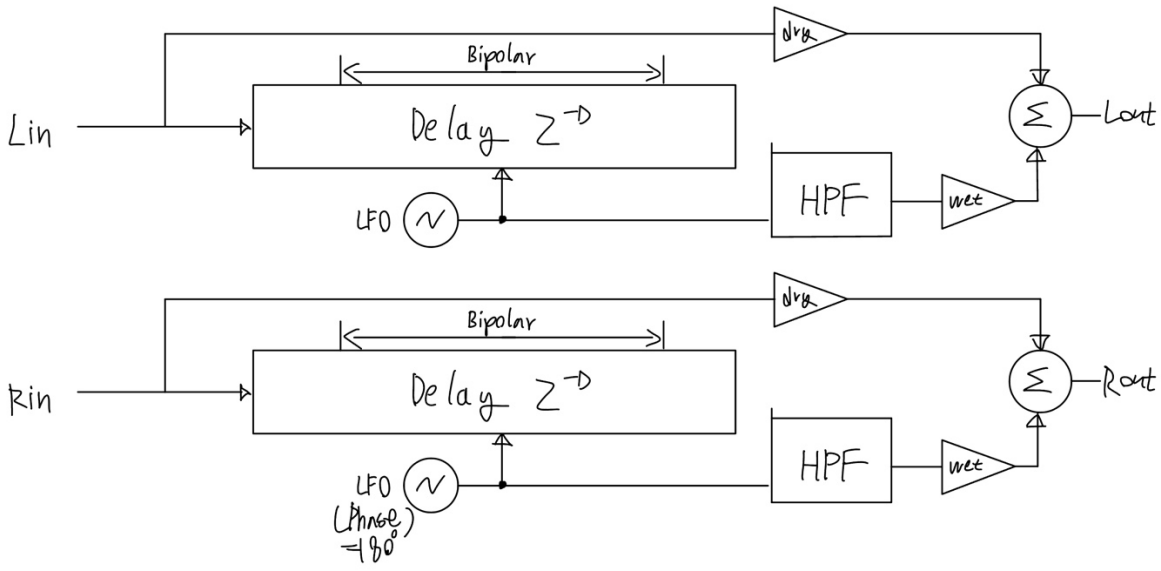
Haizhou Yu

It is worth noting that I don't know much about the chorus, so I designed the project to be close to Logic's stock chorus plugin, both in terms of layout and sound.
I have six parameters in my project code, and below are the names and their range.

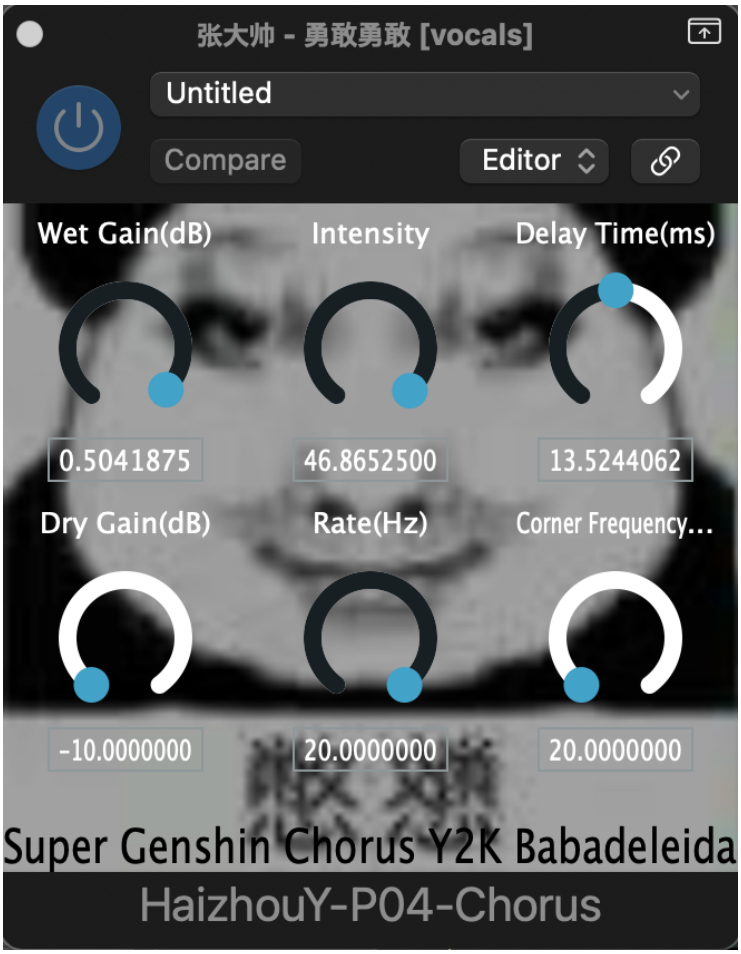
- Wet Gain: -50 to 2 (dB). I wanted to give the wet signal more flexibility, so I made the range very high. -50dB is more like off, and 2dB give a nice little gain over the normal 0dB. Fun fact, I wanted to use a mix knob to control how much the wet is in the signal, but with that it didn't give me a lot of controls, especially with fully wet, the HPF I implemented will make the sound lack low end frequencies, so I stuck with the separate dry/wet controls. The default of -6 is just some practice in music making.
- Dry Gain: -10 to 2 (dB). This is very standard, I don't want the range to be big if not necessary, because it makes controlling sliders very hard. Default is 0dB, very standard for dry signal.
- LFO Depth/Intensity: 0 to 48. I actually played around with it and tested with different values, and I found out 48 sounds pretty close to the logic stock plug-in. The number range is set like this because of the way I calculate depth.
- LFO Speed/rate: 0 to 20 (Hz) this is the same as the Logic stock plug-in. default is 2, which is also the same as the logic's stock plug-in, and they sound very similar.
- Delay Time: 2 to 25 (ms). Though in class we said it is better to keep the delay time below 30 ms, but I still want flexibility, so I went with 25, which is the maximum center time of the delay. 2 for minimum is just to make room for the LFO modulation. The effective delay time is 1 to 49 (ms).
- Corner Frequency: 20 to 10000 (Hz). The default is at 20Hz, which is simply off, as it is a HPF. The reason why I choose the HPF, is because low end gets quite muddy when implementing higher LFO rates. This is one extra feature that I made for the project based on the experiences using the chorus (I had never used chorus or flanger before, but only for this course. They are fun!). The difference of the output sound after 10kHz is minimal, so I left the upper limit at 10kHz, again, for more precise controls on the sliders.

In terms of mapping, I made the two dB sliders as logarithms, so they are better to use.

Presented below, is the signal flow diagram, and the project GUI.



Project's signal flow diagram



Project GUI

Extra things done to the project:

Besides the HPF I talked about earlier, I've discovered in logic, the left and right channel outputs are pretty different, but the same time, users don't have a way to change the phase (I find it useful since messing with the phase sometimes messes up the sound). So, in my project, I modified the LFO.cpp, and included an LFO.flipPhaseTick function, that gets the output for a flipped phase LFO. This was used to control my right LFO, so it is in the opposite phase. I researched on lots of GUI stuffs this time. I first added a funky picture to use it as my background. I also made a lot of modifications to the picture, so it is translucent, which helps make the words on the label clearer. I also made outlines to my slider and figured out a way to make a panda-ish color scheme for the sliders, so it matches the background better. I also changed the size and offsets of the default text to make it the name of the plug-in and put it on a suitable spot. A peek at the plugineditor file will know I've put a lot of work into the GUI.

Verfication:

At the first two bars, the plugin is set to a mild setting, so not much effect is demonstrated. In the next two bars, I cranked the intensity all the way up, so we can clearly hear it is making some funky noises since now with a bigger intensity, the wet signal is more obvious. For the next two bars, I cranked the frequency all the way up, and we can hear that the funky sound is modulated quicker. I then turned the intensity down, leaving the frequency all the way up, we can still hear it changing quickly but not that obvious. For the next two bars, I showed the filter working, by making the corner frequency gradually higher. The wet signal becomes pitchy and nearly disappeared in the process. Lastly in one bar, I revert the settings for a comparison with the next bar, which then I cranked up the delay time. This made the sound more spital.

Ecological use:

I used the chorus in my vocal. In the first part of the chorus, I cranked up the effects, so it feels like there is a vocal backing at a smaller volume sing from behind. Then, in the first part of the instrumental, I turn down the intensity of the vocal track, since now it is a vox instrument, and playing very quietly from behind. I don't want to increase the burden of mixing, so I temporarily turned down the intensity. But, on the second part of the instrumental, I pushed the vox to a higher volume so it's now a supplement of the leading instrument. Therefore, I turned the intensity back up, so it now sounds more spatial. Even though the parameters of the two parts that implemented the chorus are pretty much the same, they are for different purposes.