The basic mathematical expression of a simple linear filter is

\[ y_t = x_t * f_t \]

where

- \( y_t \) = output of filter
- \( x_t \) = input to filter
- \( f_t \) = filter impulse response function

and \( * \) is the mathematical operation of *convolution*.

A discrete formulation of the convolution operation is:

\[ y_t = \sum_s x_s \cdot f_{t-s} \]

1. Compute (and plot) the output \( y_t \) that you would expect from the filter:

\( f_t = 6, 5, -1, 3, 2 \)

for the following inputs:

a) \( x_t = 1, 0, 0, 0, 0 \)

b) \( x_t = 0, 0, 1, 0, 0 \)

c) \( x_t = -8, 2, 3, 6, 7 \)

2. Compute (and plot) the Fourier Transforms of \( y_t, f_t, \) and \( x_t \) from 1c above.

How does the product of the Fourier Transforms of \( x_t \) and \( f_t \) compare with the Fourier Transform of \( y_t \)?
3. The convolution operation is reversible. Deconvolution can be thought of as applying a filter \( f_t^{-1} \) that negates the effect of some original filter \( f_t \), i.e.,

\[
f_t * f_t^{-1} = 1, \text{ therefore}
\]

\[
y_t * f_t^{-1} = x_t * f_t * f_t^{-1} = x_t * 1 = x_t
\]

The computation of this inverse can be aided by the use of Fourier (transform, divide, inverse transform) or Z (polynomial division) transforms. MATLAB has a command that performs deconvolution by polynomial division.

Compute (and plot) the first 4 coefficients of the spiking deconvolution filter corresponding to \( f_t = 36, -2, 18, -2 \).

What is the result of convolving this deconvolution filter with \( f_t \)?

Plot the result.

Compute (and plot) the first 10 coefficients of the spiking deconvolution filter corresponding to \( f_t \). How does this result compare with the output from the 4 term decon filter?

Compute the least squares error of the two resulting deconvolution outputs above assuming that the desired output is the perfect “spike”, i.e. 1,0,0,0,.....

How does this measure of goodness of fit compare between the two filters?

What does this suggest as an approach to picking a spiking deconvolution filter?