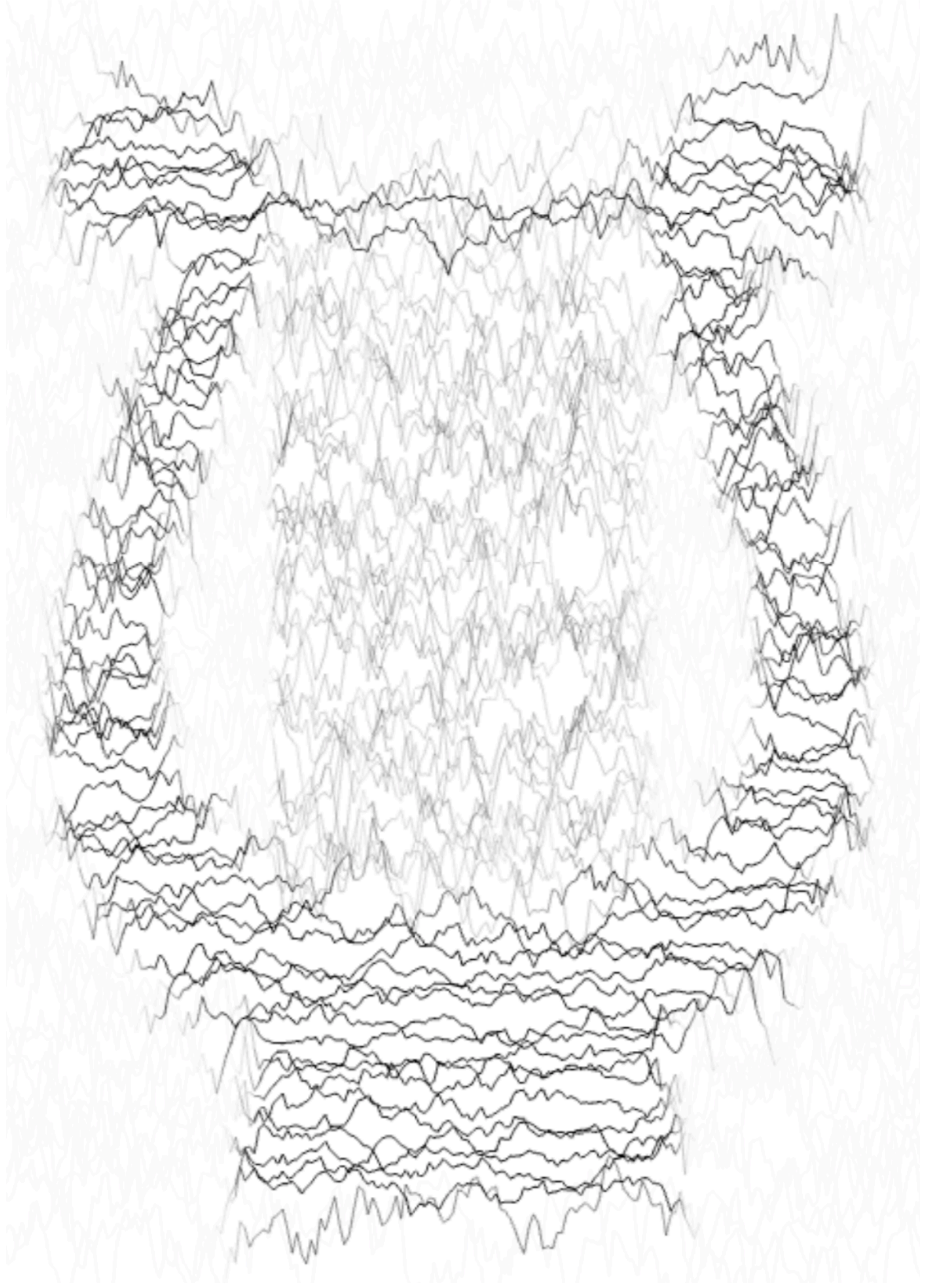
University of Illinois @ Urbana-Champaign



CS 498PS – Audio Computing Lab

Denoising

Paris Smaragdis
paris@illinois.edu
paris.cs.illinois.edu

Overview

- How do we remove noise?

- Single-channel methods
  - Filtering, spectral subtraction
  - How to model noise

- Measuring performance

NOISE!

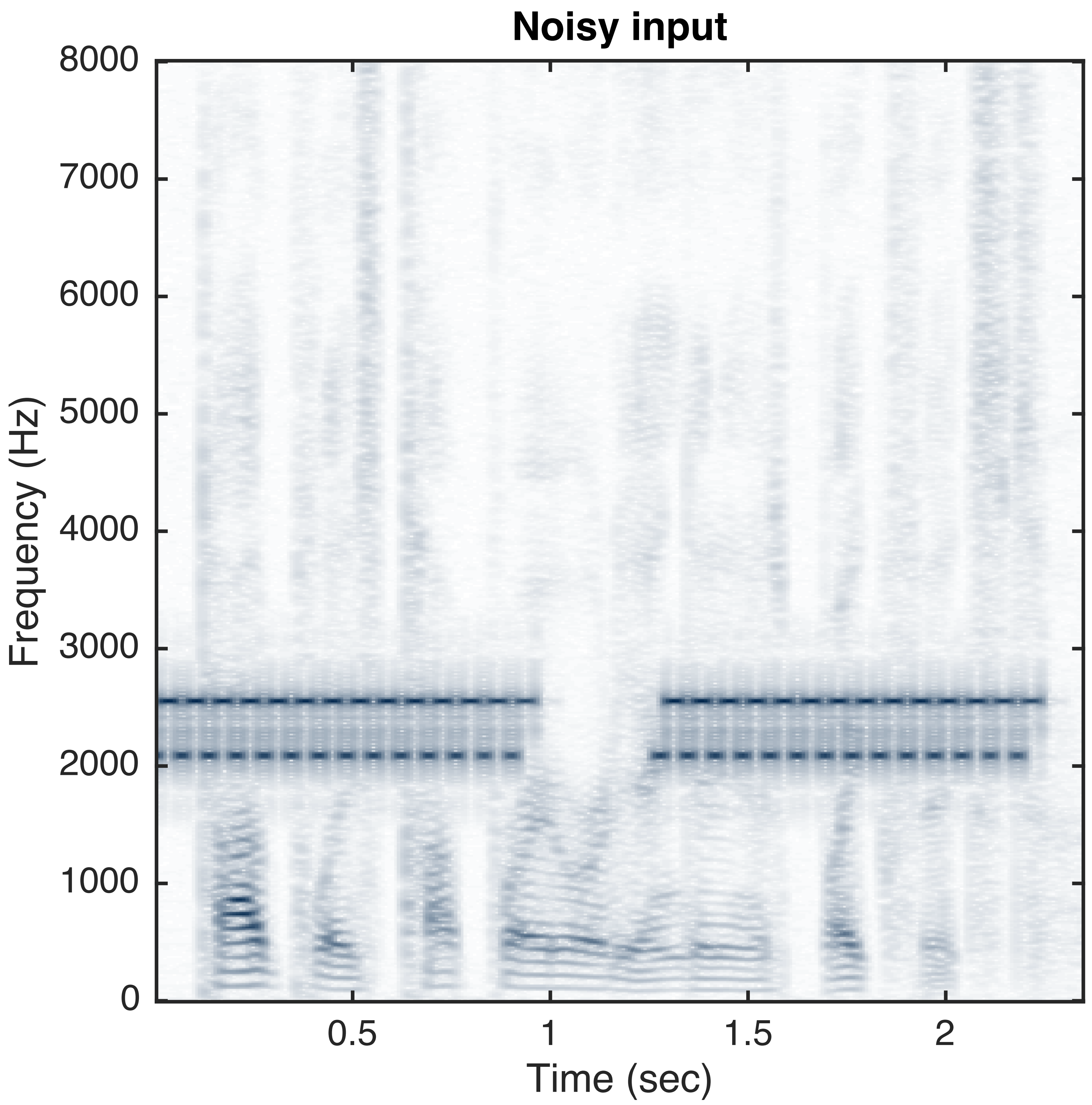
- We don’t like it!
  - Degrades intelligibility
  - Lowers sound quality
- An unfortunate fact of life
  - It is almost impossible to avoid noise
  - We need to be able to deal with it
- Denoising to the rescue!

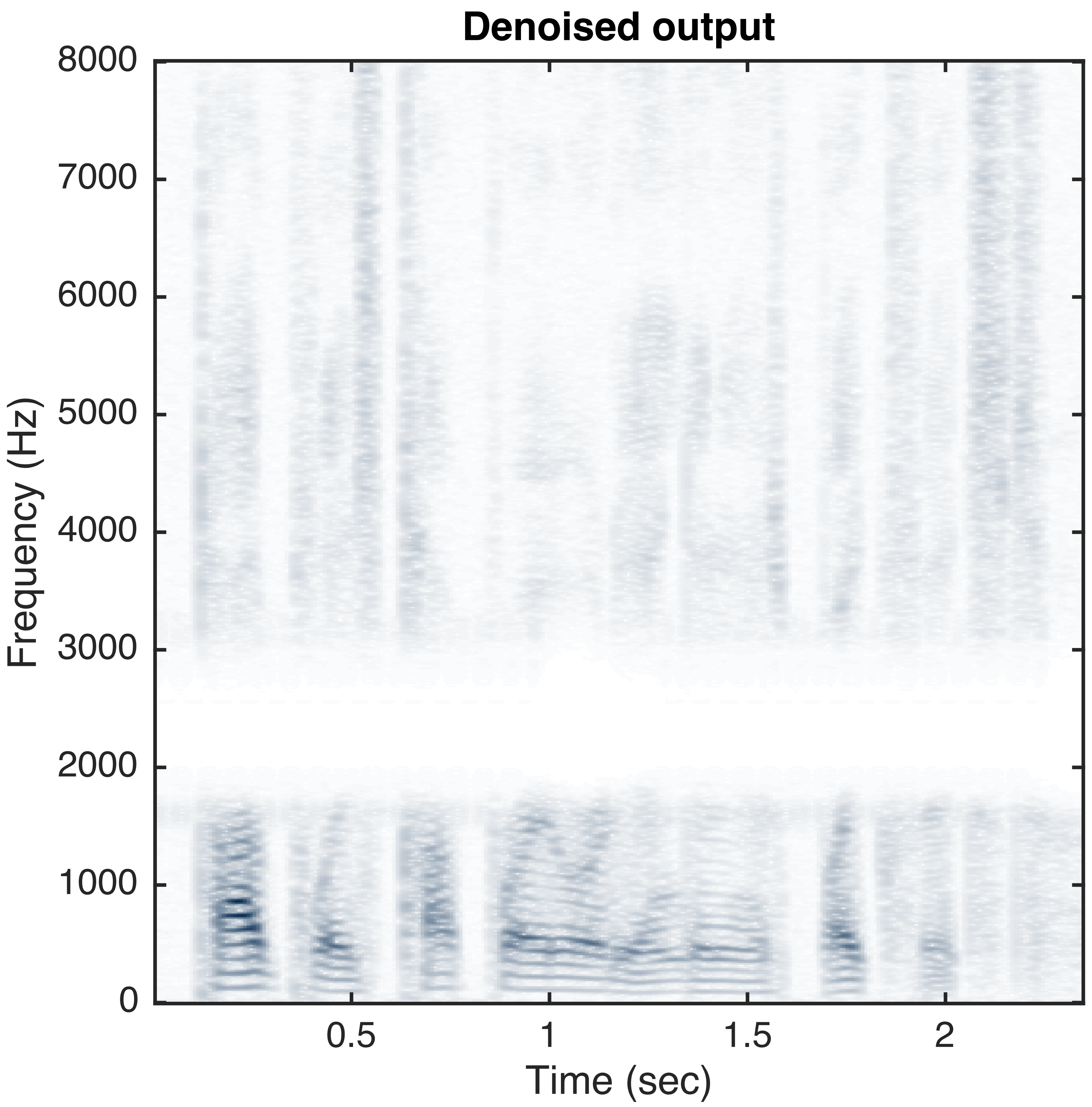
Standard applications

- Cell phones / voice communications
  - Removing non-speech elements (e.g. traffic)
- Data preprocessing
  - Preparing data for analysis (e.g. for speech recognition)
- Forensics
  - Improving intelligibility to help parsing

You've already done denoising

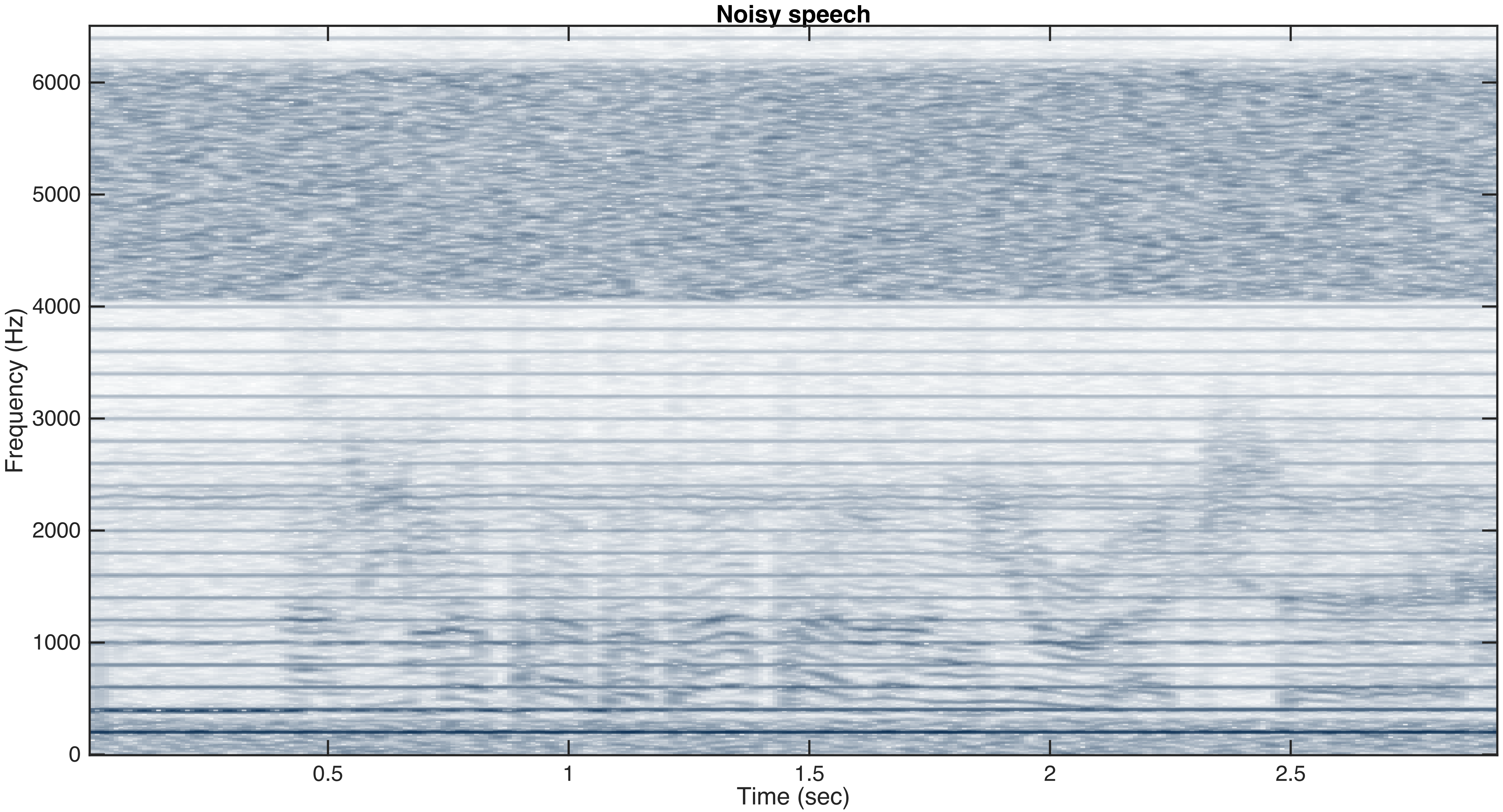
- Lab 1: Filters
  - Remove a cell phone ring using a bandpass filter





A noisy speech sample

- Pilot chatter in a noisy cockpit
  - Where is the noise?

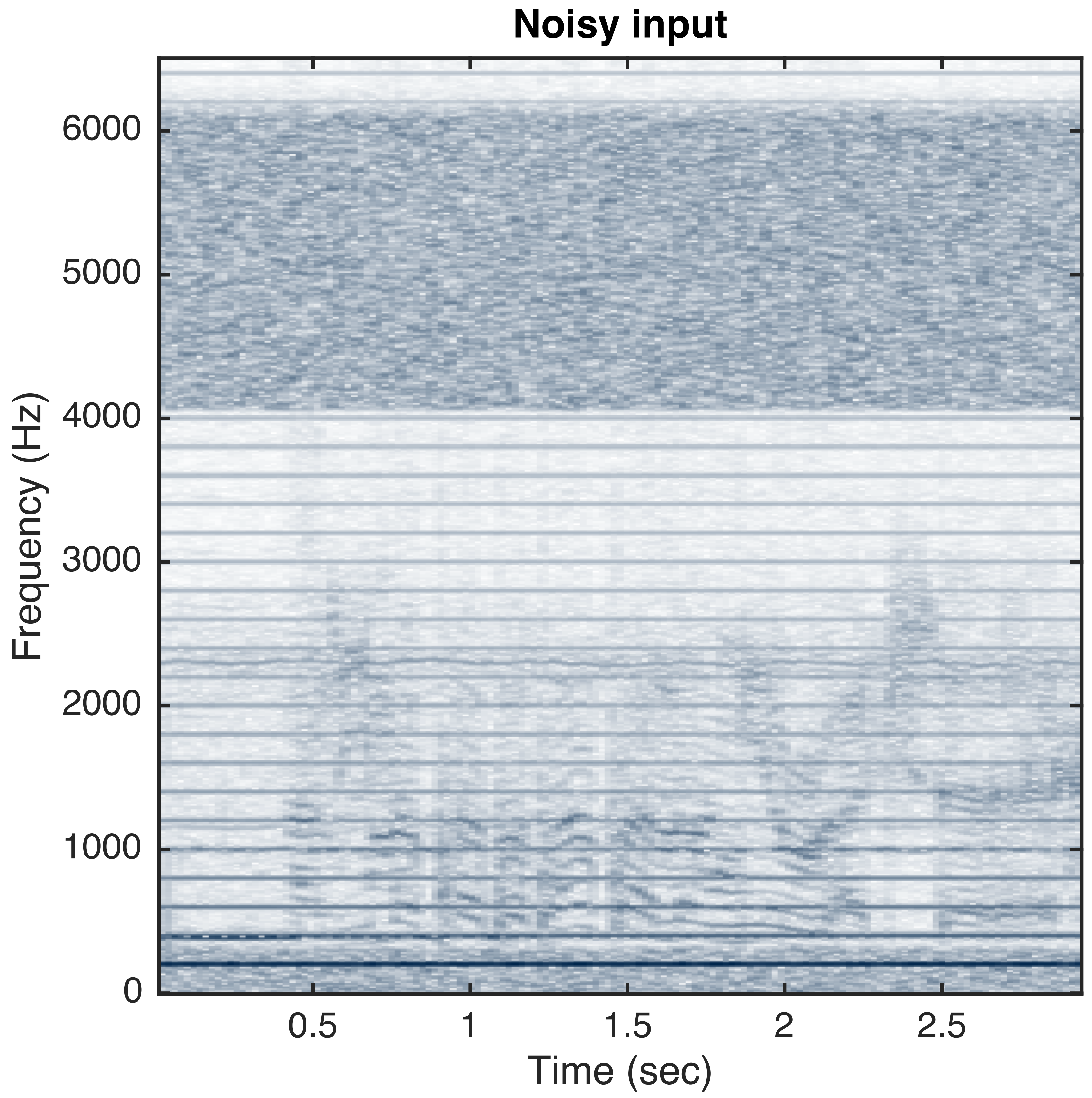


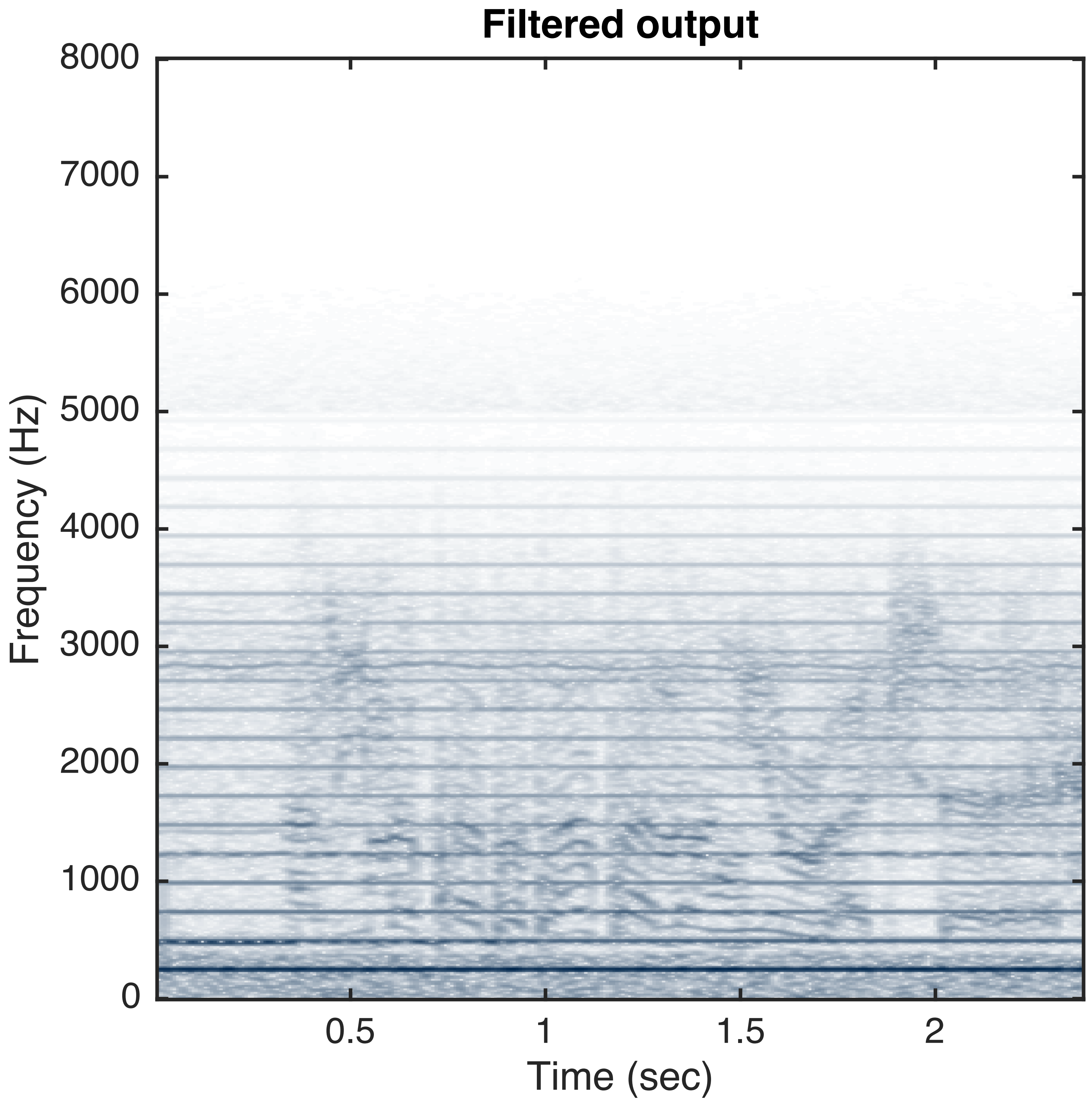
How do we remove the noise?

- Design a filter to remove unwanted parts
- This isn't a single frequency band though
  - We need to use a complicated filter shape
- Or we can use multiple filters in series
  - Each knocking down only one noisy band at a time

A piecewise approach

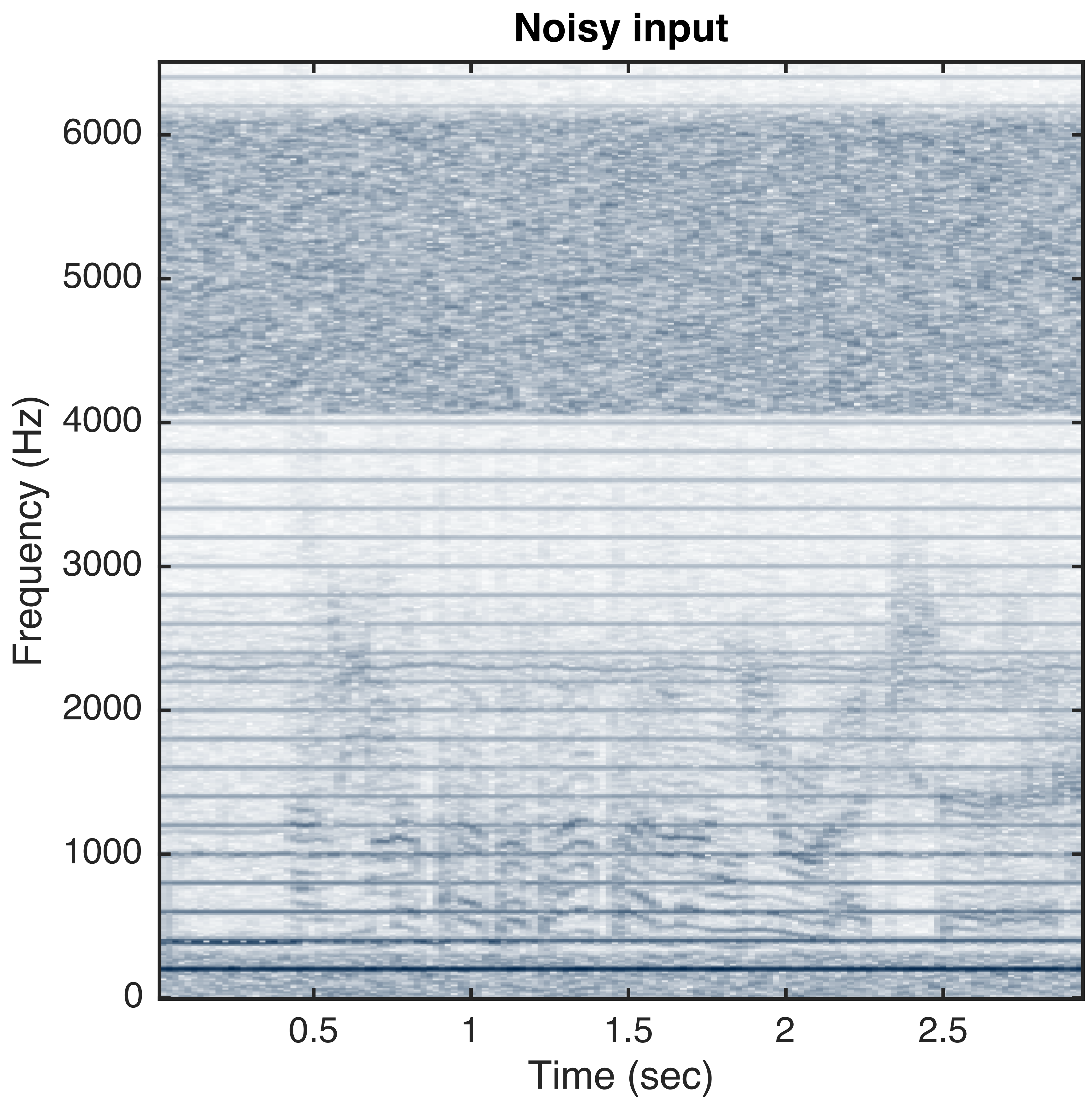
- Remove the high frequency noise
  - Use a lowpass filter

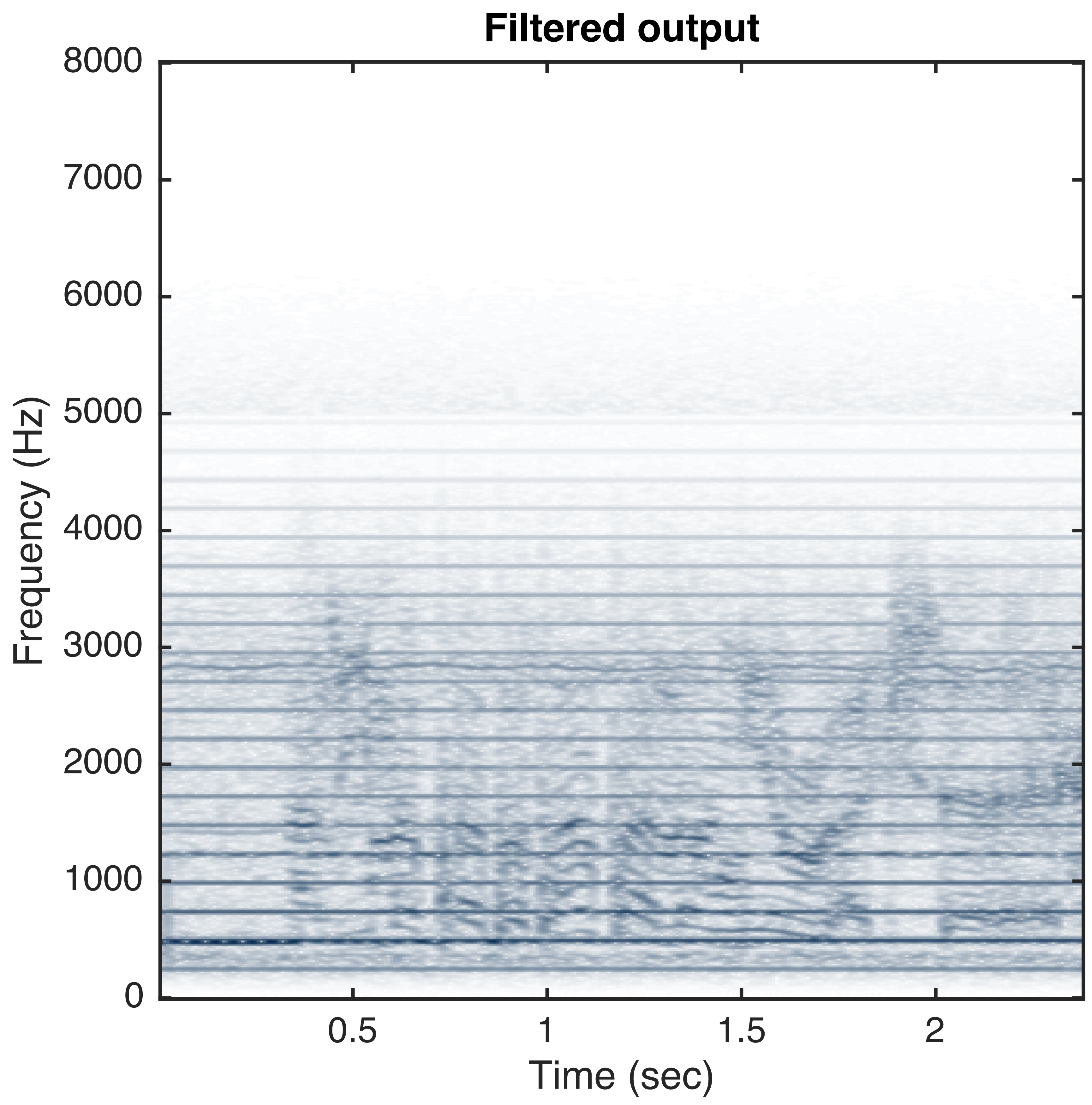




A piecewise approach

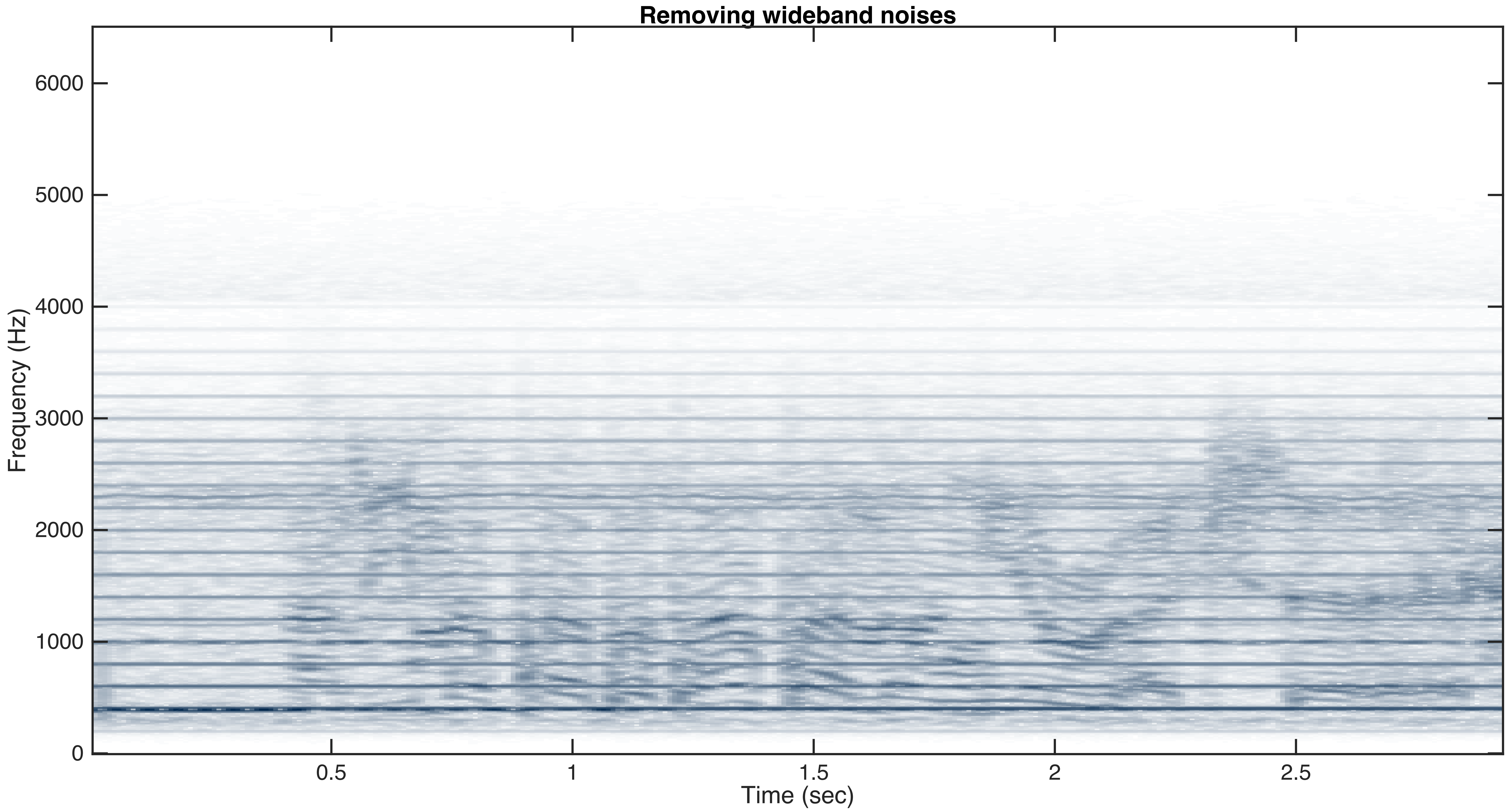
- Remove the low frequency noise
  - With a highpass filter





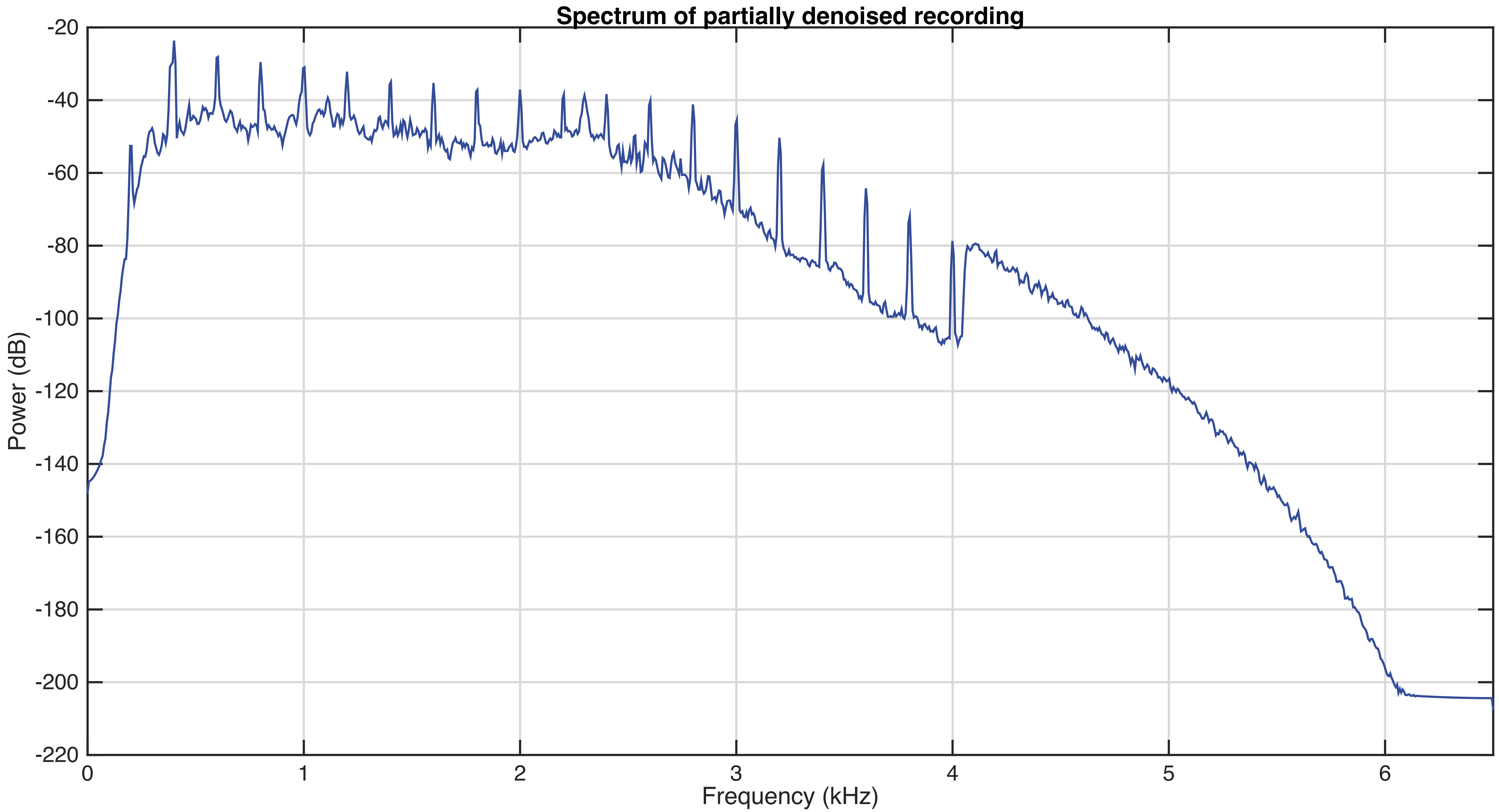
A little better now

- We could have just used a bandpass filter



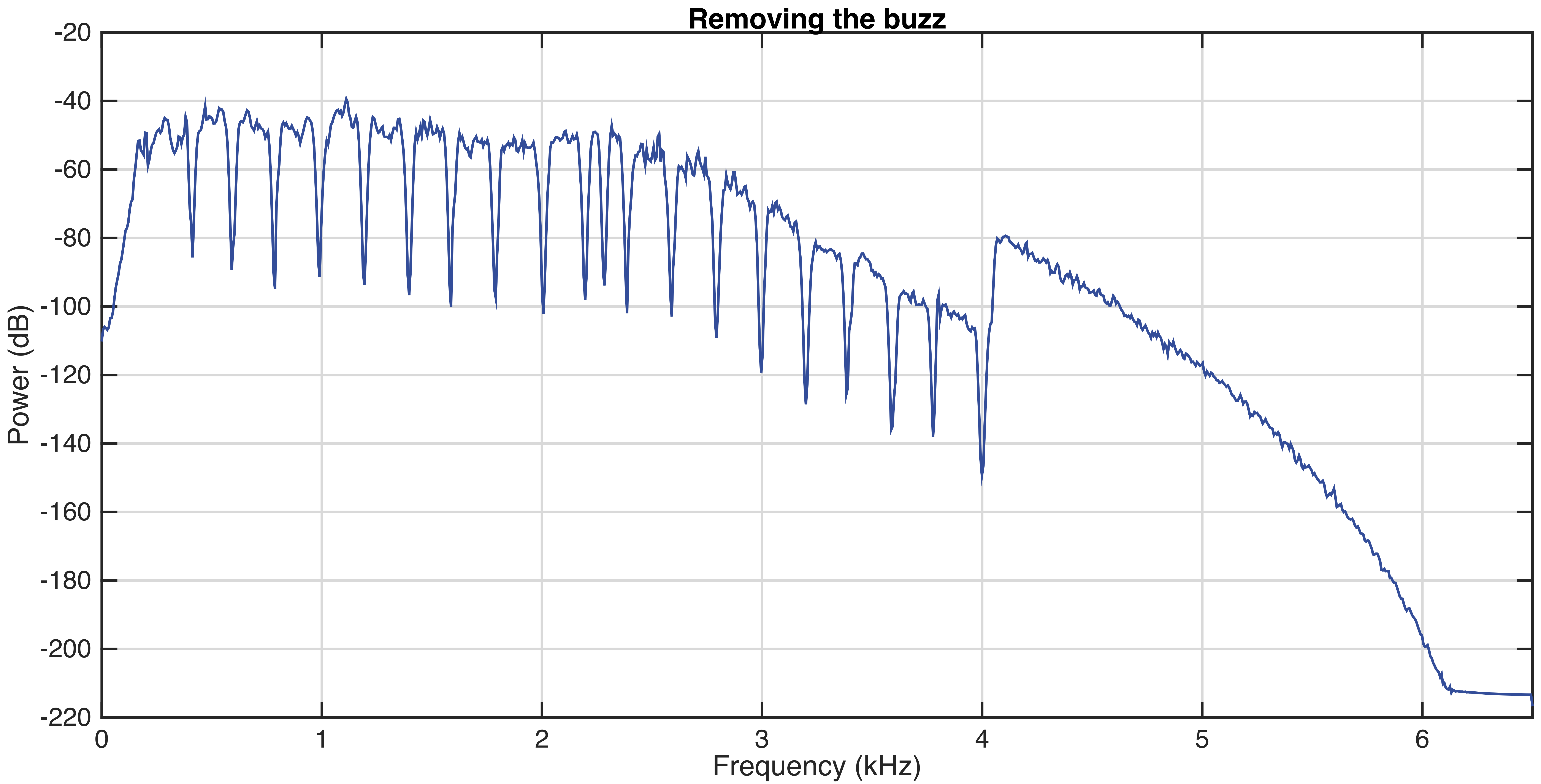
Removing the buzzing

- How do we get rid of that?
  - Multiple noise peaks at regular intervals



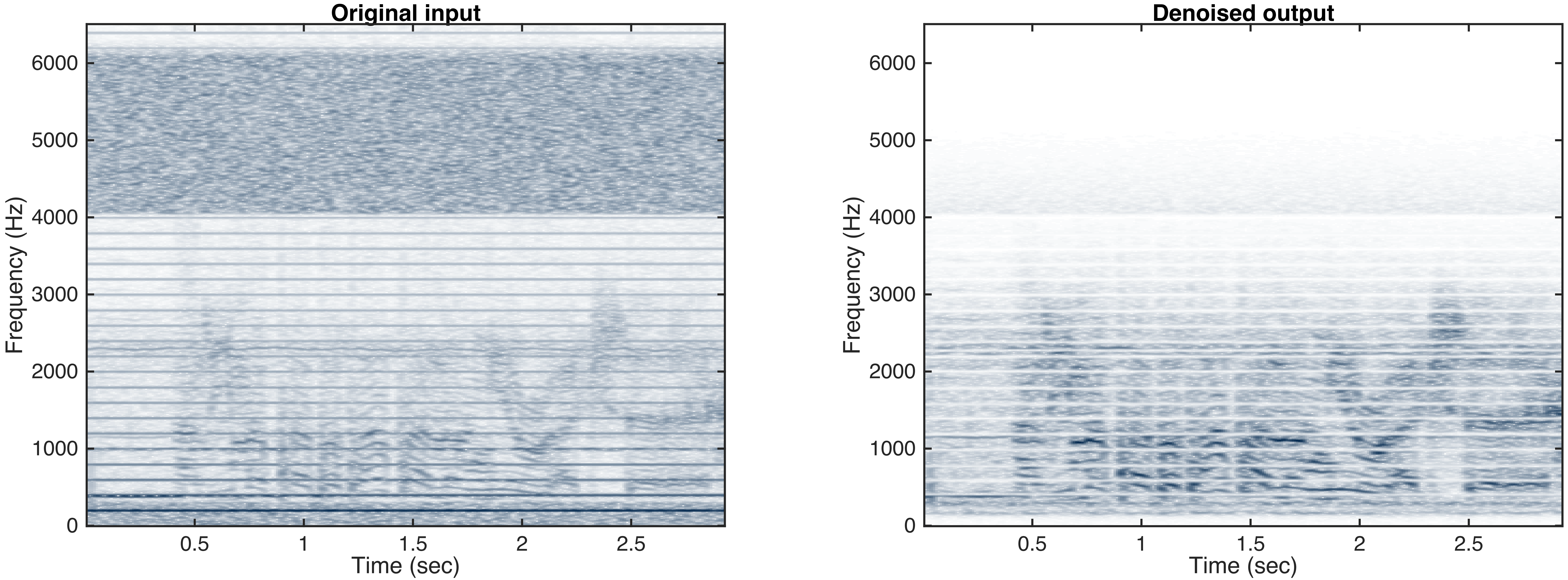
Piling on more filters

- Bank of bandstop filters to remove buzz
  - Center each filter on each noise peak



Final result

- Noise elements are suppressed, speech is cleaner
  - But we get some audible spectral notches (hollow sound)

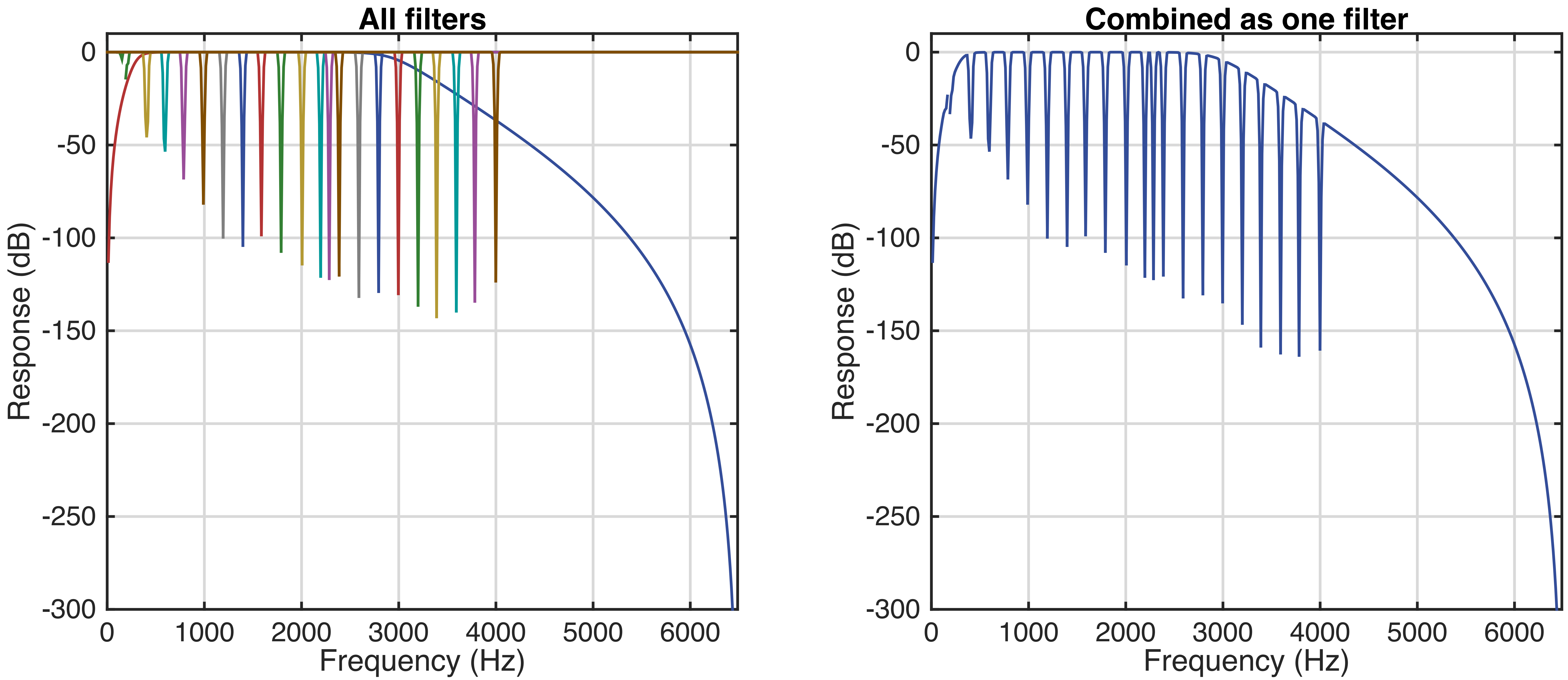


Overall filter

- The overall response is thus:

$$H[\omega] = H_{low}[\omega] \cdot H_{high}[\omega] \cdot H_{f_1}[\omega] \cdot H_{f_2}[\omega] \cdot \ldots$$

- With the final filter being:



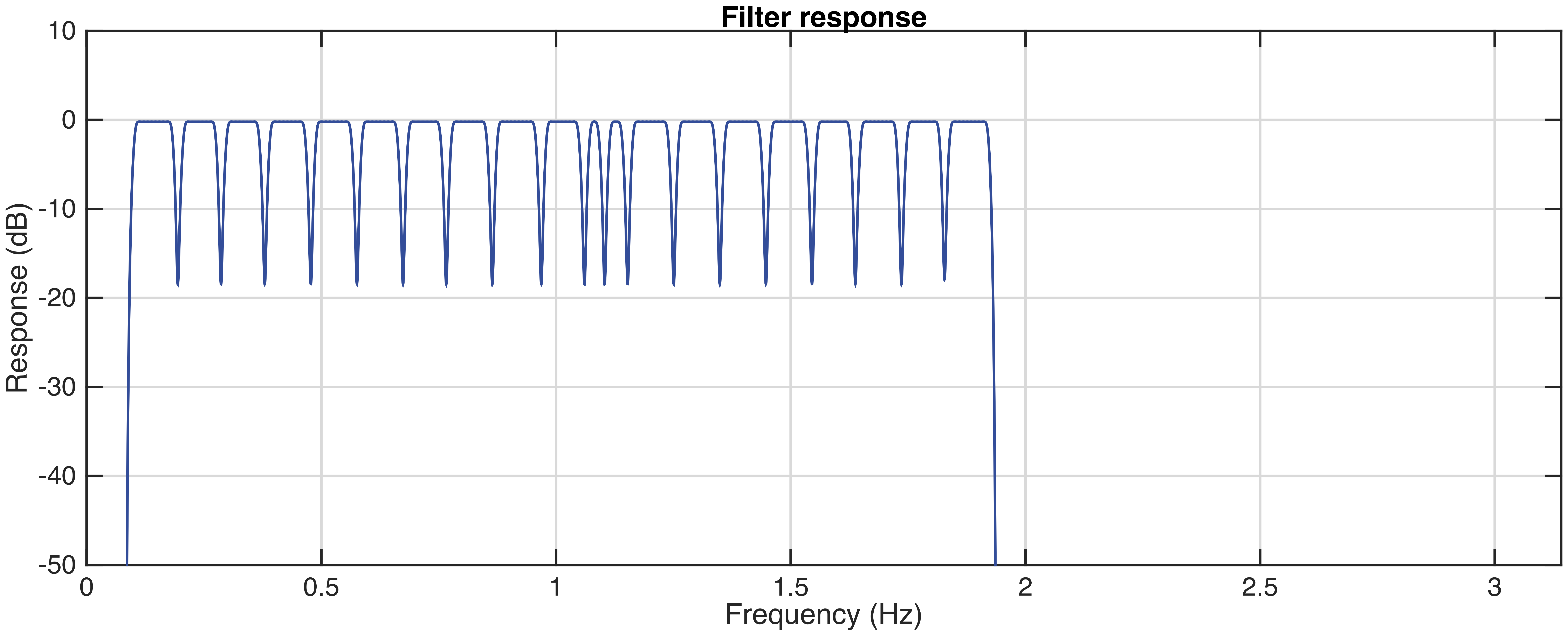
That was a lot of work!

- Can we do this faster?

- Easy answer: we can design the filter in one go
  - E.g. with frequency sampling (`firwin2`)

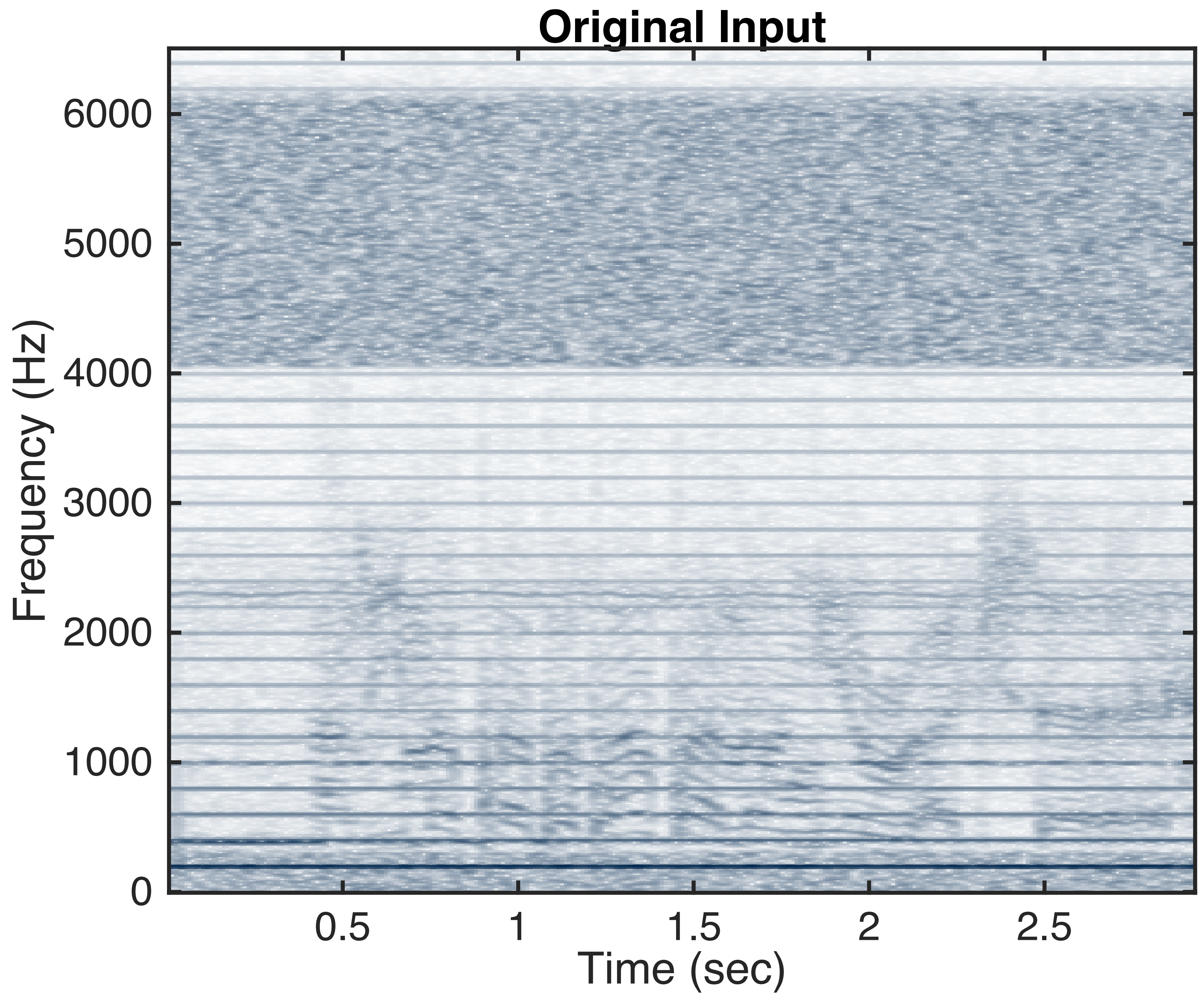
Designing the filter

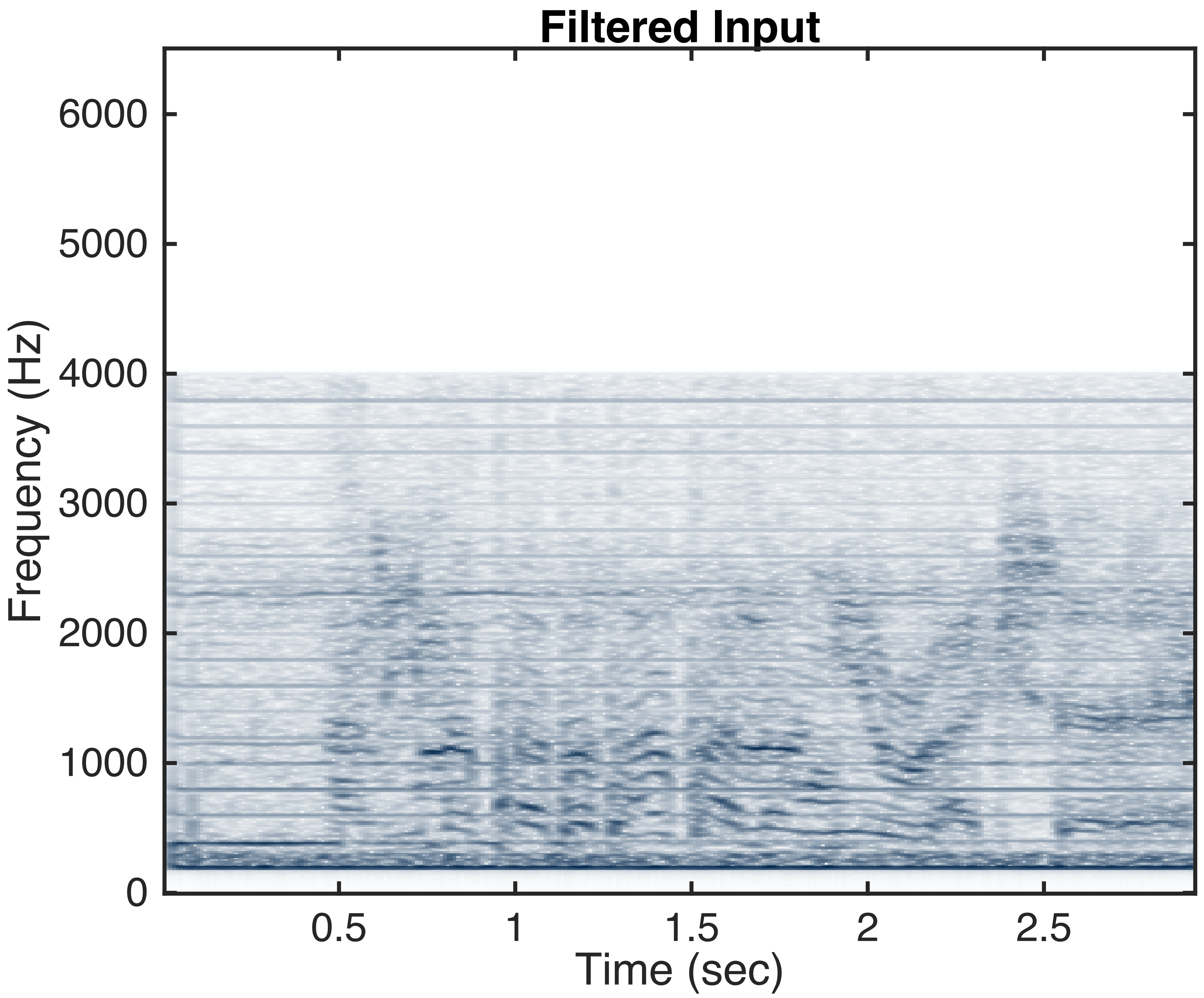
- Setup constraints explicitly
  - Zero response at high/low bands, and at the buzz nodes
  - Unit response otherwise



Result

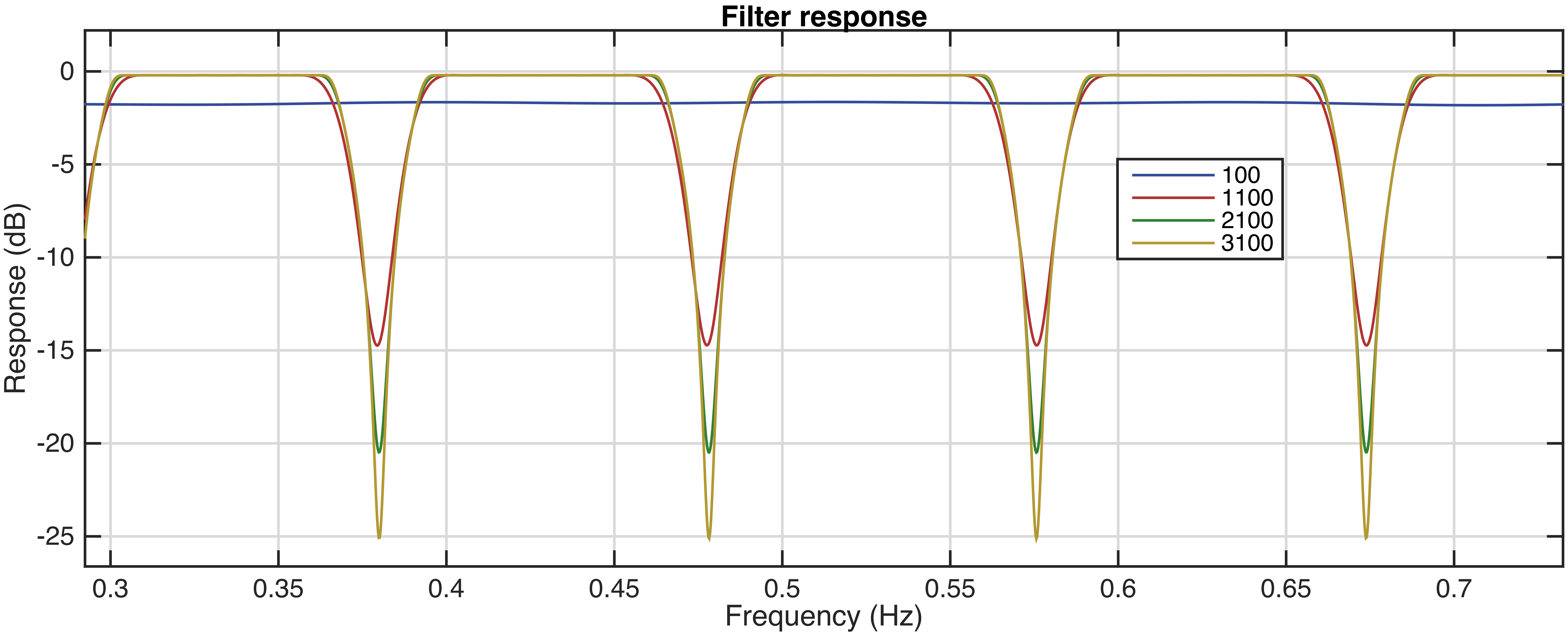
- Works pretty good (as expected)
  - Effectively the same thing as before
  - But with just a single design step





And with more taps

- The longer the filter the better we're off
  - At the expense of more computation



Still too much work …

- Finding all the peaks is tedious
  - What if they are changing?
- Is there a way to automate this?
  - Can we use an alternative approach to denoising?
    - Hint: you did in Lab 1

Editing the spectrogram directly

- What if we just zero the unwanted STFT bins?
- Process:
  - Take the magnitude and phase of the STFT transform

$$X_a[\omega,\tau] = \left\| X[\omega,\tau] \right\|, \quad X_p[\omega,\tau] = \angle X[\omega,\tau]$$

  - Zero the magnitudes of the unwanted noise

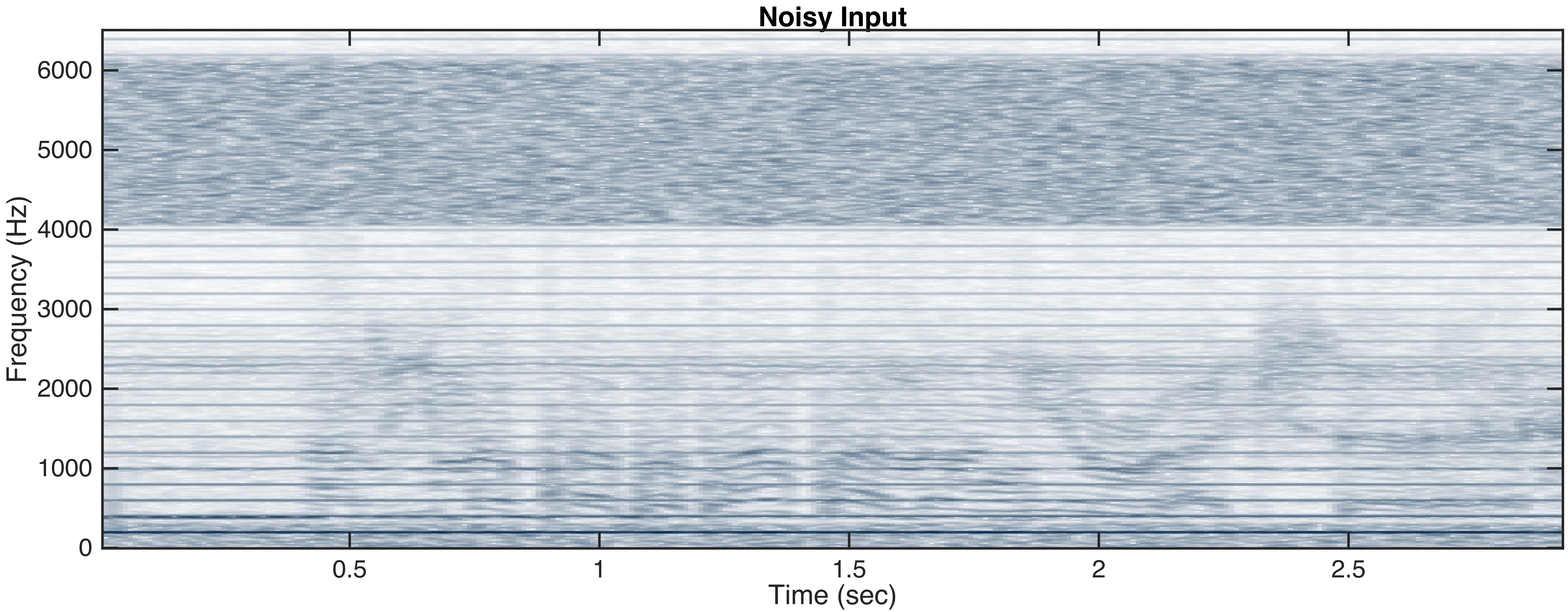
$$X_a[\omega_n,\tau] = 0, \quad \forall \omega_n \in noise$$

  - Resynthesize using the original phase

$$X[\omega,\tau] = X_a[\omega,\tau] e^{jX_p[\omega,\tau]}$$

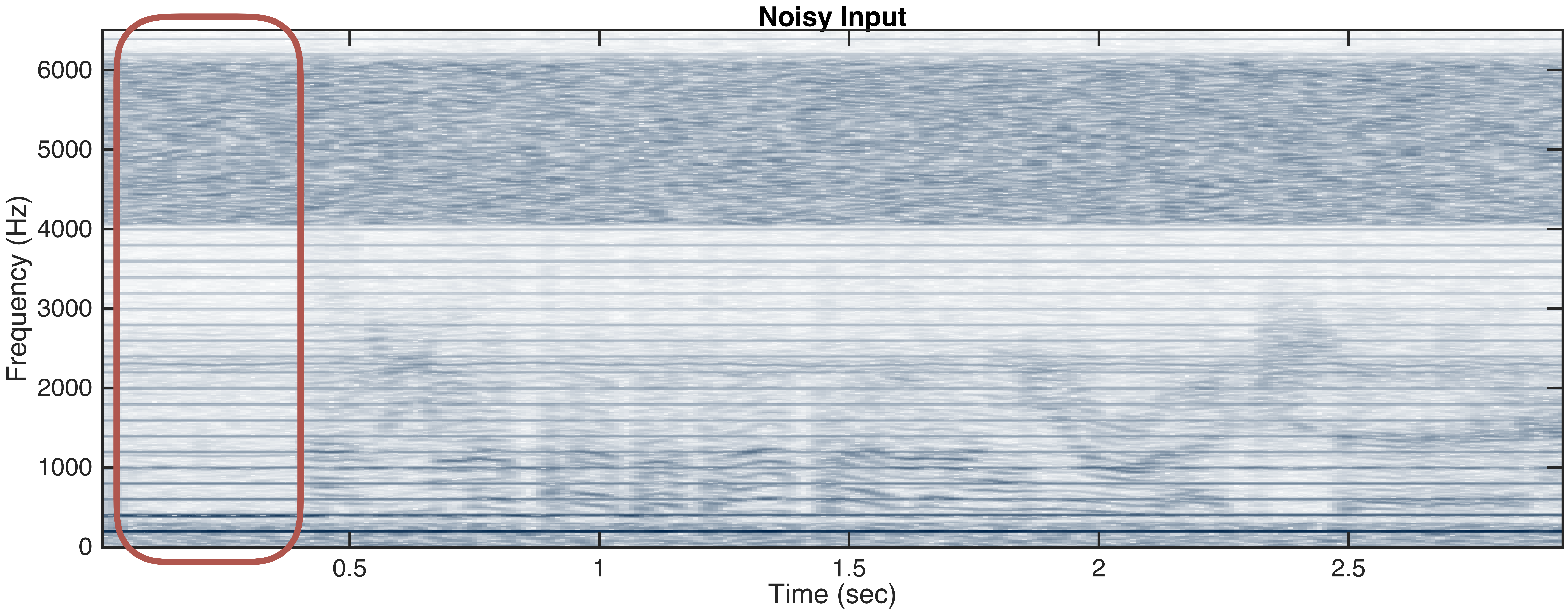
Where is the noise in the spectrogram?

- How do we find the noise frequencies?
  - And how do we remove them? (by hand?)



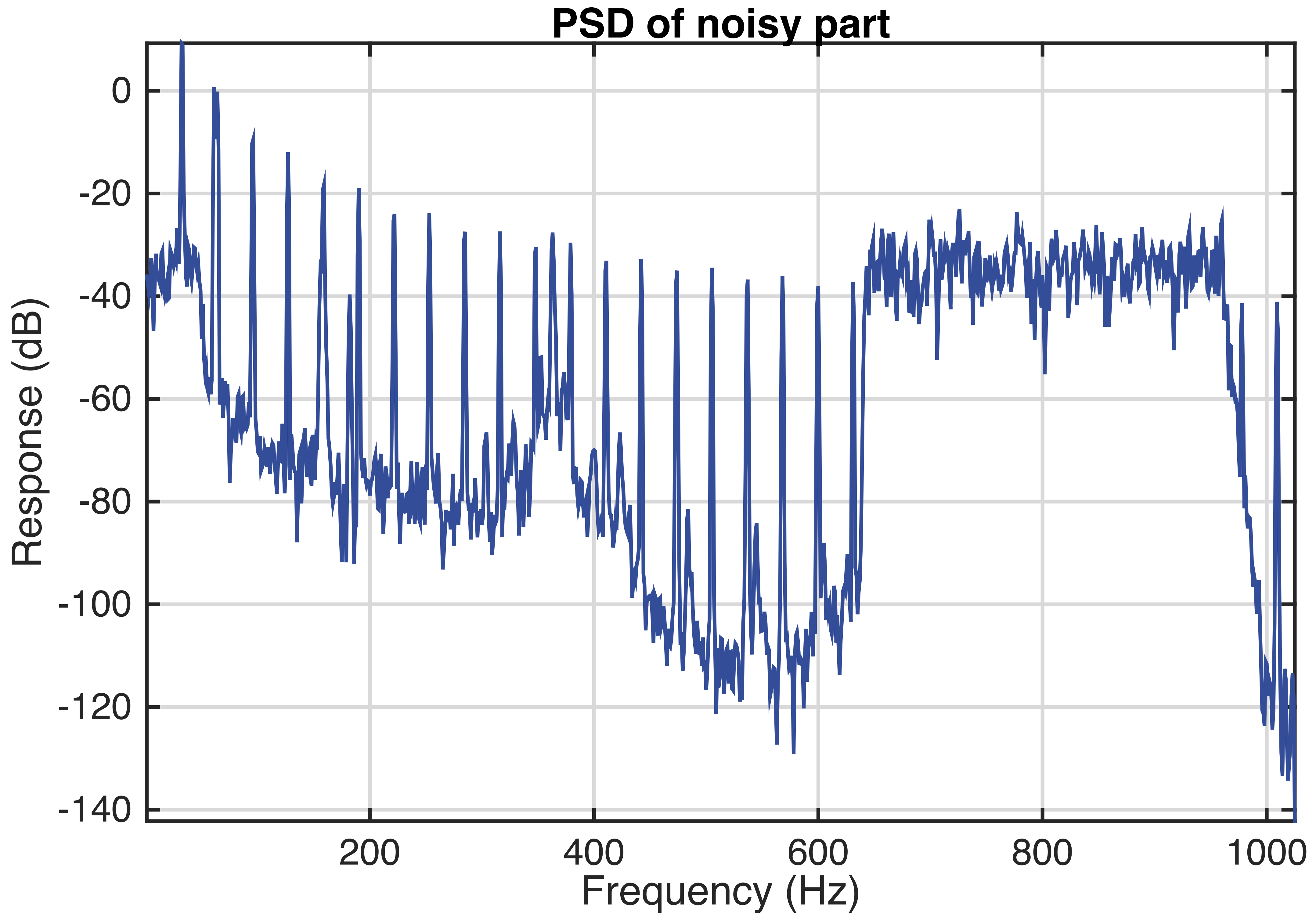
First take: Use an only-noise section

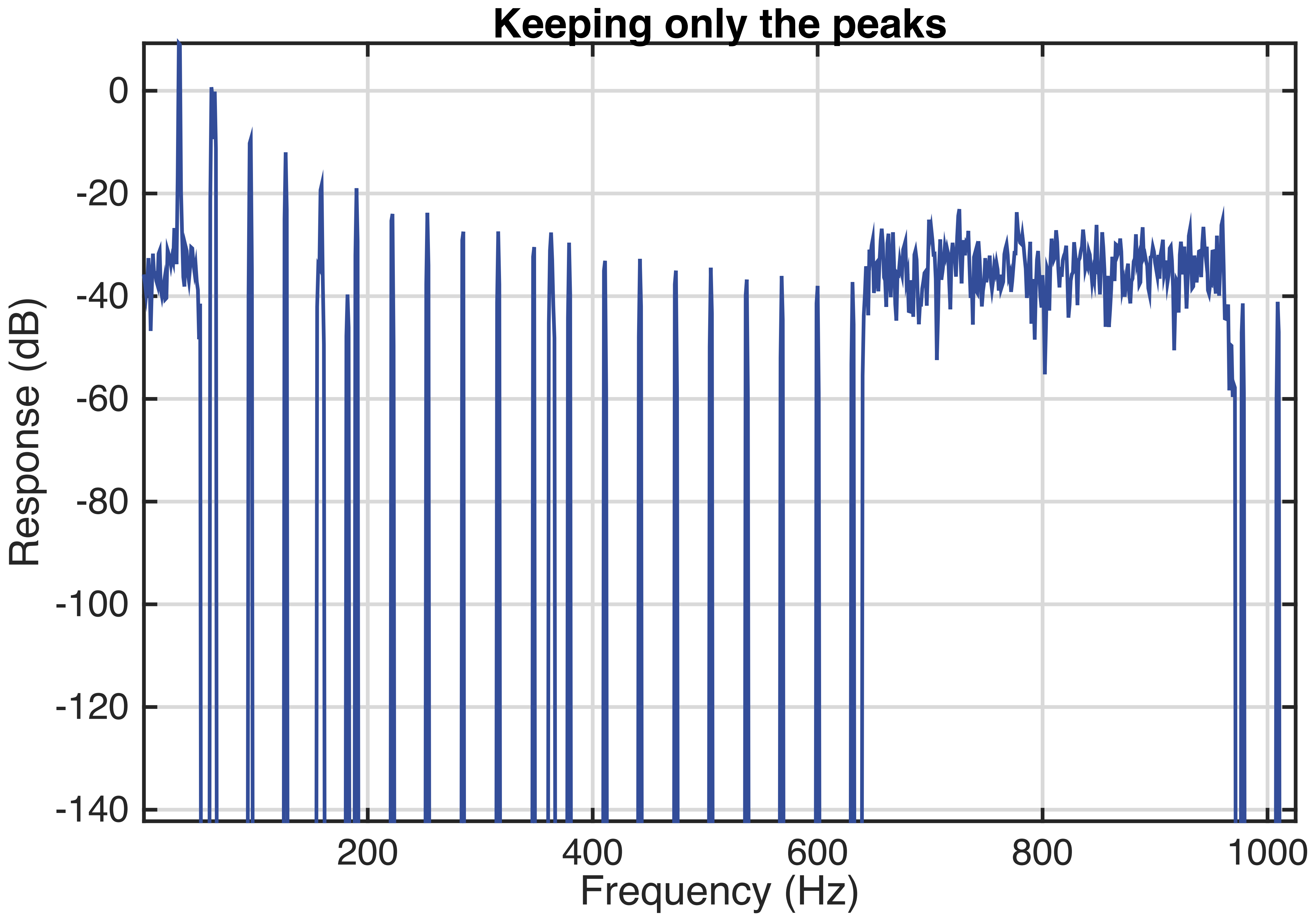
- There are sections in the recording which are noise-only and can reveal the noise characteristics
  - Find their large magnitudes and assume they are the noise



Finding the noise pattern

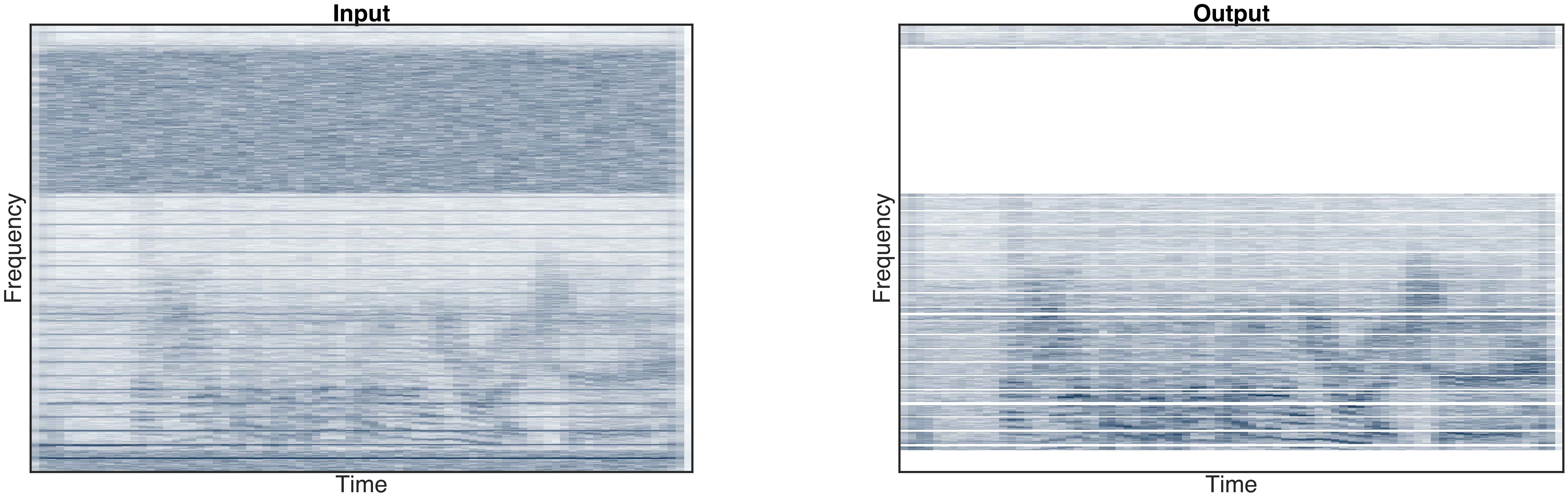
- Get the local peaks
  - These are the frequencies where noise is dominant
    - And the frequencies we want to silence





Result

- More accurate denoising
  - Noise is obliterated from spectrogram
    - Maybe a little too much?



A tunable approach

- Instead of setting unwanted components to zero, we can simply suppress them

- Advantages
  - We can select how much to remove them
  - We won't need to find the noise peaks
    - Simpler!

Spectral subtraction

- Subtract noise spectrum from input:

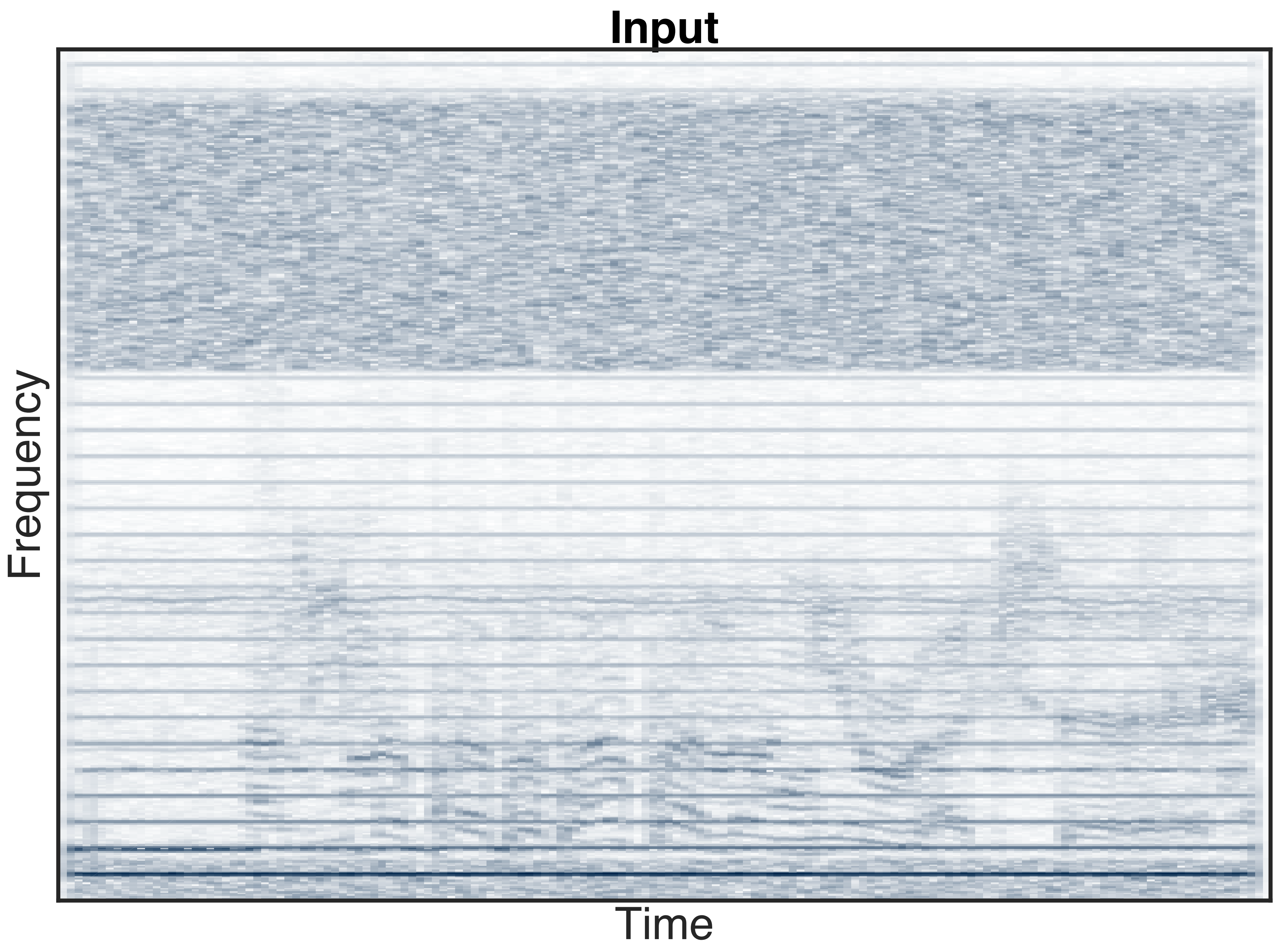
$$Y_t[\omega] = \left(\left\| X_t[\omega] \right\| - \alpha \left\| N[\omega] \right\| \right) e^{j\angle X_t[\omega]}$$

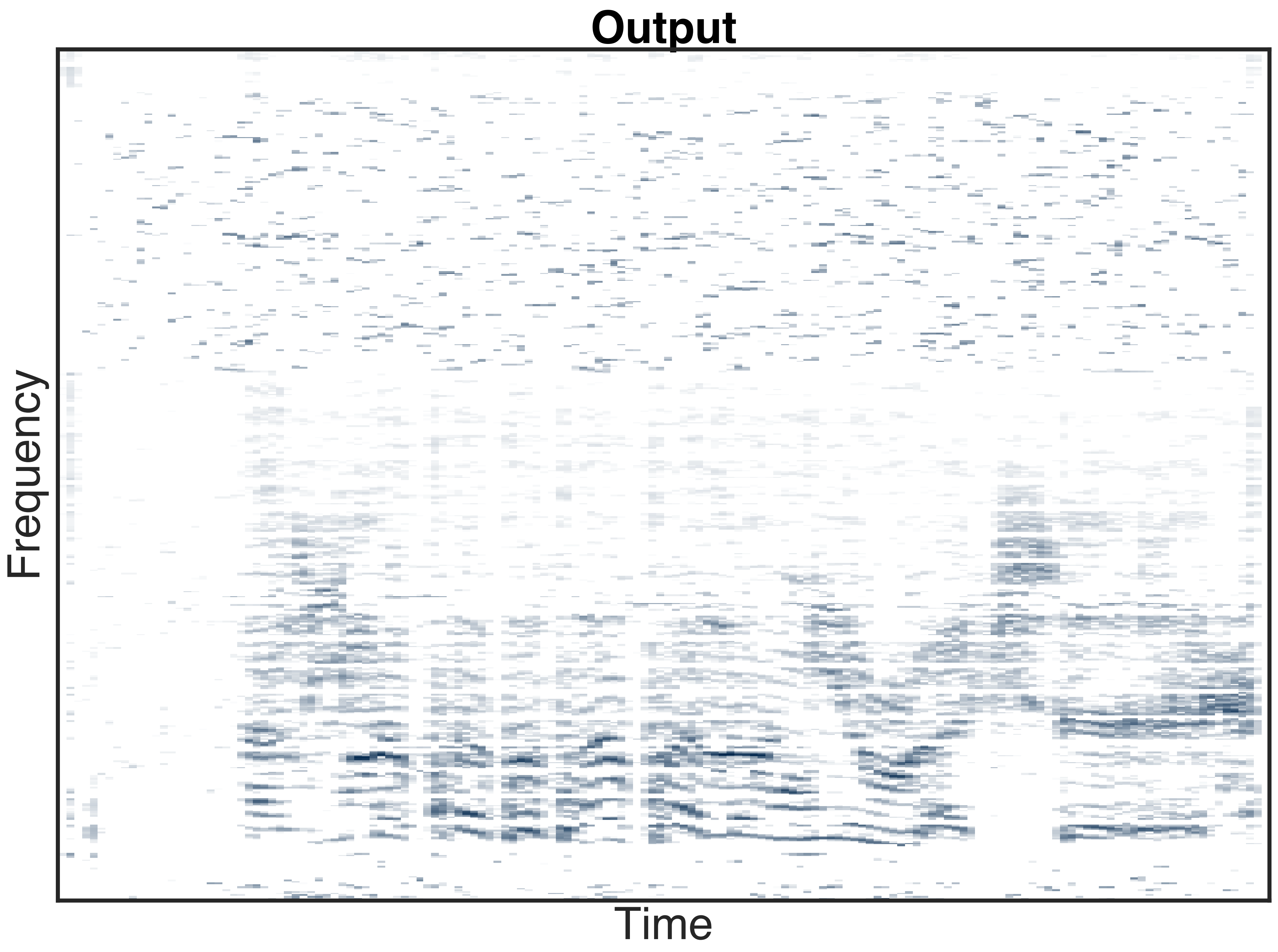
  - The parameter a defines how much noise we want to remove

- What happens when the result is less than zero?
  - For now, we can set negative values to zero
    - Why? If we use negative values we won't mute these frequencies!

Example

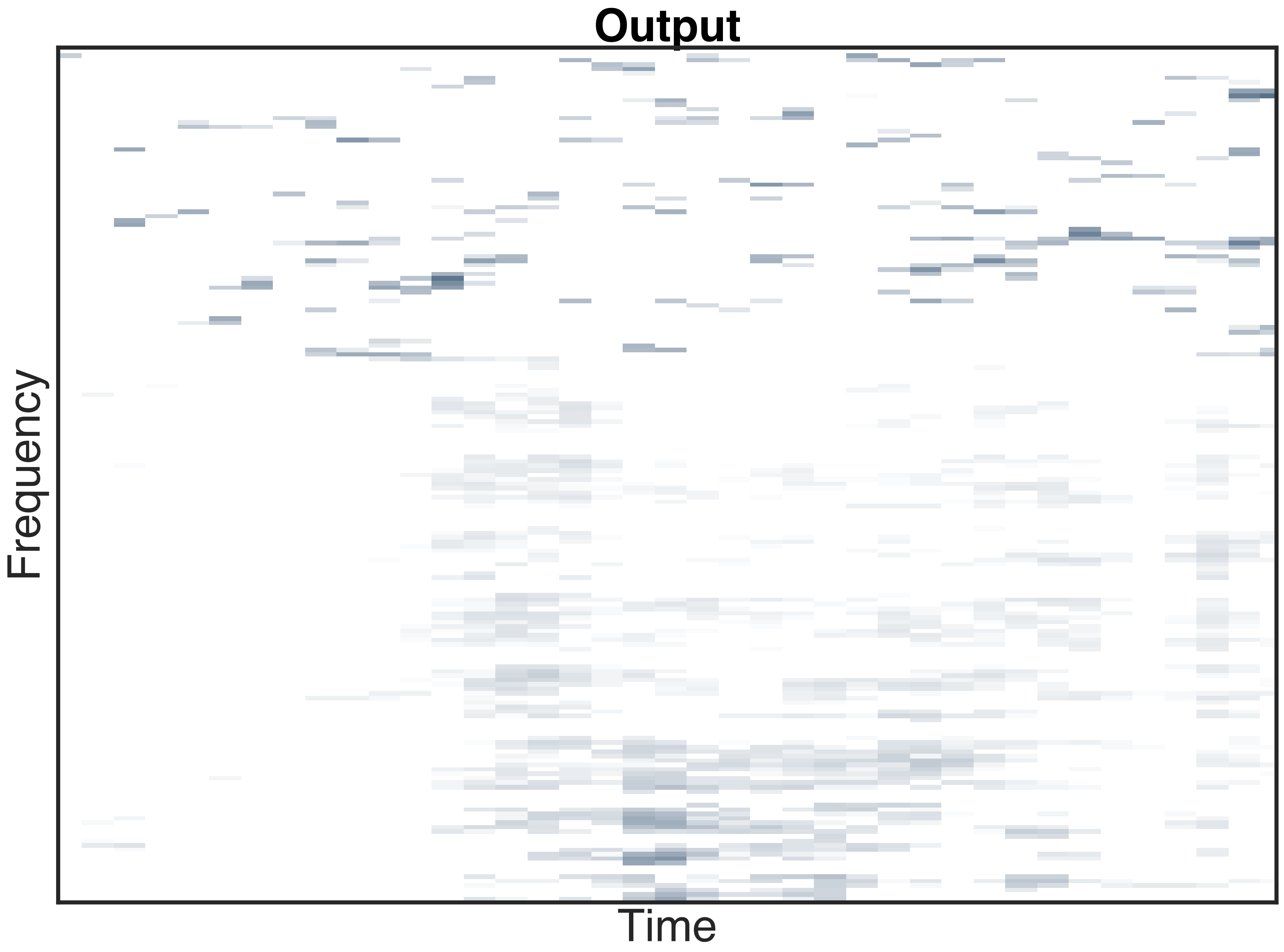
- Subtracting the noise spectrum removes noise!
  - But leaves some strange artifacts (can you explain them?)





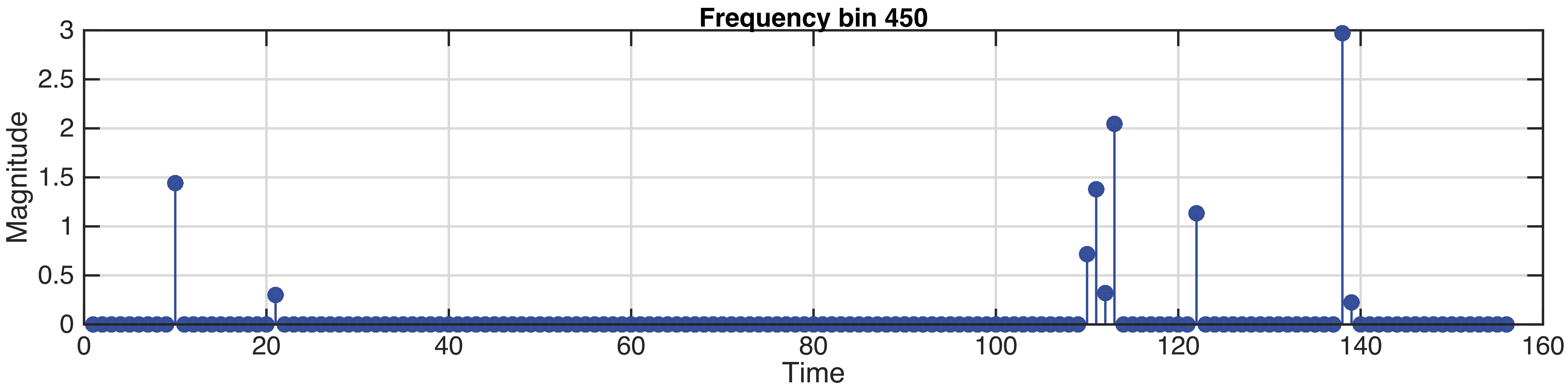
"Musical noise"

- The bleepy artifacts that remain
  - Frequencies that get turned off and on rapidly



Removing musical noise

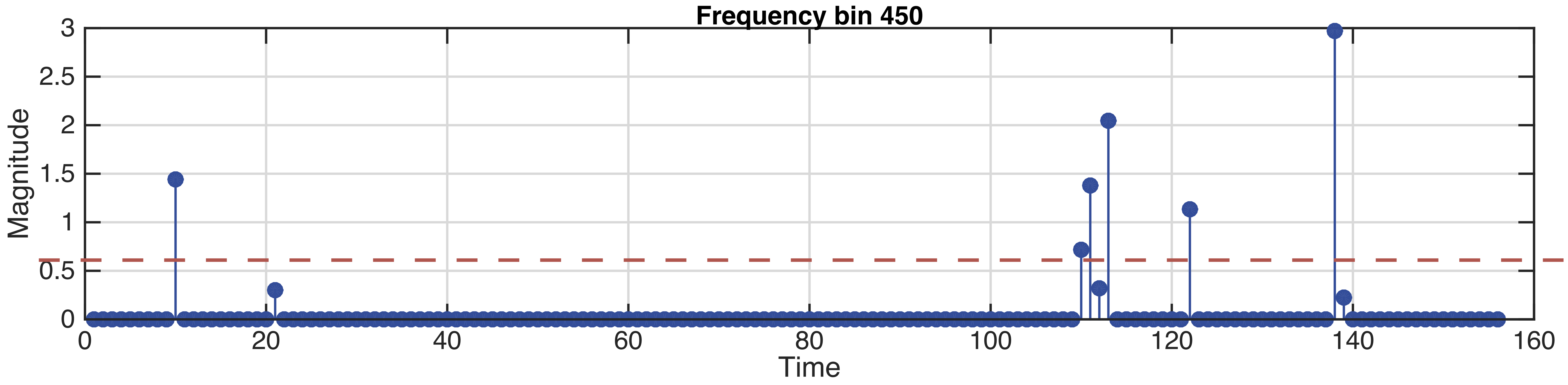
- Looking at a single frequency band



- How do we minimize these ups and downs?
  - We can use some heuristics

Some simple rules

- To keep a coefficient
  - It must exceed a certain threshold
  - It must be part of a longer stretch of activity



- But that's a little hacky

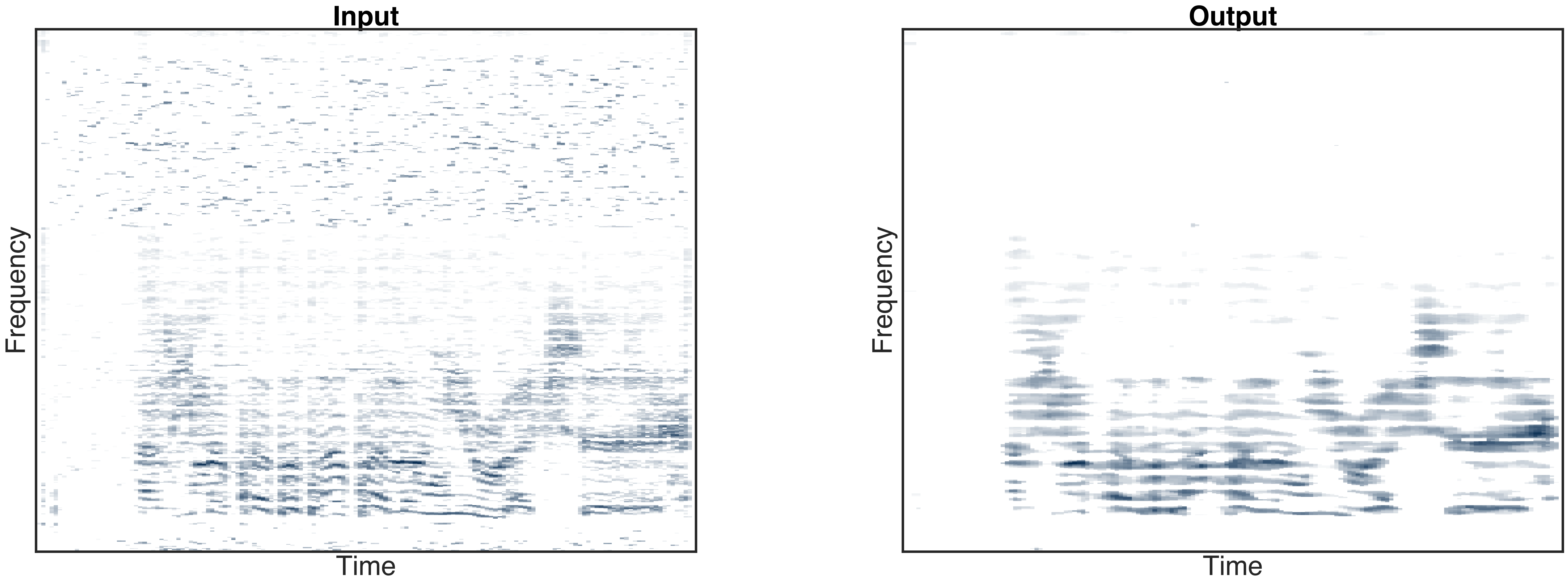
Median filtering

- A simpler approach
  - Median filtering
- Replace each STFT bin with a local median
  - Removes outlier points

$$\begin{matrix} 0 & 0 & \tfrac{1}{2} \\ 1 & \color{red}{1} & 0 \\ 0 & 0 & 0 \end{matrix} \quad \rightarrow \quad \begin{matrix} 0 & 0 & \tfrac{1}{2} \\ 1 & \color{red}{0} & 0 \\ 0 & 0 & 0 \end{matrix}$$

The result

- The speckles that form the musical noise can be significantly suppressed using this approach
  - But we still alter the sound a bit



Power version

- We can also subtract the spectral power

$$Y_t[\omega] = \sqrt{\left(\left\| X_t[\omega] \right\|^2 - \alpha \left\| N[\omega] \right\|^2 \right)} e^{j\angle X_t[\omega]}$$

- Or use a generalized form:

$$Y_t[\omega] = \sqrt[p]{\left(\left\| X_t[\omega] \right\|^p - \alpha \left\| N[\omega] \right\|^p \right)} e^{j\angle X_t[\omega]}$$

Spectral subtraction as a filter

- We can express this process as a linear filter:

$$X_t[\omega] \approx S_t[\omega] + N[\omega]$$

$$Y_t[\omega] = G_t[\omega] X_t[\omega] = g\Big(X_t[\omega], N[\omega]\Big) X_t[\omega]$$

- We can use a *gain function*, e.g.:

$$G_t[\omega] = 1 - \frac{\left\|N[\omega]\right\|}{\left\|X_t[\omega]\right\|}$$

$$\Rightarrow Y_t[\omega] = \left(1 - \frac{\left\|N[\omega]\right\|}{\left\|X_t[\omega]\right\|}\right) X_t[\omega]$$

Some common gain functions

Magnitude subtraction

$$G_t[\omega] = 1 - \frac{\left\|N[\omega]\right\|}{\left\|X_t[\omega]\right\|}$$

Wiener filter

$$G_t[\omega] = 1 - \frac{\left\|N[\omega]\right\|^2}{\left\|X_t[\omega]\right\|^2}$$

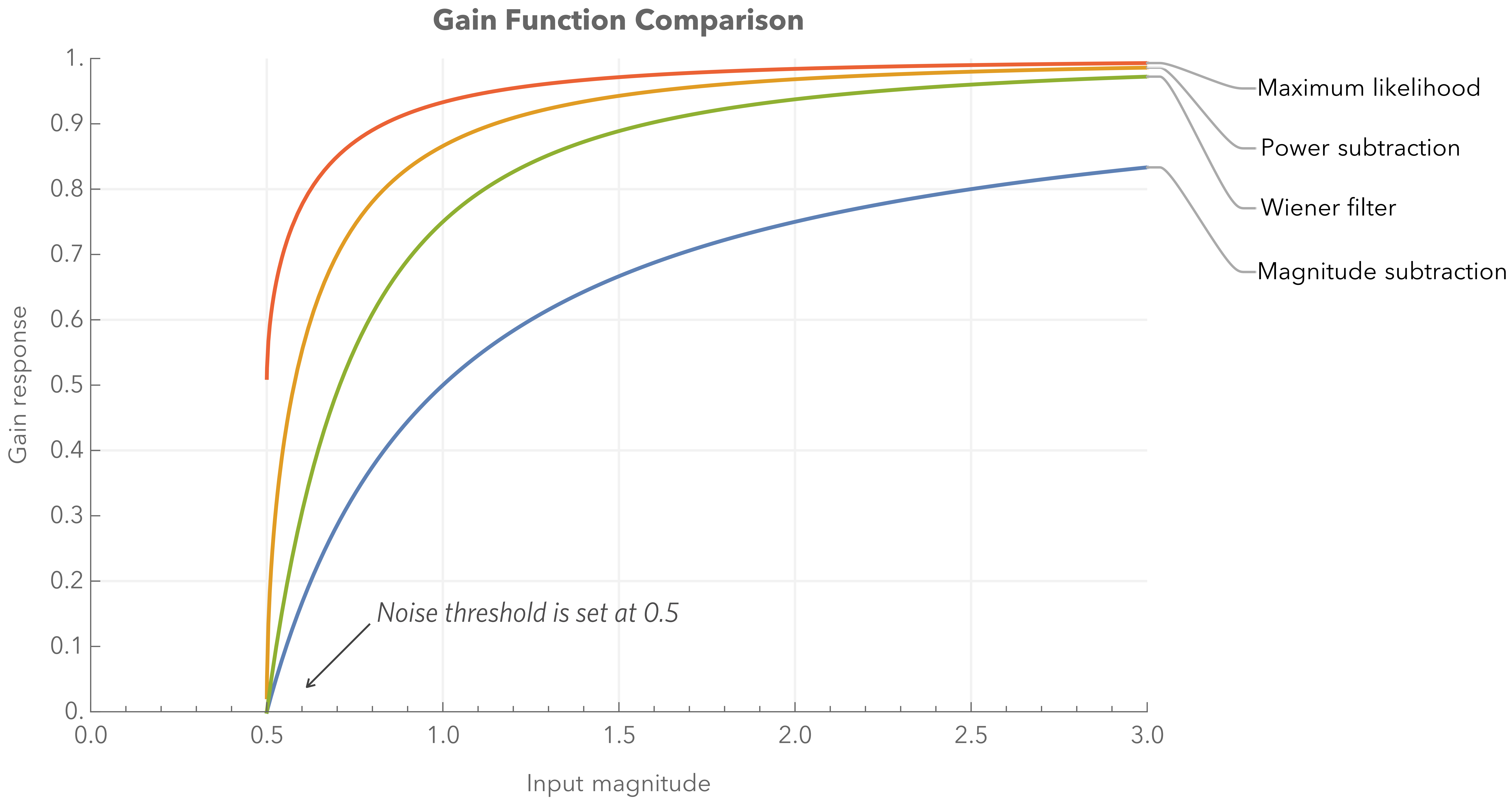
Power subtraction

$$G_t[\omega] = \sqrt{1 - \frac{\left\|N[\omega]\right\|^2}{\left\|X_t[\omega]\right\|^2}}$$

Maximum likelihood

$$G_t[\omega] = \frac{1}{2}\left[1 + \sqrt{1 - \frac{\left\|N[\omega]\right\|^2}{\left\|X_t[\omega]\right\|^2}}\right]$$

Comparing gain functions

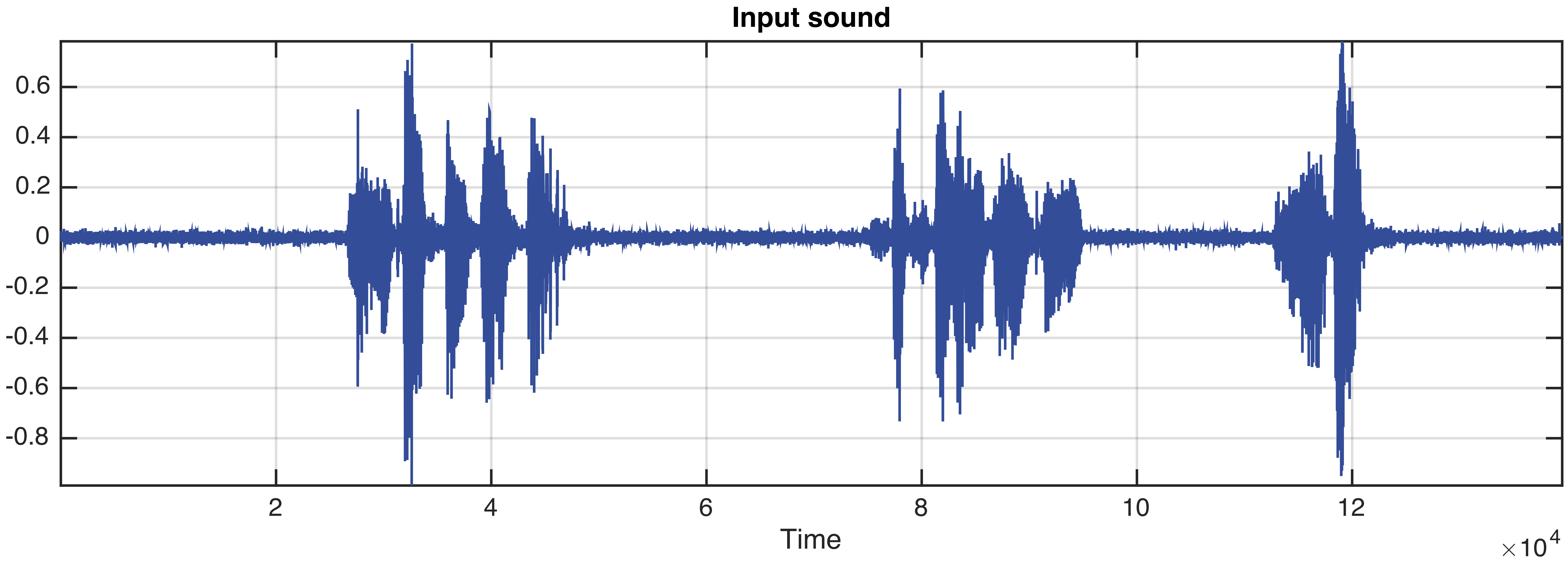


One lingering question

- How do we learn the noise profile?
  - How do we know where we have only noise?
- Not a trivial operation, hard to generalize
  - Two approaches:
    - Ask the user!
    - Use *Voice Activity Detection* (VAD)
      - (if you are denoising voice recordings)

Simple example

- Where is the noise?

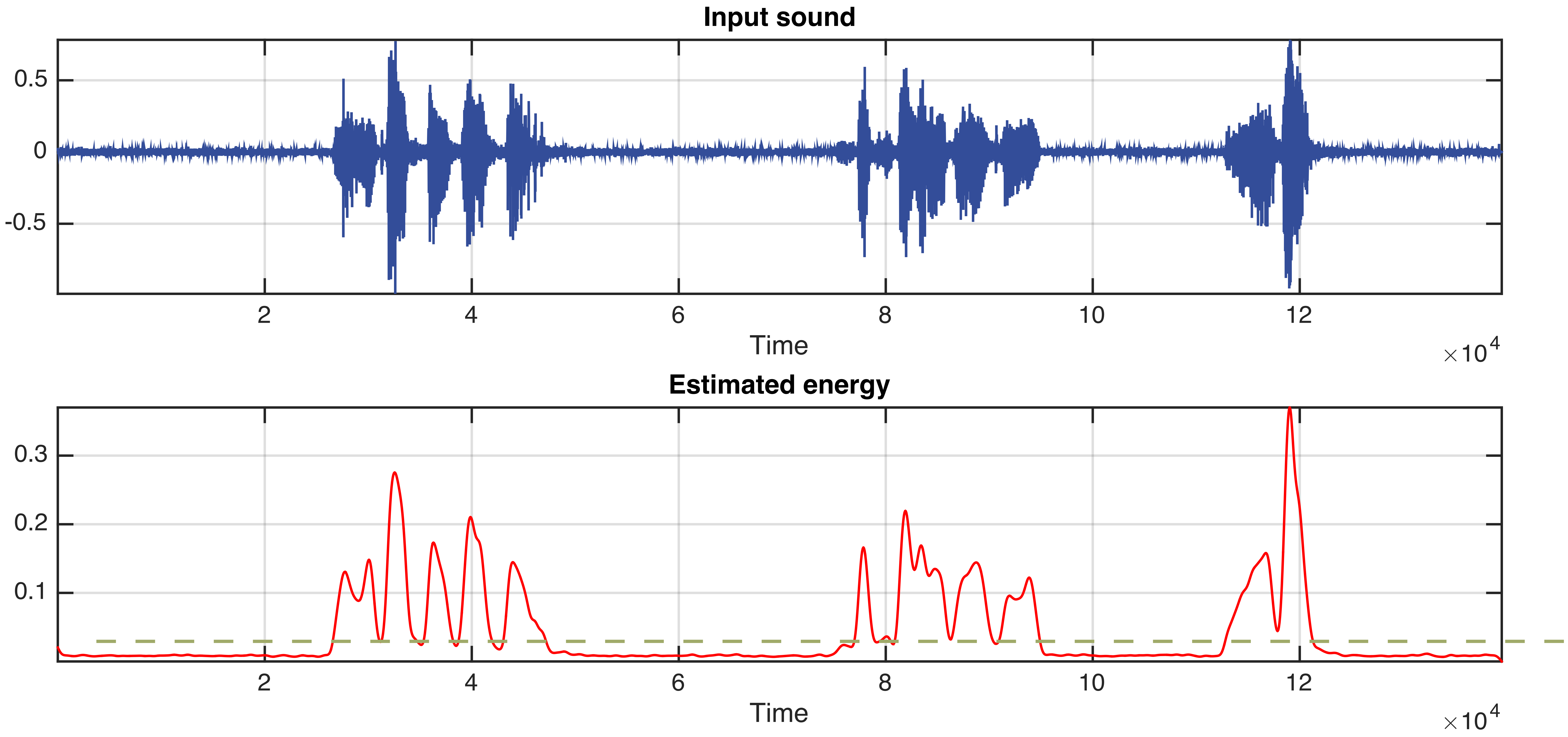


Voice Activity Detection

- Assume that speech is louder than background noise
  - Track only parts that contain high energy
- Simple approach
  - Rectify and lowpass filter the time domain signal
  - Results in a smooth energy contour

Example

- Use a threshold to define speech activity regions



Or we can use more complex ways

- E.g. G.729 standard
  - Line Spectral Frequencies (LSFs)
  - Full band energy
  - Low band energy
  - Zero-crossing rate
  - Full and low band energy differences
  - …

- Complex decision rule aggregating all of the above

Or we can use classifiers

- We can learn a speech model
  - Use training data to learn speech statistics
    - E.g. mean and covariance of speech spectra
- Classify each input frame according to model
  - If classified as non-speech then it's noise
- More later in the semester

Denoising with VAD

- Two main advantages
  - We can learn a noise model
    - Update the noise spectrum when there’s no speech
  - We can ignore the noise parts
    - When you don’t detect speech, set output to zero

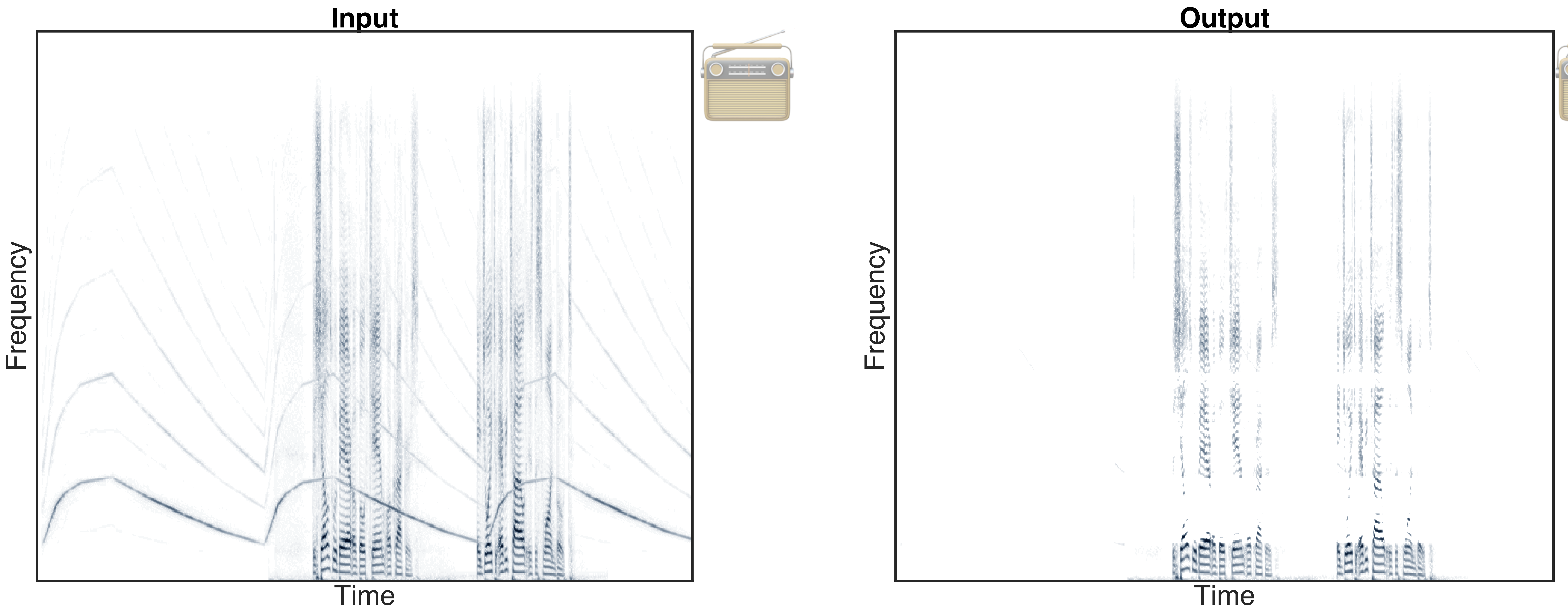
- VADs are integral parts of speech technology

Stationary noise vs. not

- What will happen if the noise keeps changing?
  - E.g. a siren?
- Noise model is not capable of tracking that
- Spectral subtraction is for *stationary* noise
  - When dealing with dynamic noises it will fail miserably

Example of non-stationary noise

- The noise sweeps all over a wide range
  - Denoiser gets to remove lots of speech



Multichannel methods

- Classical array approaches
  - Beamforming
    - Focus on speech directions (use localization)
    - Minimize energy of non-speech directions
    - We've already done this in Lab 3
- Alternative approaches
  - Use noise model from extra microphone
    - E.g. dual microphone cell-phones
    - Eliminate the need for a VAD

Dual microphone denoising

- One mic picks mostly voice, the other mostly noise

$$V_t[\omega] = S_t[\omega] + \alpha N_t[\omega], \quad 0 < \alpha < 1$$

$$E_t[\omega] = \beta S_t[\omega] + N_t[\omega], \quad 0 < \beta < 1$$

- Denoised output:

$$Y_t[\omega] = \left(\left| V_t[\omega] \right| - \gamma \left| E_t[\omega] \right| \right) e^{j \angle V_t[\omega]}$$

- When will this fail?

Measuring noise reduction

- How do we evaluate denoising?
- Not straightforward
  - Noise reduction is subjective, and task-dependent
- Two takes on this:
  - Get a *Mean Opinion Score* (MOS) from listeners
  - Use standardized measures

Measuring noise reduction

- Standard metric: *Signal to Noise Ratio* (SNR)

$$\mathrm{SNR} = 10\log_{10} \frac{\left\|signal\right\|^2}{\left\|noise\right\|^2}$$

- One problem:
  - You need to know the clean signal
- Other metrics as well
  - SDR, SAR, …

What about other noise types?

- Recording artifacts?
  - Vinyl scratches/cracks, tape hiss?
- Speech contaminated by background speech?
  - Or complex non-stationary noises?
- More on these later

Recap

- Denoising with filters
- Spectral subtraction
  - Estimating noise profiles
  - Minimizing musical noise
- Multichannel methods
- Measuring denoising performance

Reading material

- Vaseghi's spectral subtraction book:
  - http://dsp-book.narod.ru/304.pdf

Next lab

- Denoising!
  - Implementing spectral subtraction