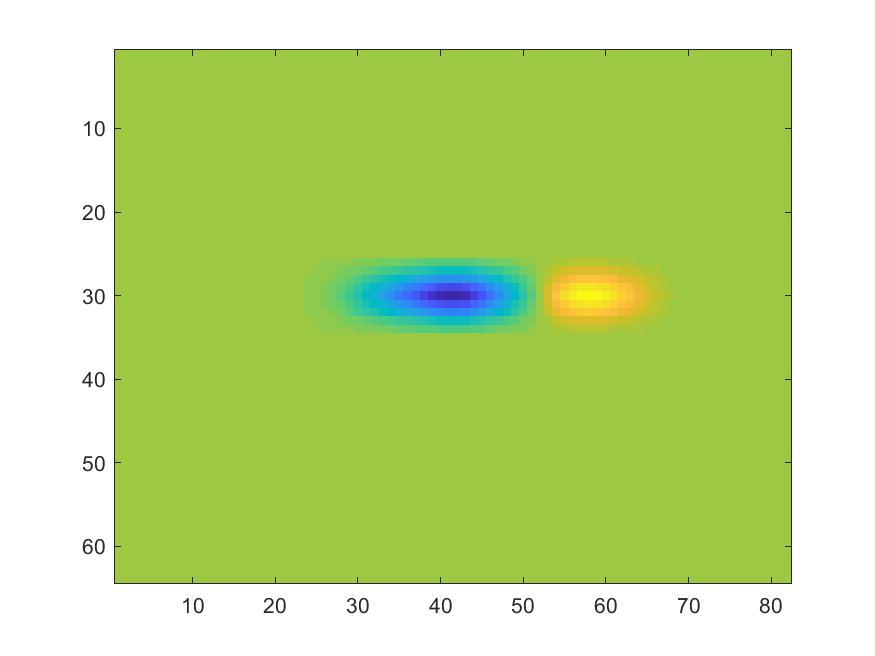
**Template Matching Algorithms**

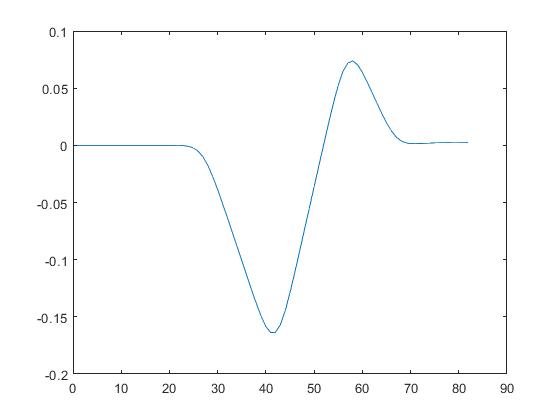
The templates:

Templates produced by the kilosort algorithm are 82 x 64 images. The number of templates depends on the number of clusters, which is set manually and is recommended to be 2-4 times more than the number of channels. The y axis tells you which tetrode the data is from



Not sure how the third dimension was obtained if I’m honest… Are they made from data from different tetrodes stitched together? All templates seem symmetrical…

Cross-sectional view taken at y-coordinate 30:



AIMS: Store a library of templates that can be used to efficiently match spikes.

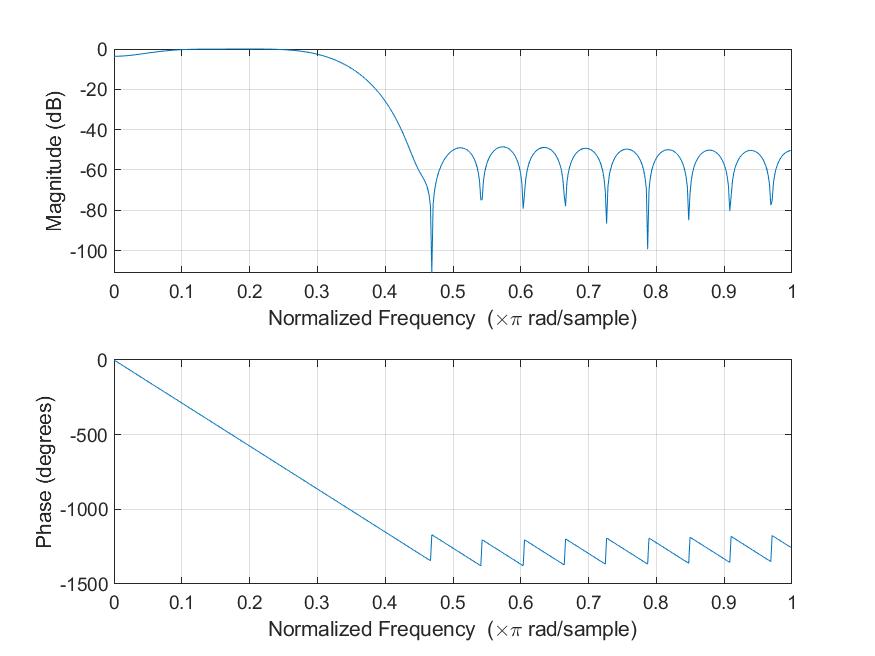
Pauline said to only compare spikes that had previously been detected on the same tetrode. The tetrode channels are in multiple of 4, so first 4 channels come from first tetrode.

POSSIBLE APPROACHES:

1. Extract Principle Components from template (spikes are essentially 3D, but 2d matters most?).
2. Mean squared error on templates from the same tetrode?
3. Detect Spike. Amplitude thresholding? For now, set a manual threshold. Later, set automatic threshold to be a multiple of the standard deviation of the noise: threshold = k \* or something better.
4. Extract spike features. How? PCA, SVD

Implementation

1. Filter data. Data is sampled at 30kHz. We need to bandpass between 300hz and 5kHz

Using MATLAB function a = fir1(32, [0.02 0.33]), I designed an order 32 fir filter that has the following frequency spectrum.

1. Detect spike by comparing to a constant
2. : isoutlier func, matlab.