Predictability of Signals

Mauricio D Sacchi
SAIG and University of Alberta

msacchi@ualberta.ca

Key References:

- Canales,1984, Random noise reduction: 54th Annual International Meeting, SEG, Expanded Abstracts, 525–527.
- Soubaras, 1994, Signal preserving random noise attenuation by the f-x projection: 64th Annual International Meeting, SEG, Expanded Abstracts, 1576–1579.
- Spitz, 1991, Seismic trace interpolation in the FX domain , GEOPHYSICS, 56(8)
- Gülünay, 2001, Signal leakage in f-x deconvolution algorithms, GEOPHYSICS, 82(5), W31–W45.

Introduction

- Prediction filters have a long history in exploration seismology
  - min phase statistical deconvolution (spiking decon) is a prediction error filter
  - SNR enhancement (FX-Deconvolution) (Canales, 1984)
  - Signal interpolation including regular upsampling and irregular signal reconstruction in FX domain (Spitz, 1991)

Predictability of Complex Sinusoids

- Consider a simple harmonic signal

$$s_n = Ae^{i\alpha n}, \qquad n = 0 \ldots N-1$$

- It is easy to show the following

$$s_n = as_{n-1}, \quad a = e^{i\alpha}$$

- Hence, there is an operator $(1, -a)$ called an annihilator such that $s_n - as_{n-1} = (1, -a) * s_n = 0$ which means the complex sinusoid is predictable. This is also called Forward Prediction.

- You can also say that the signal can be compressed because one can recover the complete signal knowing only s_0 and the coefficient a.

Predictability of Complex Sinusoids

- Consider a simple harmonic signal

$$s_n = Ae^{i\alpha n}, \qquad n = 0 \ldots N-1$$

- It is easy to show the following

$$s_{n-1} = bs_n, \quad b = e^{-i\alpha} = a*$$

- Hence, we can also do backward prediction.

Predictability of Complex Sinusoids

- Consider a sum of two harmonic signals

$$s_n = A_1 e^{i\alpha_1 n} + A_2 e^{i\alpha_2 n} \qquad n = 0 \ldots N-1$$

- It is easy to show the following

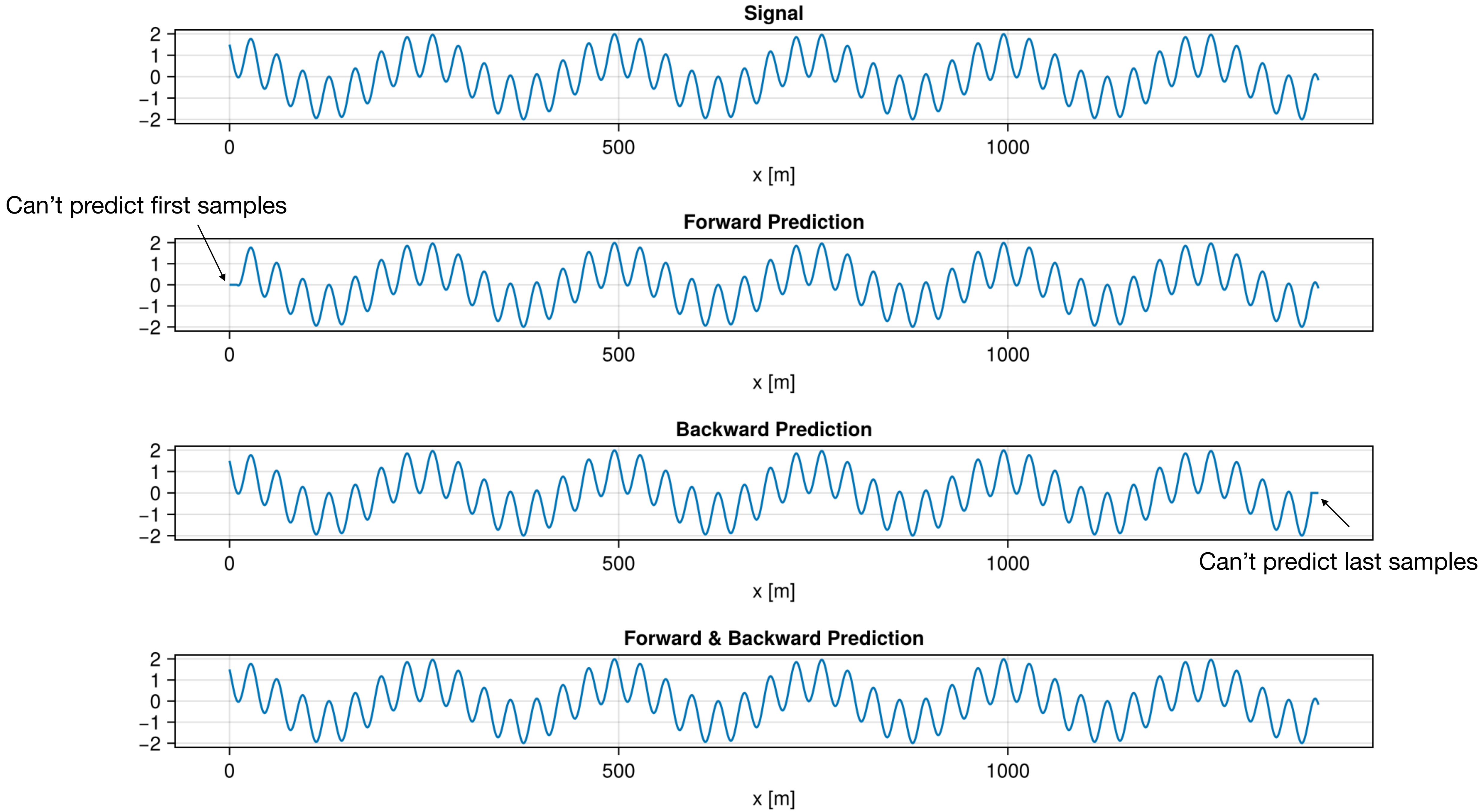
$$s_n = a_1 s_{n-1} + a_2 s_{n-2}$$
$$s_n = b_1 s_{n+1} + b_2 s_{n+2}$$

Similarly, the annihilator is now $(1, -a_1, -a_2)$ (Forward) and $(1, -b_1, -b_2)$ (Backward)

- The general case corresponds to a the supervision of p harmonics in which case

$$s_n = a_1 s_{n-1} + a_2 s_{n-2} + a_3 s_{n-3} + \ldots a_p s_{n-p}$$
$$s_n = b_1 s_{n+1} + b_2 s_{n+2} + b_3 s_{n+3} + \ldots a_p s_{n+p}$$

2 complex exponentials, no noise



AR models

- AR(p) : Autoregressive model of order p (Forward)

$$s_n = a_1 s_{n-1} + a_2 s_{n-2} + a_3 s_{n-3} + \ldots a_p s_{n-p} + \epsilon_n$$

- ϵ_n is a random innovation (not to be confused with additive noise). One can rewrite the latter in Prediction error form

$$\epsilon_n = s_n - a_1 s_{n-1} - a_2 s_{n-2} - a_3 s_{n-3} \ldots - a_p s_{n-p}$$

- Similarly, we can rewrite the latter as follows

$$\epsilon_n = f_0 s_n + f_1 s_{n-1} + f_2 s_{n-2} + f_3 s_{n-3} \ldots + f_p s_{n-p}$$

$$f_0 = 1, \qquad f_n = -a_n, \qquad n = 1 \ldots p$$

AR models

- AR(p) : Autoregressive model of order p (Backward)

$$s_n = b_1 s_{n+1} + b_2 s_{n+2} + b_3 s_{n+3} + \ldots b_p s_{n+p} + \epsilon_n$$

- ϵ_n is a random innovation (not to be confused with additive noise). One can rewrite the latter in Prediction error form

$$\epsilon_n = s_n - b_1 s_{n+1} - b_2 s_{n+2} - b_3 s_{n+3} \ldots - b_p s_{n+p}$$

- Similarly, we can rewrite the latter as follows

$$\epsilon_n = f_0 s_n + f_1 s_{n+1} + f_2 s_{n+2} + f_3 s_{n+3} \ldots + f_p s_{n+p}$$

$$f_0 = 1, \qquad f_n = -b_n, \qquad n = 1 \ldots p$$

Example

$$p = 3, N = 7, s = (s_0, s_1, s_2, s_3, s_4, s_5, s_6)$$

$$\begin{aligned} s_3 &= a_1 s_2 + a_2 s_1 + a_1 s_0 + \epsilon_3 \\ s_4 &= a_1 s_3 + a_2 s_2 + a_1 s_1 + \epsilon_4 \\ s_5 &= a_1 s_4 + a_2 s_3 + a_1 s_2 + \epsilon_5 \\ s_6 &= a_1 s_5 + a_2 s_4 + a_1 s_3 + \epsilon_6 \end{aligned} \Rightarrow \begin{pmatrix} s_3 \\ s_4 \\ s_5 \\ s_6 \end{pmatrix} = \begin{pmatrix} s_2 & s_1 & s_0 \\ s_3 & s_2 & s_1 \\ s_4 & s_3 & s_2 \\ s_5 & s_4 & s_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix}$$

In matrix form: $\mathbf{s}_f = \mathbf{M}_f \, \mathbf{a} + \mathbf{e}_f$

Least-squares Prediction Filter $\hat{\mathbf{a}} = (\mathbf{M}_f^H \mathbf{M}_f + \mu \mathbf{I})^{-1} \mathbf{M}_f^H \mathbf{s}_f$

Predicted signal: $\hat{\mathbf{s}}_f = \mathbf{M}_f \, \hat{\mathbf{a}}$ (not predicting s_0, s_1, s_2)

*** To discuss: It can be viewed as a learning process*

Example

$$p = 3, N = 7, s = (s_0, s_1, s_2, s_3, s_4, s_5, s_6)$$

$$s_0 = b_1 s_1 + b_2 s_2 + b_1 s_3 + \epsilon_0$$
$$s_1 = b_1 s_2 + b_2 s_3 + b_1 s_4 + \epsilon_1$$
$$s_2 = b_1 s_3 + b_2 s_4 + b_1 s_5 + \epsilon_2$$
$$s_3 = b_1 s_4 + b_2 s_5 + b_1 s_6 + \epsilon_3$$

$$\Rightarrow \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \\ s_3 & s_4 & s_5 \\ s_4 & s_5 & s_6 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}$$

In matrix form: $\mathbf{s}_b = \mathbf{M}_b \, \mathbf{b} + \mathbf{e}_b$

Least-squares Prediction Filter $\hat{\mathbf{b}} = (\mathbf{M}_b^H \mathbf{M}_b + \mu \mathbf{I})^{-1} \mathbf{M}_b^H \mathbf{s}_b$

Predicted signal: $\hat{\mathbf{s}}_b = \mathbf{M}_b \, \hat{\mathbf{b}}$ (not predicting s_4, s_5, s_6)

Two approaches

Main goal is to denoise signals exploding signal predictability

- 1) Forward and backward independent operators
- 2) Solve simultaneously for one operator for forward and backward prediction using ($\mathbf{a} = \mathbf{b}*$)

Solve for forward and backward prediction

$$\begin{pmatrix} s_3 \\ s_4 \\ s_5 \\ s_6 \end{pmatrix} = \begin{pmatrix} s_2 & s_1 & s_0 \\ s_3 & s_2 & s_1 \\ s_4 & s_3 & s_2 \\ s_5 & s_4 & s_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix}$$

$$\hat{\mathbf{a}} = (\mathbf{M}_f^H \mathbf{M}_f + \mu \mathbf{I})^{-1} \mathbf{M}_f^H \mathbf{s}_f$$

$$\hat{\mathbf{s}}_f = \mathbf{M}_f \, \hat{\mathbf{a}}$$

$$(\hat{s}_3 \;\; \hat{s}_4 \;\; \hat{s}_5 \;\; \hat{s}_6)$$

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \\ s_3 & s_4 & s_5 \\ s_4 & s_5 & s_6 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}$$

$$\hat{\mathbf{b}} = (\mathbf{M}_b^H \mathbf{M}_b + \mu \mathbf{I})^{-1} \mathbf{M}_b^H \mathbf{s}_b$$

$$\hat{\mathbf{s}}_b = \mathbf{M}_b \, \hat{\mathbf{b}}$$

$$(\hat{s}_0 \;\; \hat{s}_1 \;\; \hat{s}_2 \;\; \hat{s}_3)$$

$$(\hat{s}_0 \;\; \hat{s}_1 \;\; \hat{s}_2 \;\; \hat{s}_3 \;\; \hat{s}_4 \;\; \hat{s}_5 \;\; \hat{s}_6)$$

Central repeated samples are averaged

Solve for one operator

$$\begin{pmatrix} s_3 \\ s_4 \\ s_5 \\ s_6 \end{pmatrix} = \begin{pmatrix} s_2 & s_1 & s_0 \\ s_3 & s_2 & s_1 \\ s_4 & s_3 & s_2 \\ s_5 & s_4 & s_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix}$$

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \\ s_3 & s_4 & s_5 \\ s_4 & s_5 & s_6 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}$$

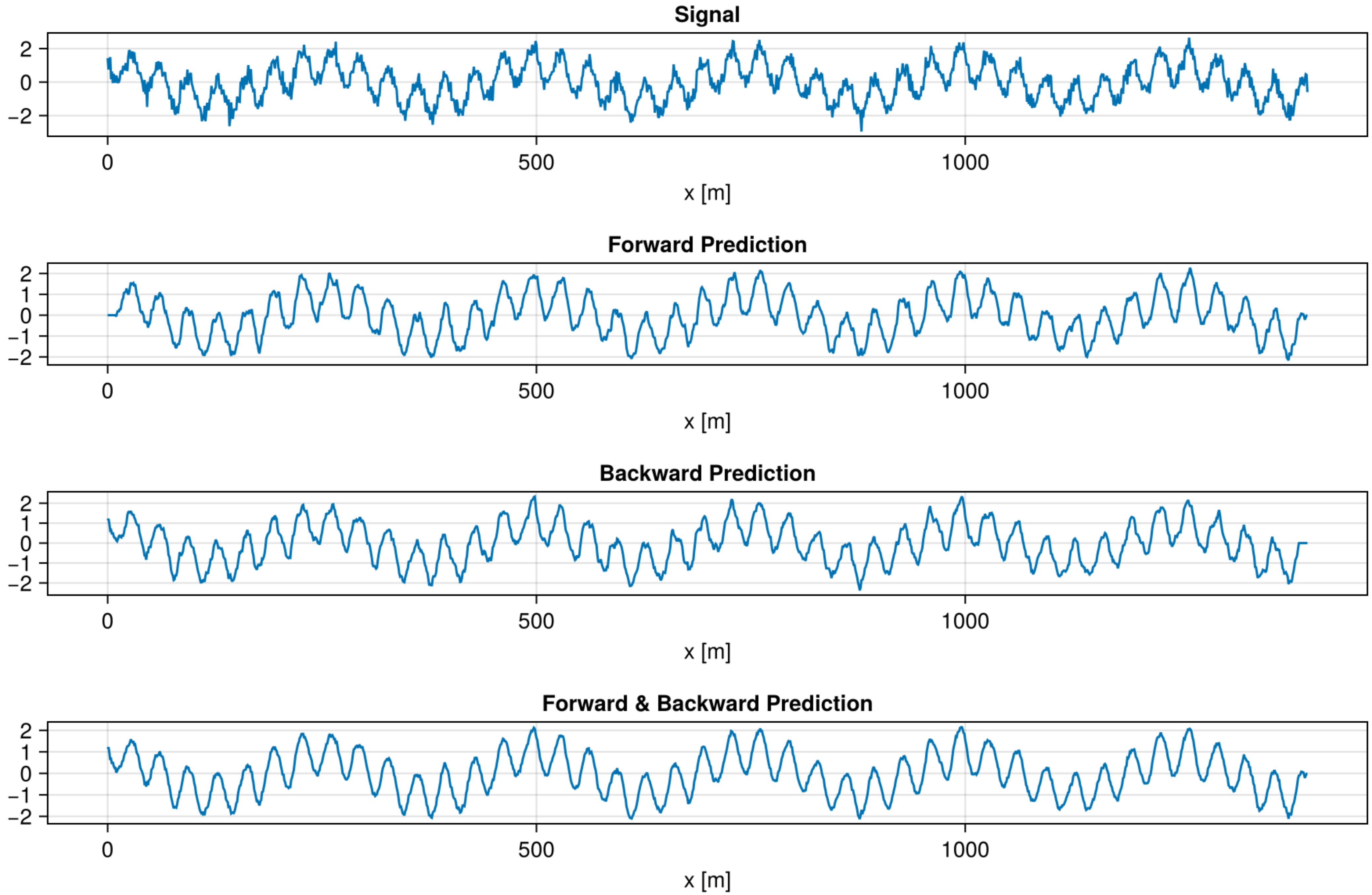
$$\mathbf{s}_f = \mathbf{M}_f\, \mathbf{a} + \mathbf{e}_f \qquad \mathbf{s}_b = \mathbf{M}_b\, \mathbf{a}^* + \mathbf{e}_b$$

Solve augmented system

$$\begin{pmatrix} \mathbf{s}_f \\ \mathbf{s}_b^* \end{pmatrix} = \begin{pmatrix} \mathbf{M}_f \\ \mathbf{M}_b^* \end{pmatrix} \mathbf{a} + \mathbf{e}$$

Estimate the operator and predict signal as before $(\hat{s}_0\ \hat{s}_1\ \hat{s}_2\ \hat{s}_3\ \hat{s}_4\ \hat{s}_5\ \hat{s}_6)$

2 complex exponentials + noise



Prediction Filter Inconsistency

- I have taken innovation as additive noise which is not correct.

- Soubaras proposed to alleviate the aforementioned inconsistency via the so-called Projection Filter

- Projection filter model: $\mathbf{d} = \mathbf{s} + \mathbf{n}$ and $\mathbf{e} = \mathbf{F}\mathbf{s}$

- Where $\mathbf{e} = \mathbf{F}\mathbf{s}$ mens the signal is predictable. In the case of perfect predictability $\mathbf{e} = 0$, otherwise $\mathbf{e}$ is an innovation.

- The observation is $\mathbf{d}$ and we want to estimate $\mathbf{s}$.

Projection filter

- Two goals

  1. $\mathbf{d} = \mathbf{s} + \mathbf{n} \longrightarrow \mathbf{d} \approx \mathbf{s}$

  2. $\mathbf{e} = \mathbf{F}\mathbf{s} \longrightarrow \mathbf{F}\mathbf{s} \approx \mathbf{0}$

- The last two goals are combined into a single cost function

$$J = \|\mathbf{s} - \mathbf{d}\|_2^2 + \mu \|\mathbf{F}\mathbf{s}\|_2^2$$

$$\nabla J = \mathbf{0} \longrightarrow (\mathbf{I} + \mu \mathbf{F}^H \mathbf{F})\mathbf{s} = \mathbf{d}$$

Projection Filter

$$\nabla J = \mathbf{0} \longrightarrow (\mathbf{I} + \mu \mathbf{F}^H \mathbf{F})\mathbf{s} = \mathbf{d}$$

Then

$$\hat{\mathbf{s}} = (\mathbf{I} + \mu \mathbf{F}^H \mathbf{F})^{-1}\mathbf{d}$$

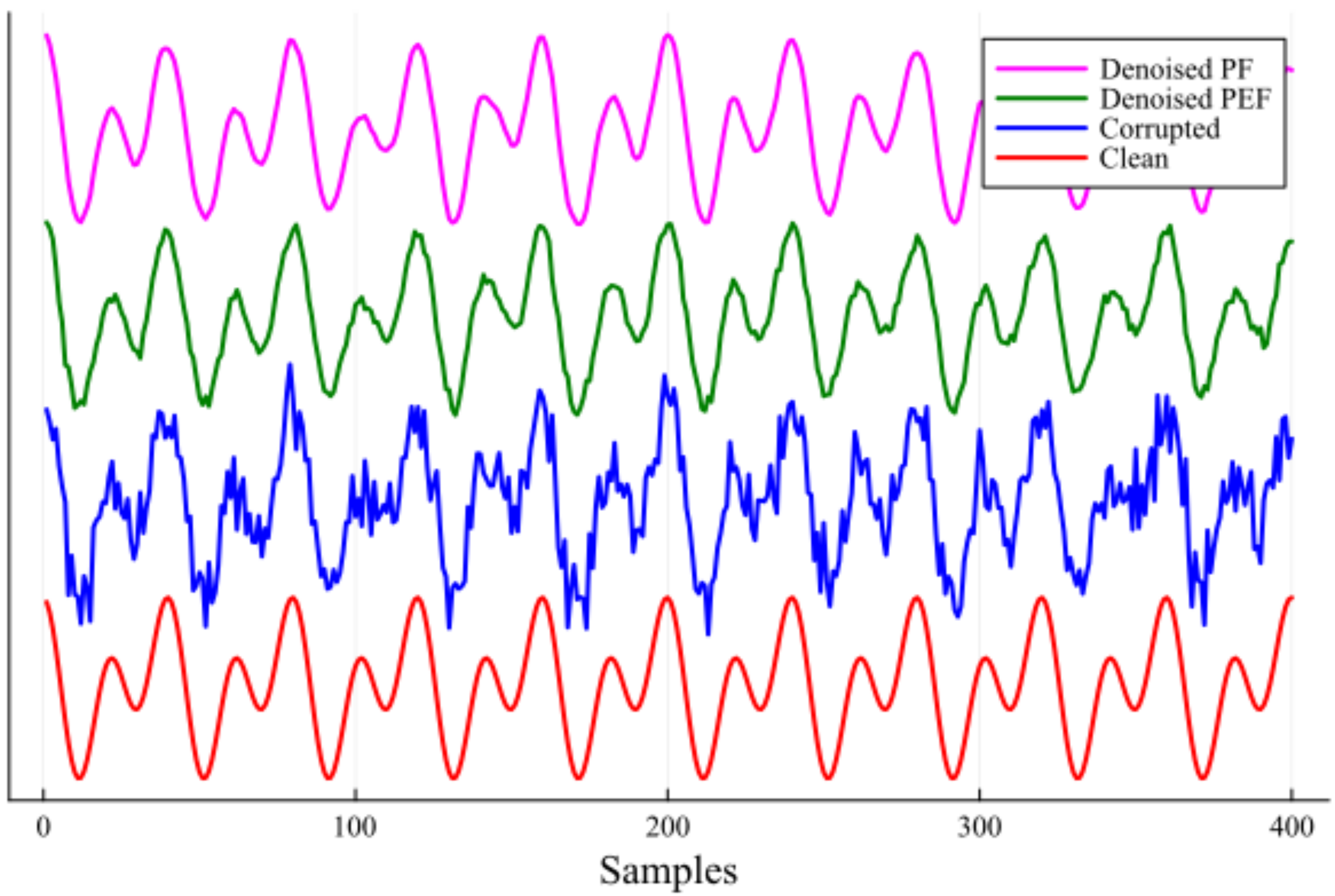
$$\hat{\mathbf{n}} = \mathbf{d} - (\mathbf{I} + \mu \mathbf{F}^H \mathbf{F})^{-1}\mathbf{d} = (\mathbf{I} - \mathbf{A})\mathbf{d} = \mathbf{P}\mathbf{d}$$

Interesting to note that $\mathbf{P}$ must behave in the frequency/wavenumber domain like a Notch Filter. But….. like in any inverse problem, there is no free lunch!

$$\begin{aligned} \hat{\mathbf{n}} &= \mathbf{P}\mathbf{d} \\ &= \mathbf{P}(\mathbf{s} + \mathbf{n}) \end{aligned}$$

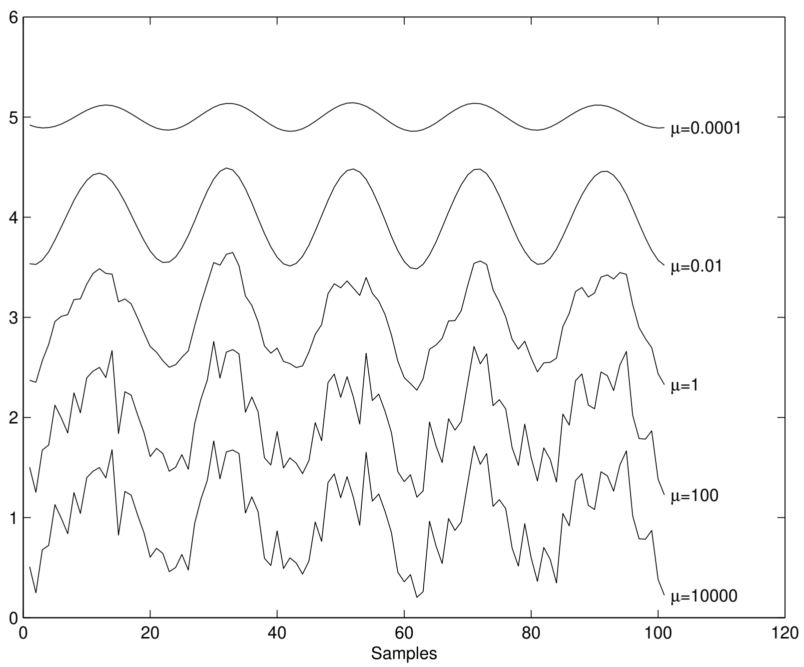
We can’t make $\mathbf{Pn} = \mathbf{n}$ and $\mathbf{Ps} = \mathbf{0}$ simultaneously.

Comparison prediction filter (PEF) vs. Projection Filter (PF)

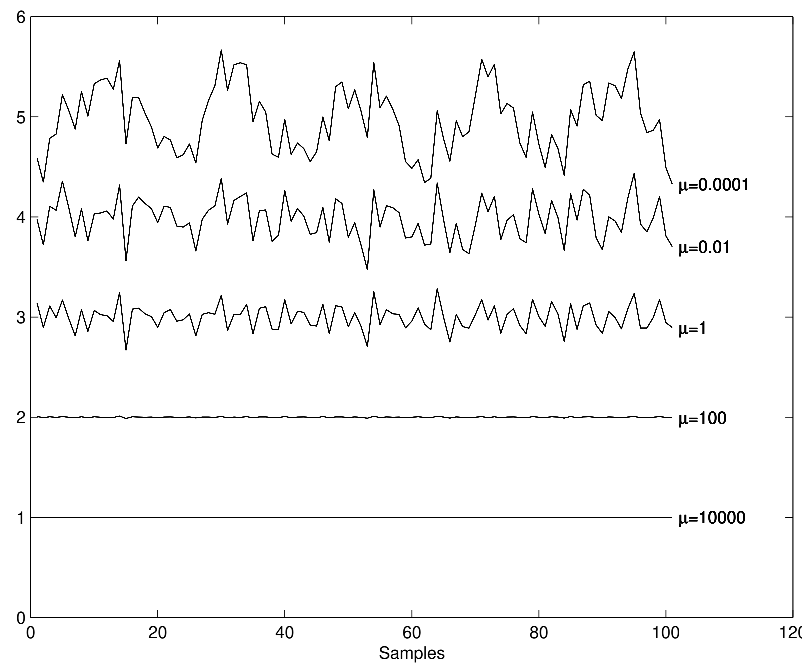


PF and Signal Leakage

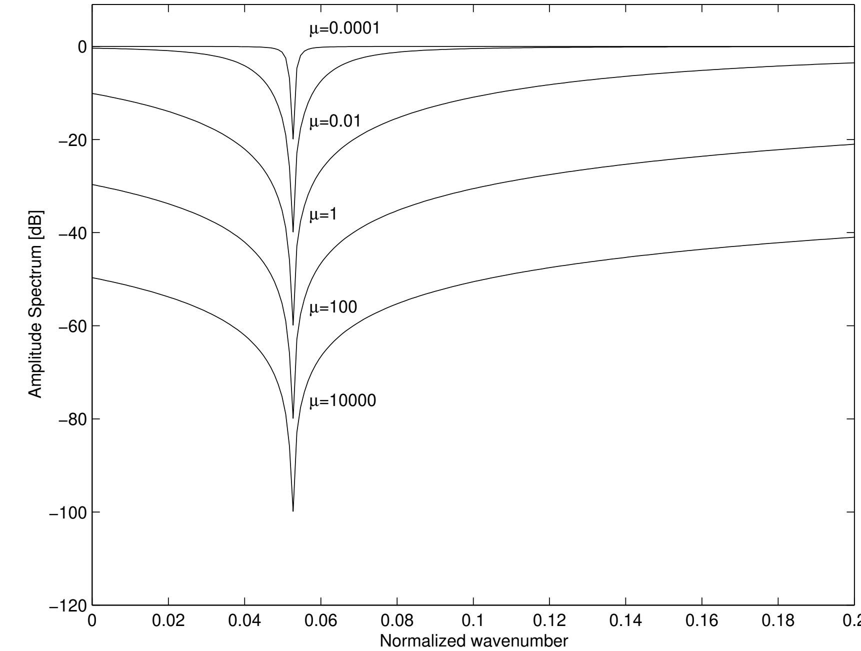
Recovered Signal



Recovered Noise

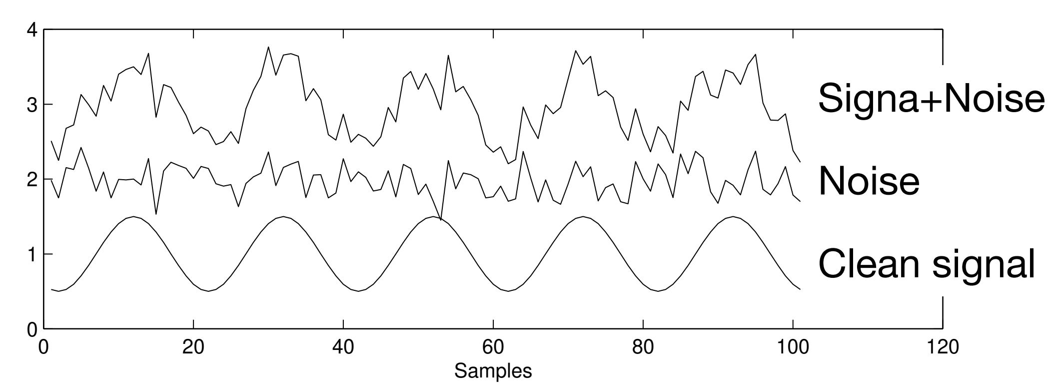


Wavenumber Filter Response



Cost function $J = \|\mathbf{s} - \mathbf{d}\|_2^2 + \mu \|\mathbf{F}\mathbf{s}\|_2^2$

Projection filter
$$\hat{\mathbf{n}} = \mathbf{P}\mathbf{d}$$
$$= \mathbf{P}(\mathbf{n} + \mathbf{s})$$



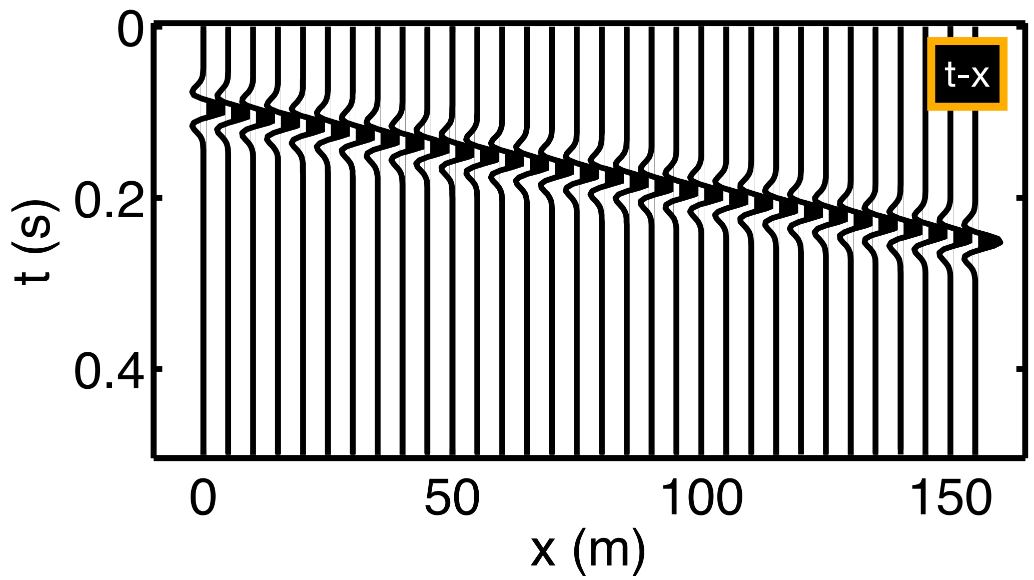
FX and FXY filters

- The previous analysis was 1D and remains 1D when applied in the FX domain. In other words, we consider signals in x for constant frequency slides. This is why theses methods are often called FX Deconvolution or FX Projection Filter.

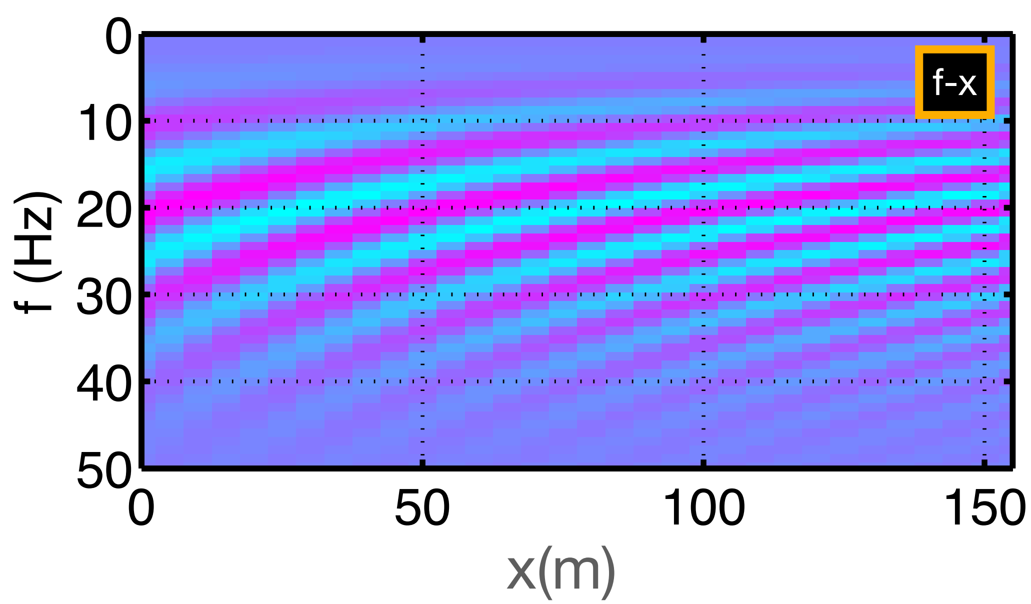
- For problems in more than one dimensions, for instance FXY filters, I prefer to adopt implicit form operators and Conjugate Gradients to compute the filters iteratively. I am sure there are more elegant ways of solving the FXY filter.

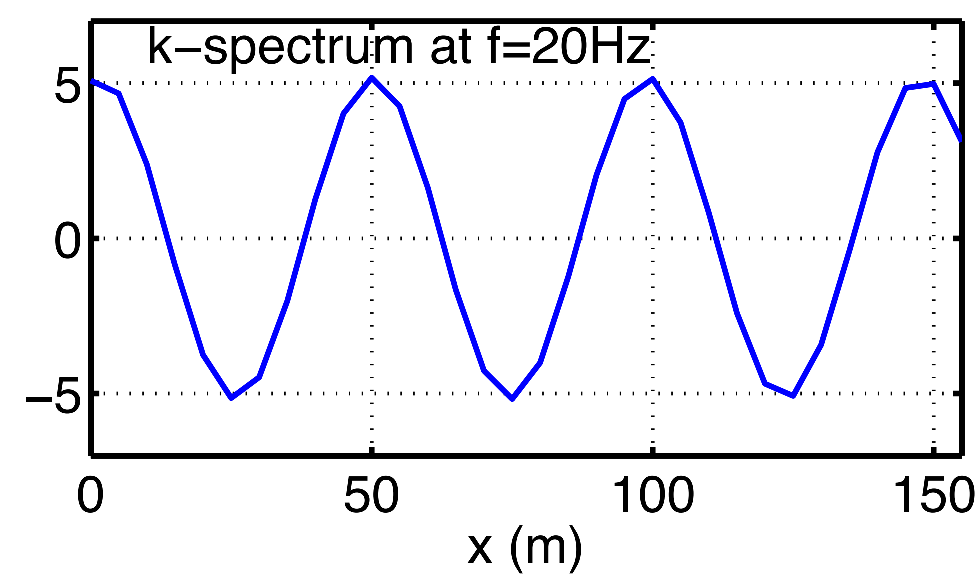
FX model

$$d(t,x) = a(t - px) \rightarrow D(\omega, x) = A(\omega)e^{-i\omega px}$$

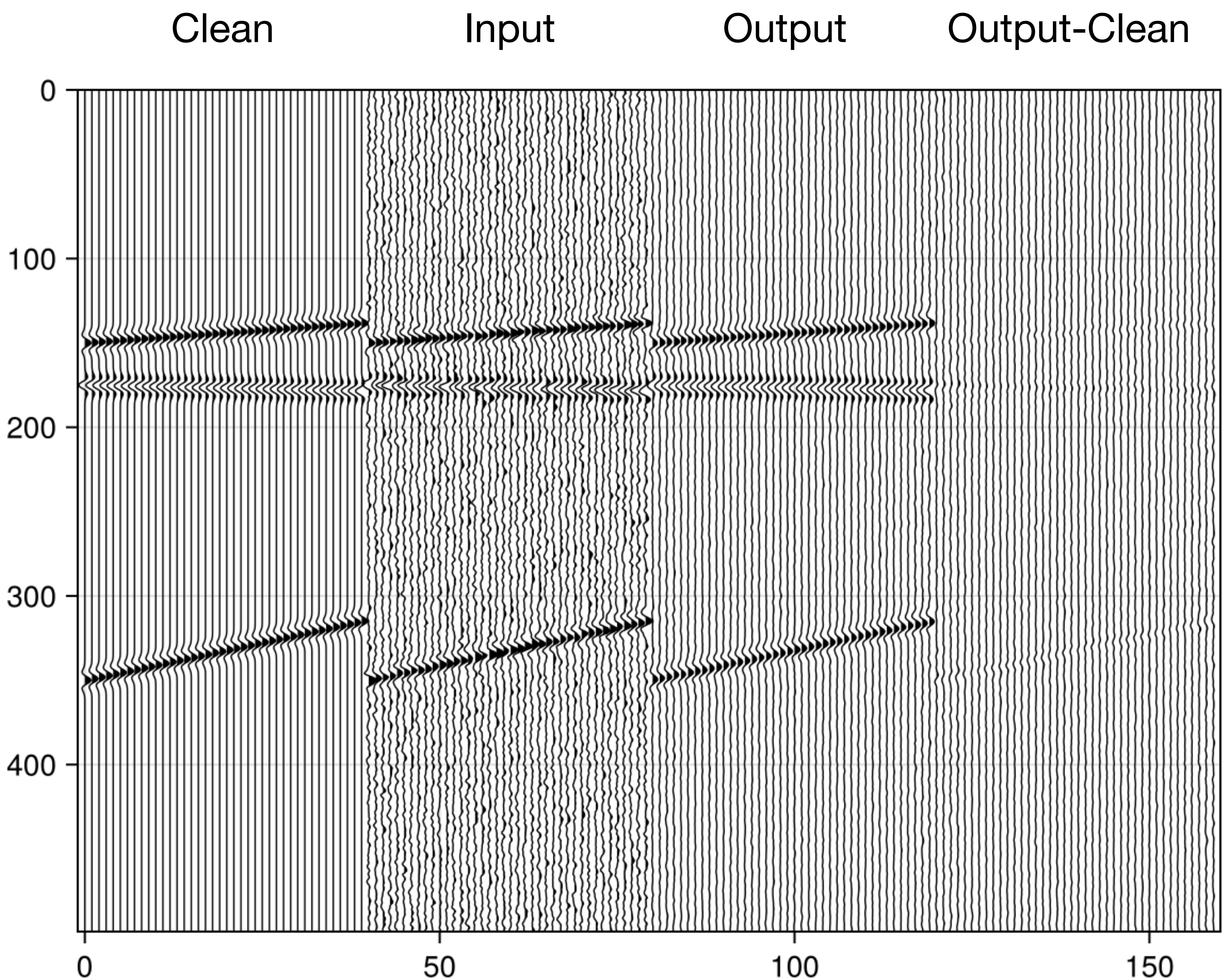


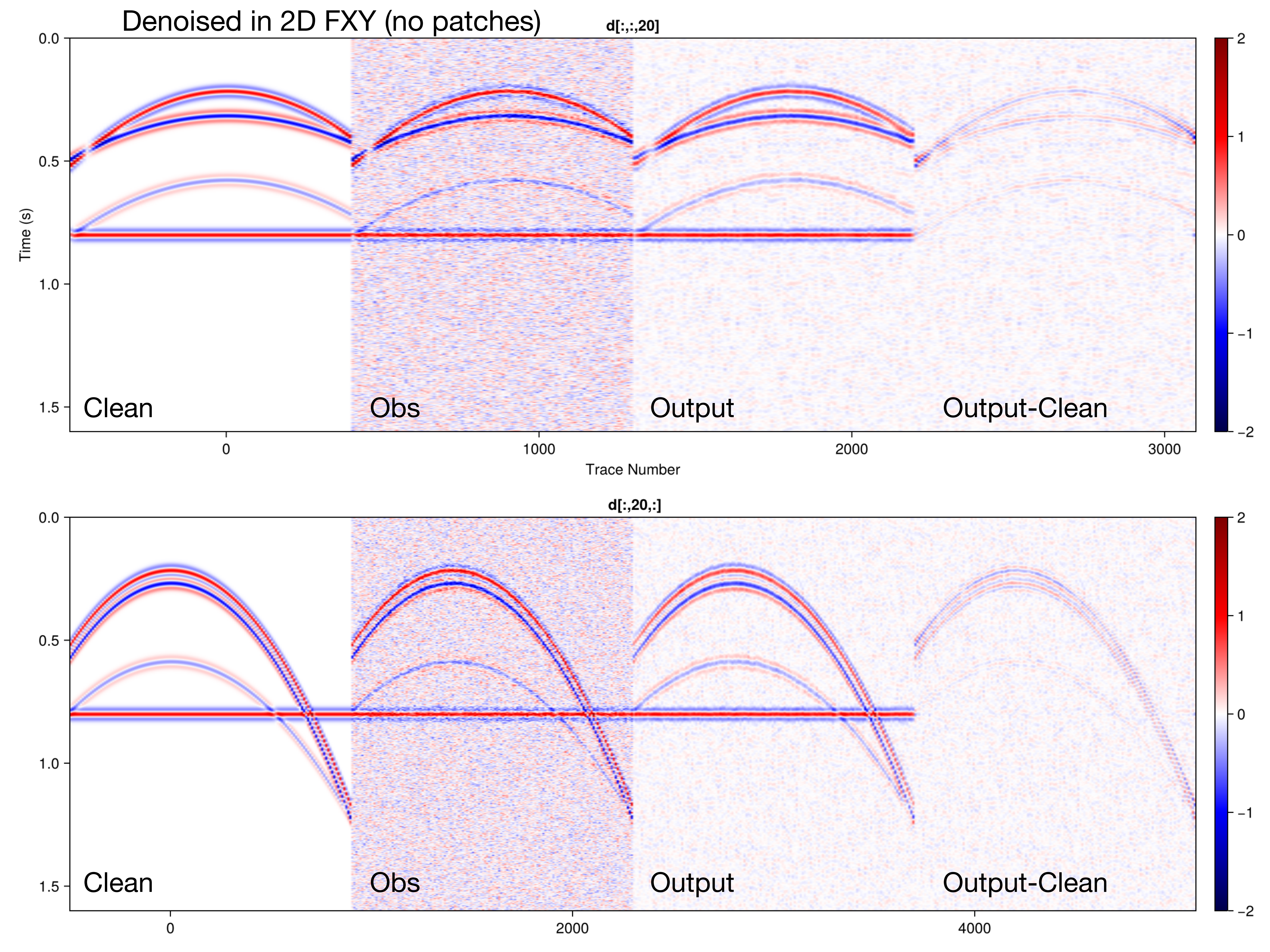
A dip in t-x is a complex exponential in f-x

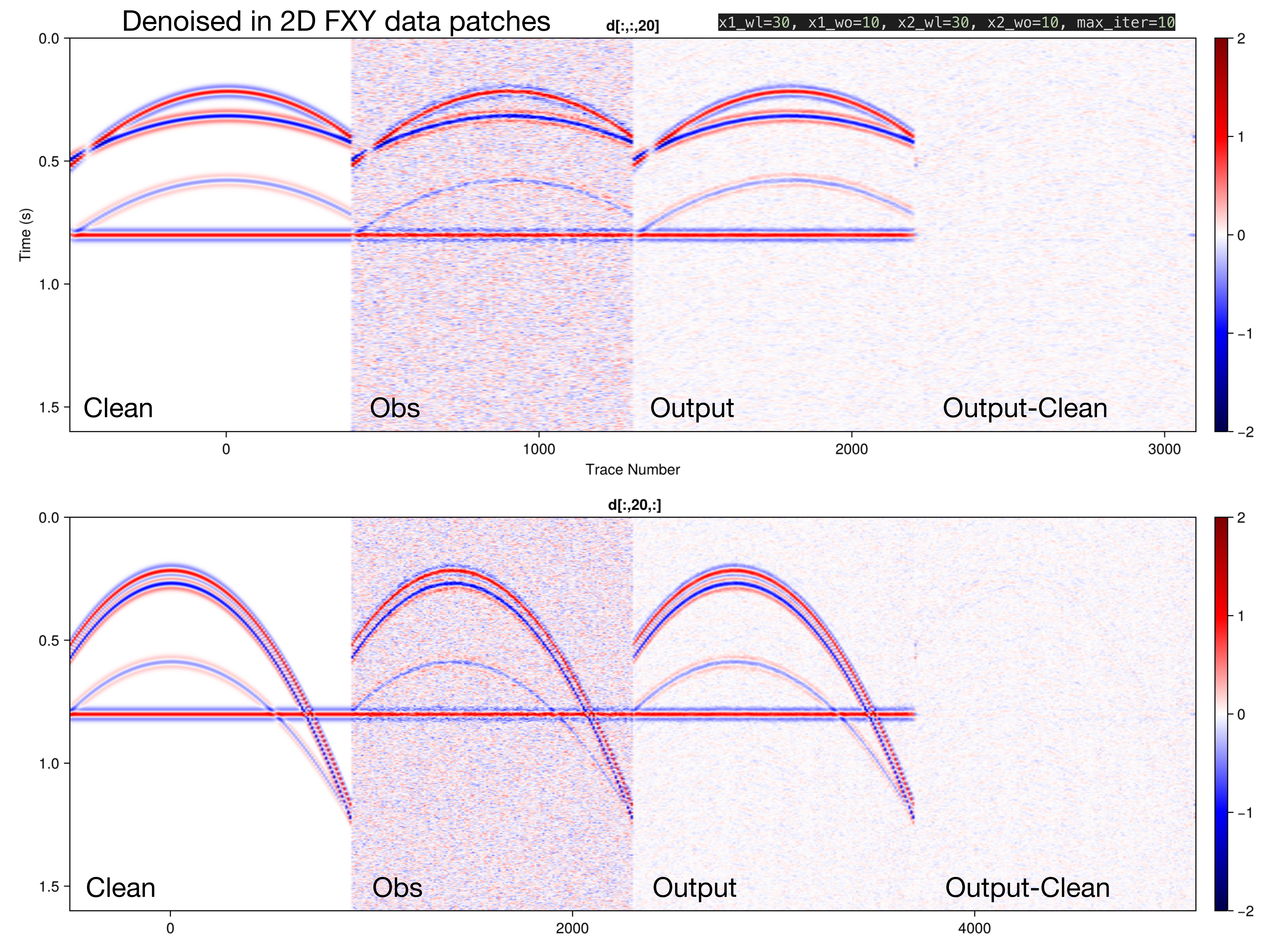




Demo_FX_decon.ipynb

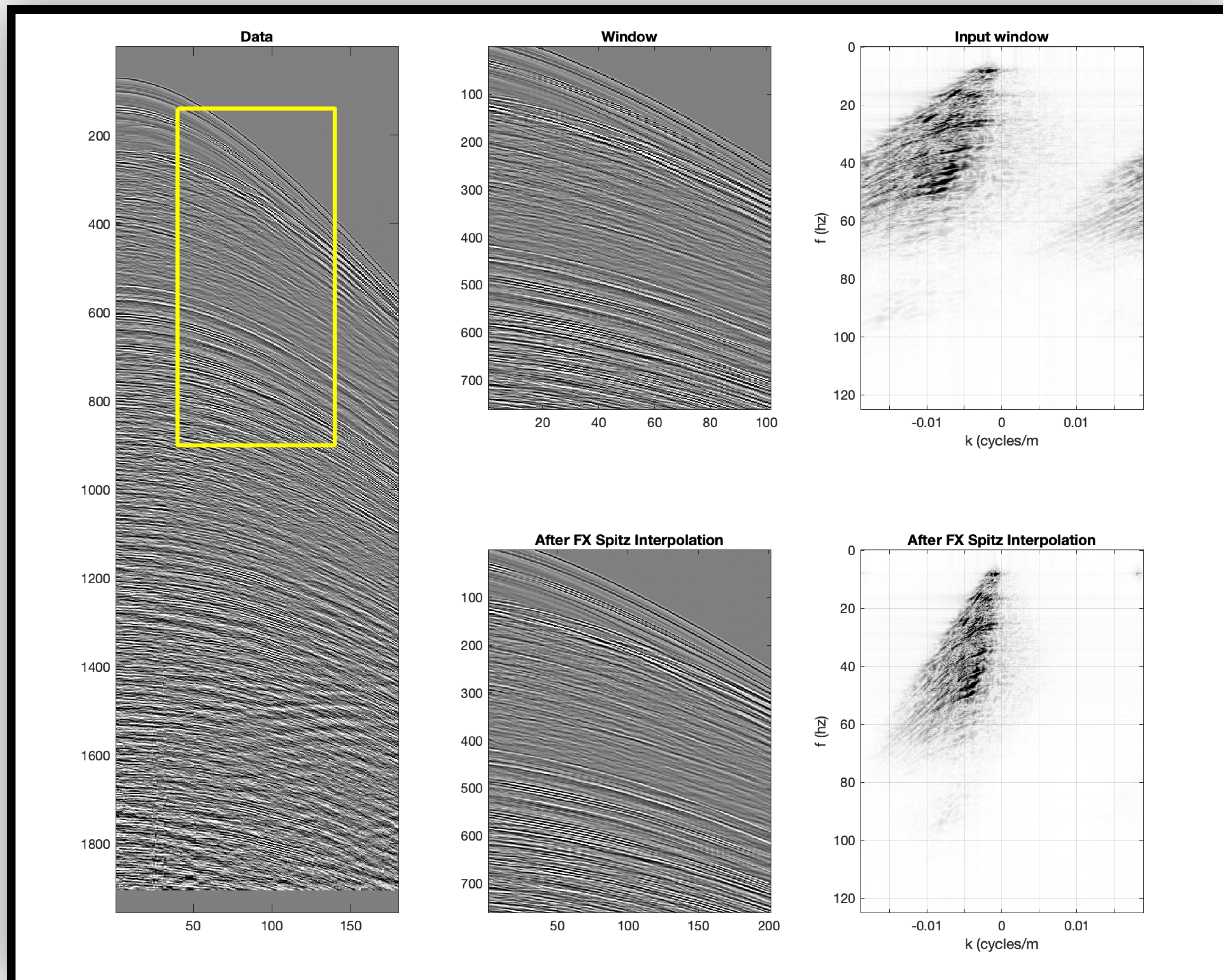



Denoised in 2D FXY (no patches)
d[:,:,20]
d[:,20,:]
Time (s)
Trace Number
Clean
Obs
Output
Output-Clean
0.0
0.5
1.0
1.5
0
1000
2000
3000
4000
2
1
0
−1
−2



Denoised in 2D FXY data patches
d[:,:,20]
x1_wl=30, x1_wo=10, x2_wl=30, x2_wo=10, max_iter=10
Time (s)
Trace Number
Clean
Obs
Output
Output-Clean
d[:,20,:]
Clean
Obs
Output
Output-Clean

FX Interpolation



Spitz, S., 1991, Seismic trace interpolation in the F-X domain: Geophysics, 56, 785-794.

- /Users/msacchi/Dropbox/SeisDenoise.jl/New/FX_Example.ipynb