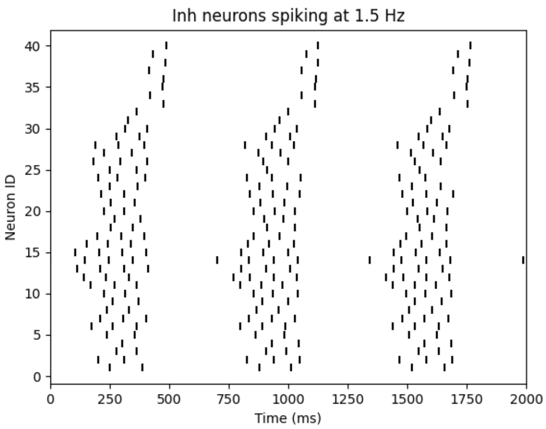
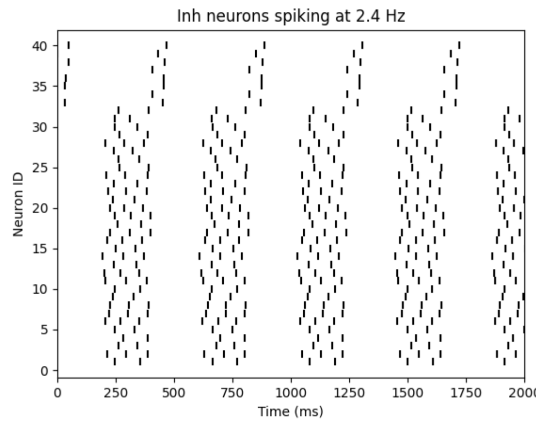
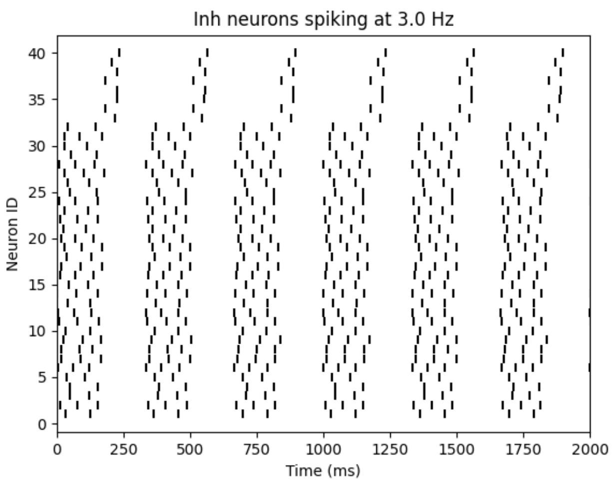
As a first step we created coupled oscillators that oscillate at frequencies within the range of the human heart beat. We did this using coupled oscillators with a regular input from FPGA spike generators. Higher oscillation frequencies were achieved with higher spike generator frequencies:

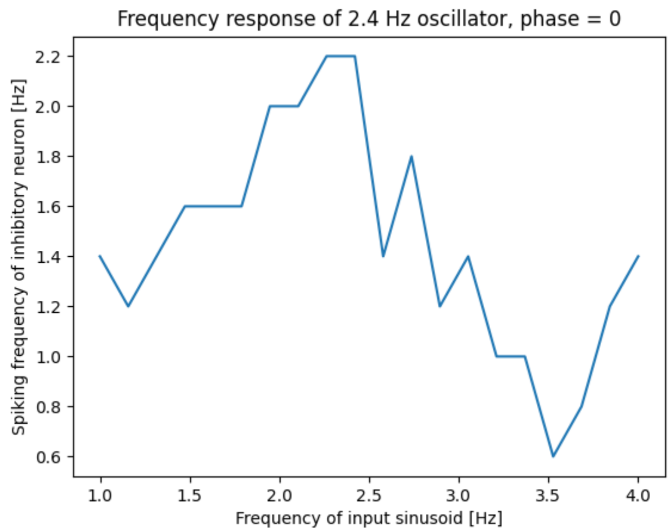




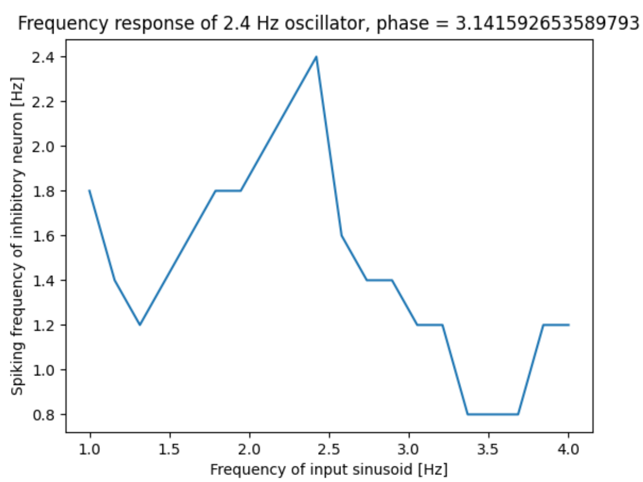


The output frequency of the oscillator is measured as the spiking rate of the inhibitory neurons.

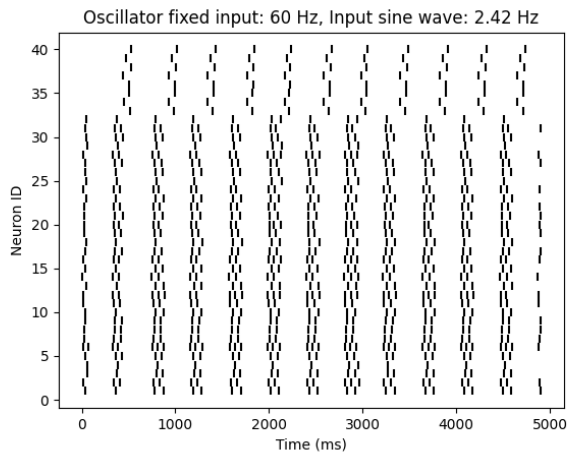
To see what effect sinusoidal inputs with different frequencies have, we created two more spike generators with synapses onto the excitatory population of the oscillator. The spike times of these input spikes were computed from a sine wave using ADM. The up spike times were given to one spike generator, which connects to the oscillator with AMPA synapses, and the down spike times were given to the other spike generator, which connects to the oscillator with GABA_B synapses. We took the oscillator with a frequency of 2.4 Hz, fed it different sinusoidal input frequencies, and measured the firing rate of the inhibitory neurons:



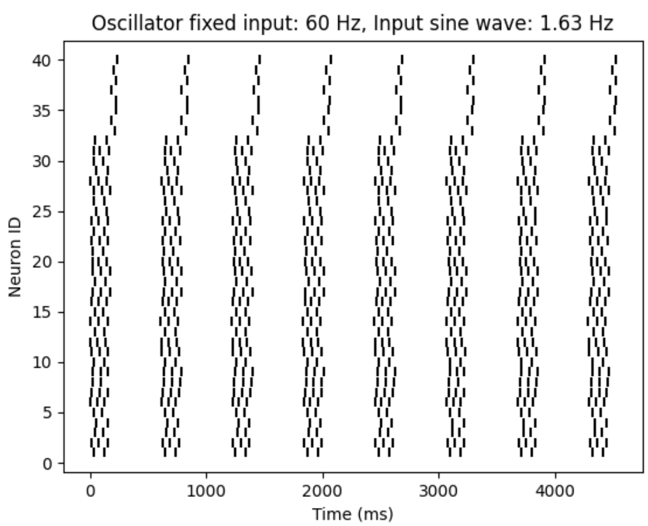
We can see that the highest output firing rate is reached when the input frequency roughly matches the oscillators frequency with no input, which in this case is at 2.4Hz. The same behavior can also be produced when repeating the experiment with the same input frequencies, but with phases shifted by pi:



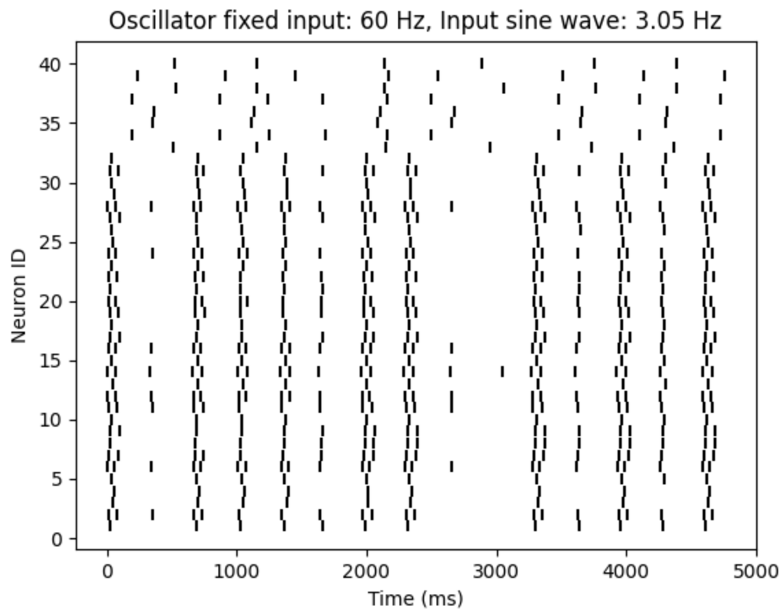
Let’s look at what happens to the spiking pattern as we vary the input frequency. If the input frequency is right at the oscillator’s natural frequency, we pretty much get the same spiking pattern that we would if there was no input:



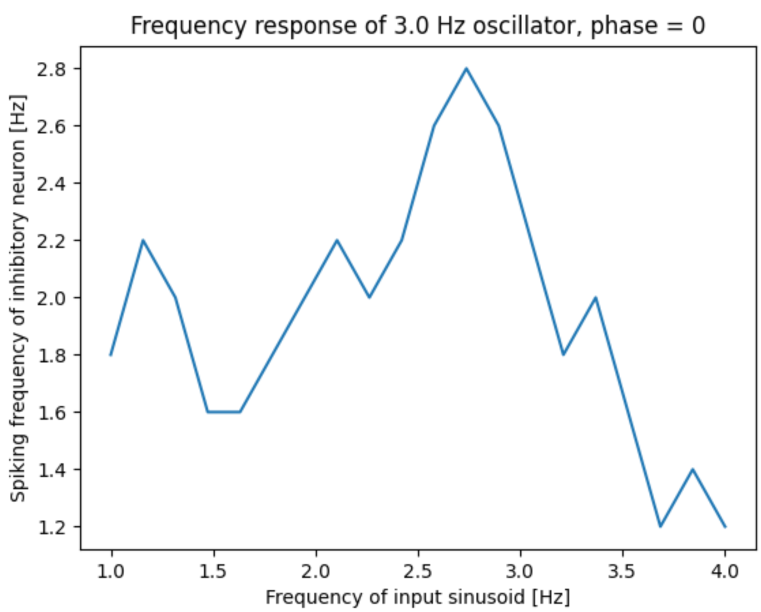
Depending on whether we move the input frequency above or below the oscillator's natural frequency, 2 different things happen. If we go below, the oscillator seems to still oscillate regularly, but at a lower frequency, such as in this example with an input frequency of 1.6 Hz:



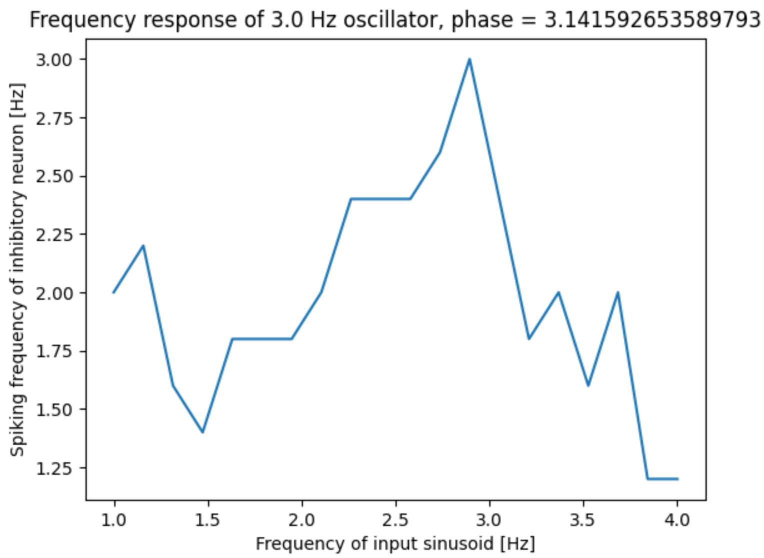
If we go above 2.4 Hz input frequency, the oscillators activity starts to become chaotic, like in this instance with an input frequency of 3 Hz:



If we plug in a an oscillator with a higher frequency, 3.0 Hz in this case, we also get our response peak at a higher frequency:



However, the peak lies a bit below the oscillator's natural frequency. Also, the peak of the anti-phase version is slightly shifted:



However, as long as we can generate oscillators with many different peaks, we should technically be able to implement a neuromorphic oscillator-based fourier transform approximation.