MASTER OF SCIENCE
IN ENGINEERING

Solution of Adrian Willi

Lab 2: Perception of Speech Signals

Issue: 16th November 2022

Submission: 23th November 2022

Task 1: LTI Systems (Linear Time-Invariant Systems (LTI))

Before analyzing a digital signal, some preprocessing steps are normally done in order to emphasis the information of interest in the signal and get rid of the unwanted information (e.g. noise). It is important that these preprocessing steps do not change the information of interest in the original signal. That is why LTI systems (LTI operations) play an important role in the preprocessing steps as they hardly change the signal information of interest when applied correctly.

Run each section of the Live Script `LTI_systems.mlx` (type `edit LTI_systems`) one-by-one (using "Run Section" instead of "Run") and find out what auditory effects the following operations have on the given signal x(n):

1. y(n) = abs(x(n))
   Sounds like a robot and the sound is noisy. No more negative values for the amplitude.
2. y(n) = $\sqrt{x(n)}$
   Hard to understand but possible (maybe also only because I already know the text). The negative values of the amplitude disappear because of the applied operation and also the scale is different.
3. y(n) = x(n) with hard delimiter at 0.05
   The frequencies are clipped at 0.05. Sound was good hearable. I expected that the sound is noisy and the message is not present. Don't know if something went wrong.
4. y(n) = (x(n)+x(n-1)+x(n-2))/3 (mean value)
   Actually I can't hear a real difference and the message is good understandable.
5. y(n) = (x(n)-x(n-1))/2 (difference value)
   It looks like that the base is removed from the recording and only the low frequencies remain. The message is stil pretty good understandable.
6. y(n) = median(x(n), x(n-1), x(n-2))
   I can't hear a big difference for this operation. The only difference is that the background sounds more like some rain outside.

7. y(n) = (x(n) + x(n-1) + … + x(n-p))/(p+1) (Moving Average)
The bigger the window size is, the bigger the impact on the message and it also sounds duller.

Which of these operations hardly change the important information, i.e.the speech signal?
Multiply by a constant, clipping, mean, median, filter with big window

Which of these operations are LTI systems (without prove)?
Filtering, convolution, mean

Task 2: Phase Perception

Start the program **`phaseperception`**. There you see the sum s of two signals s_1 and s_2, where the phase ϕ of the second signal is lagging behind the first signal in the range of $0 \leq \phi \leq \pi$. You can change the frequencies of the two signals and the phase ϕ with the corresponding sliders.
Sum of two cosine signals with phase shift: s(omega*t) = cos(omega1*t) + cos(omega2*t + Phi)

Task:

Try out different frequency pairs. Look how the signal s in the plot changes its characteristic. Now listen to the signal, what do hear when you change the phase ϕ for 0 to π?
Changing the frequency pairs leads to higher sounds for higher frequencies and deeper sounds for lower frequencies. This is what we have seen in lab 1. Furthermore, I can see a movement of the peaks of the amplitudes of the signal. → Multiplications of frequencies result in an accumulation of the peaks.

If f_1 and f_2 are equal then no sound is hearable if ϕ is π. They cancel each other out in this very specific case.

Task 3: Frequency Perception

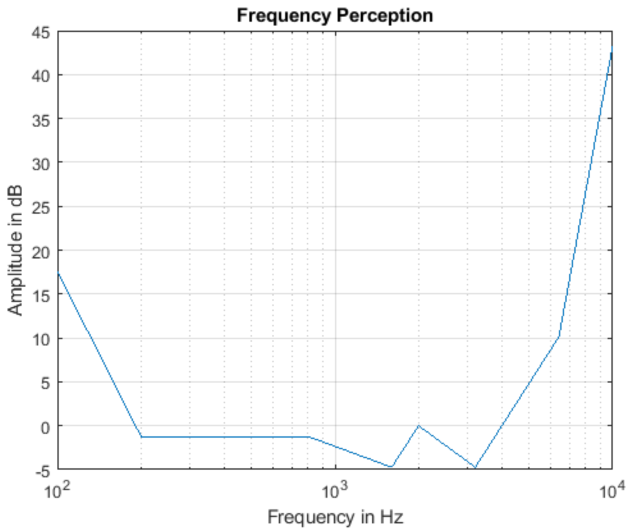
Start the program **`frequencyperception`**. Determine your absolute threshold of hearing ("Ruhehörschwelle") at 2000 Hz by adjusting its slider so that the sound is just audible. Save the sound level as reference level by pressing "Set reference sound level".

Tasks:

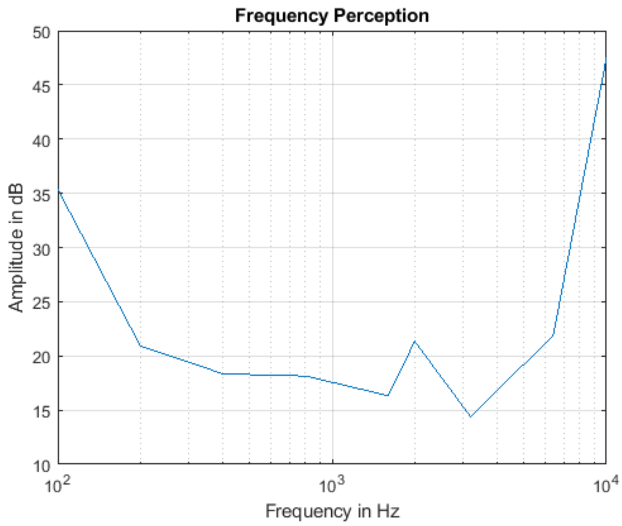
1. Adjust the level of each frequency with the corresponding slider so that the perceived sound is as loud as the reference sound. When you adjust the slider you first hear the reference sound and just afterwards the sound whose level you are adjusting. Its frequency is indicated at the bottom of the corresponding slider.
2. After you have adjusted all sliders accordingly, press the button **`Plot`** in order to see the loudness plot that shows the signal amplitudes as a function of the signal frequency that are needed so that you perceive the same loudness. Press **`Hold`** in order to copy the plot into another figure.
3. Now set the reference level to a higher value, but **do not press** "Set reference level".
4. Adjust the level of each frequency as in Task 2.1.
5. After you have adjusted all sliders accordingly, press the button **`Plot`** in order to see the new loudness plot. Press `Hold` in order copy the new plot into the figure with the other plots.

It produced for both settings plots that look similar. They are baisically just shifted in the Amplitude in dB (y-Axis).

At threshold



Above threshold



6. You can save the collected plots by pressing the save icon at the top of the figure. Choose an appropriate format, e.g. "PNG"

Task 4: Masking Effect

Start the program **`maskingeffect`**. There you see and hear a bandlimited noise signal with center frequency 1000 Hz and bandwidth 80 Hz. In addition, you see and hear a sin-wave at 500 Hz with an amplitude about 20dB less than the noise signal.

Tasks:

1. Raise the frequency of the sin-wave with the slider or the text field until you no more notice the sin-wave. Note the frequency where this happens. Further increase the frequency until you again hear the sin-wave-signal. Note this frequency, too. From which to which frequency the sin-wave is masked by the noise?

   The sin-wave is masked by the noise from the frequency 920 to 1'080 when the noise center is at 1'000 Hz.

2. Repeat Task 3.1 with another center frequency of the noise, e.g. 2000 Hz. Where do the frequency limits of the masking effect lie now?

   The sin-wave is masked by the noise from the frequency 1'840 – 2'160 when the noise center is at 2'000 Hz. The range where I can't hear the signal is a bit wider for 2'000 Hz than for 1'000 Hz. This is probably related to the higher base frequency.