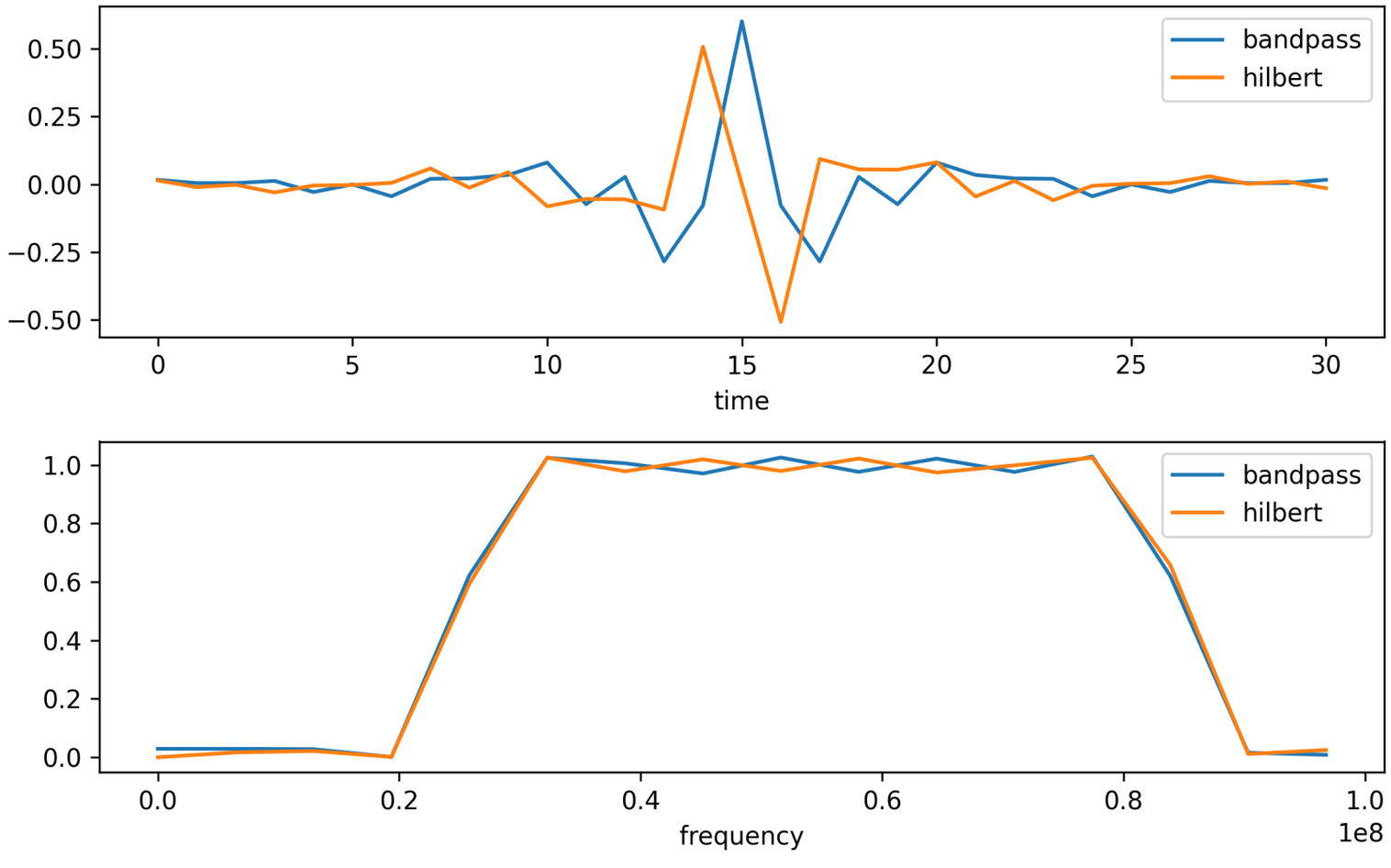
TBB Pulse finding

Finite impulse response filter

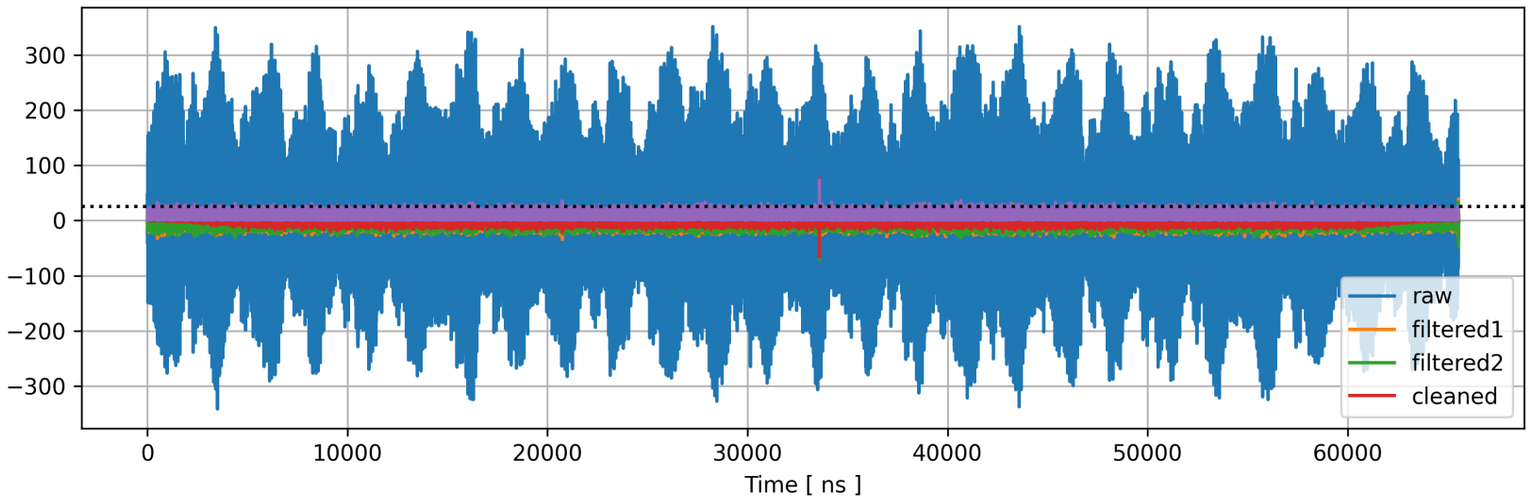
`scipy.signal.remez(numtaps, bands, desired, weight=None, Hz=None, type='bandpass', maxiter=25, grid_density=16, fs=None)` [source]

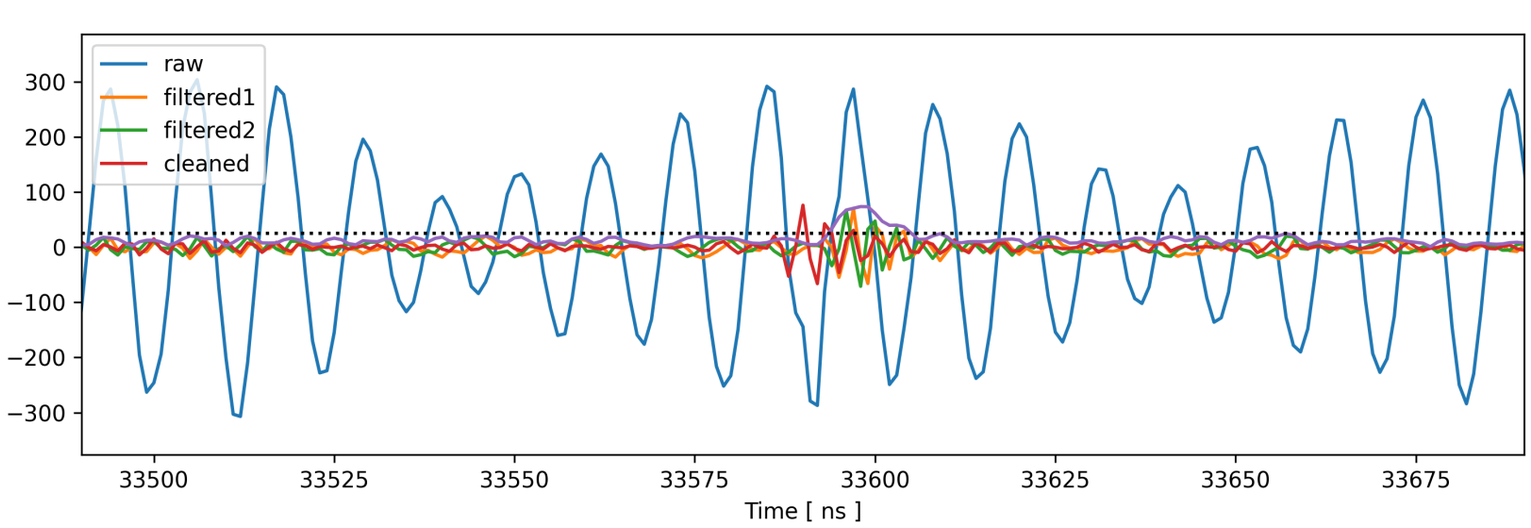


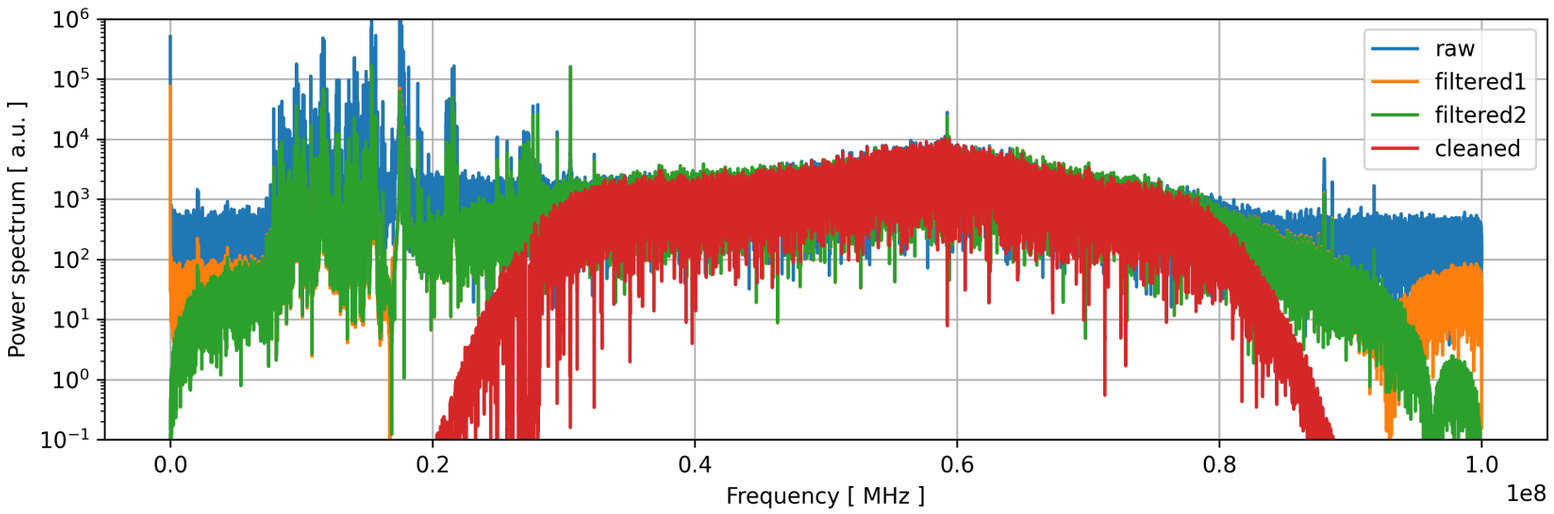
$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$
$$= \sum_{i=0}^{N} b_i \cdot x[n-i],$$

Bandpass filter with 31 taps

TBB Pulse finding

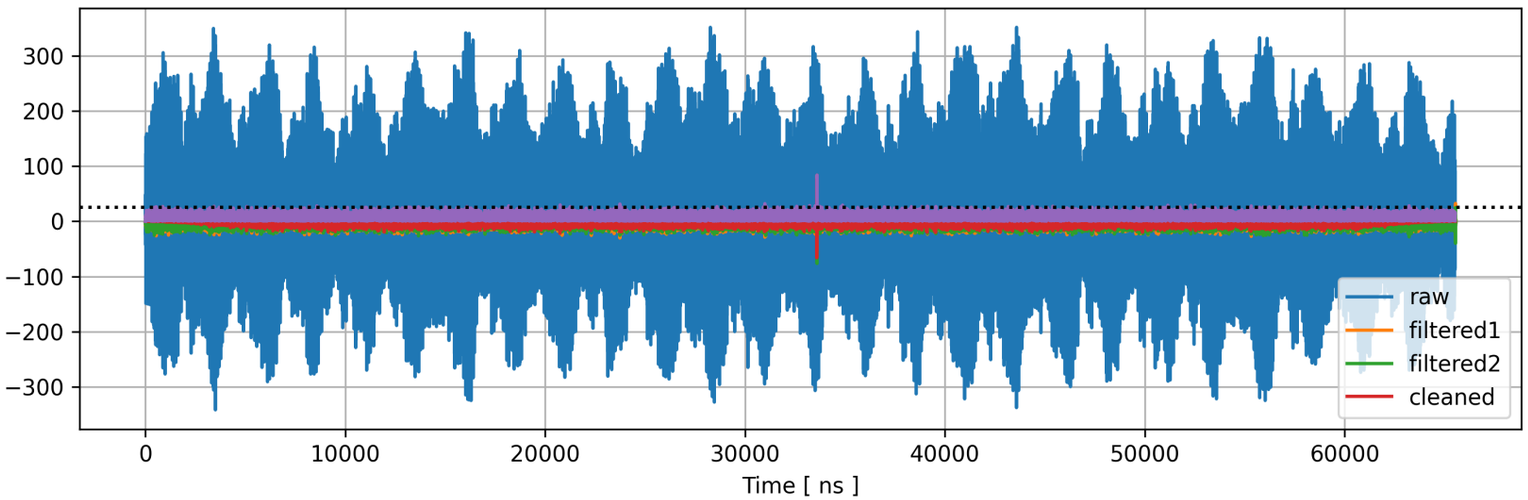


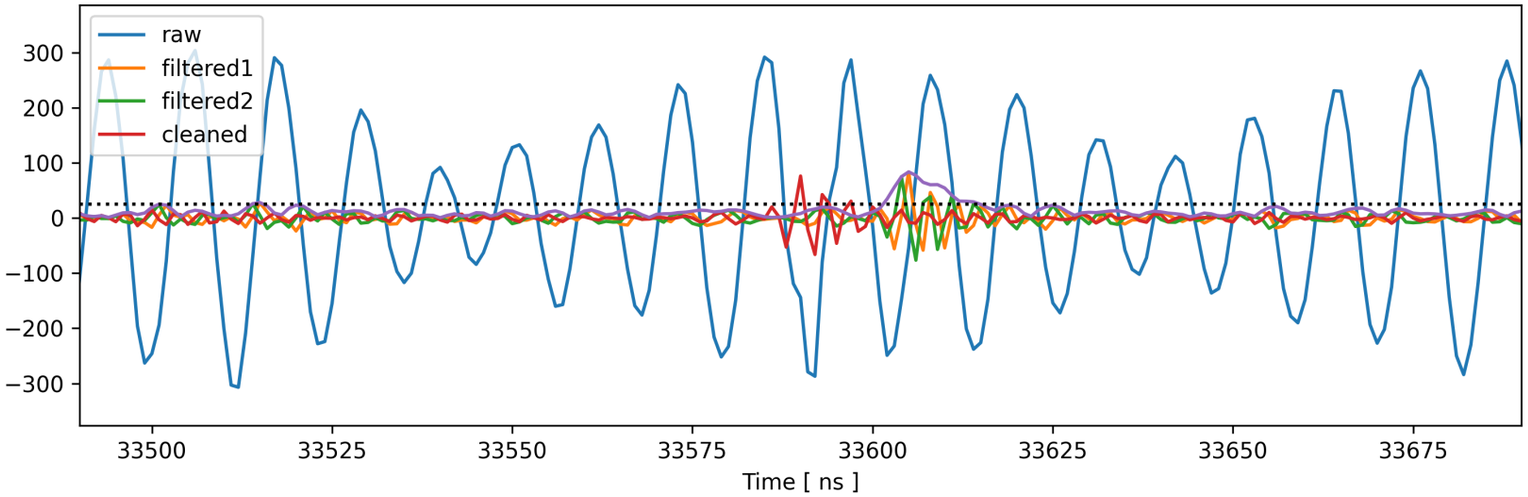


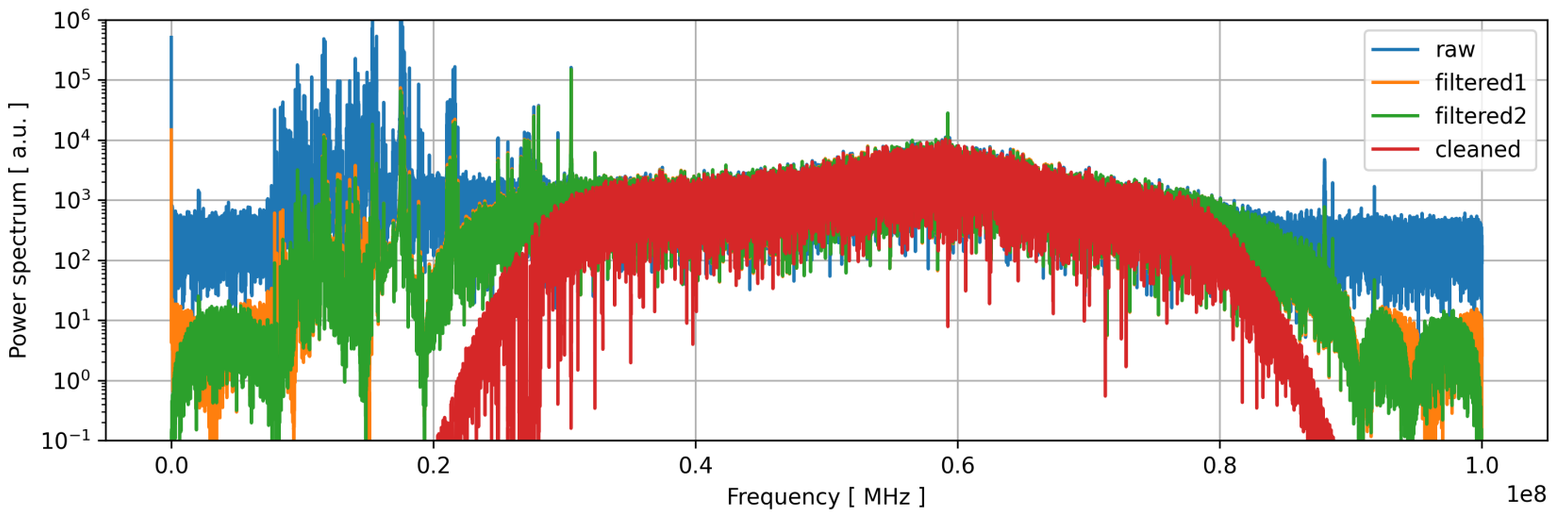


Bandpass filter with 32 taps

TBB Pulse finding

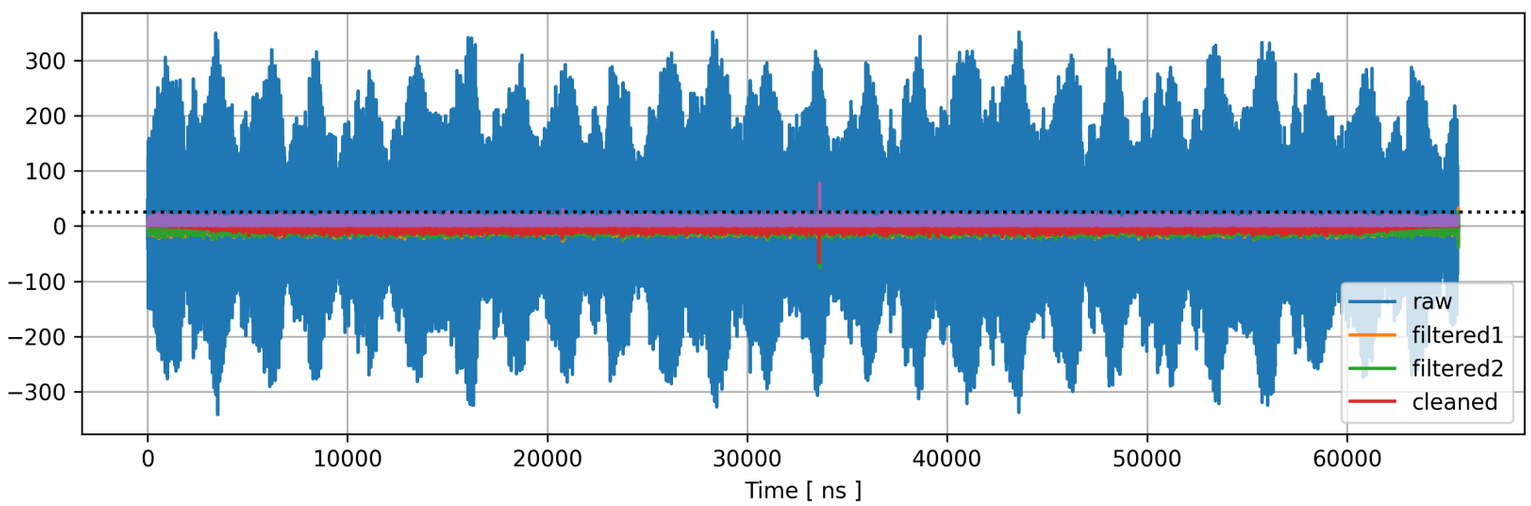


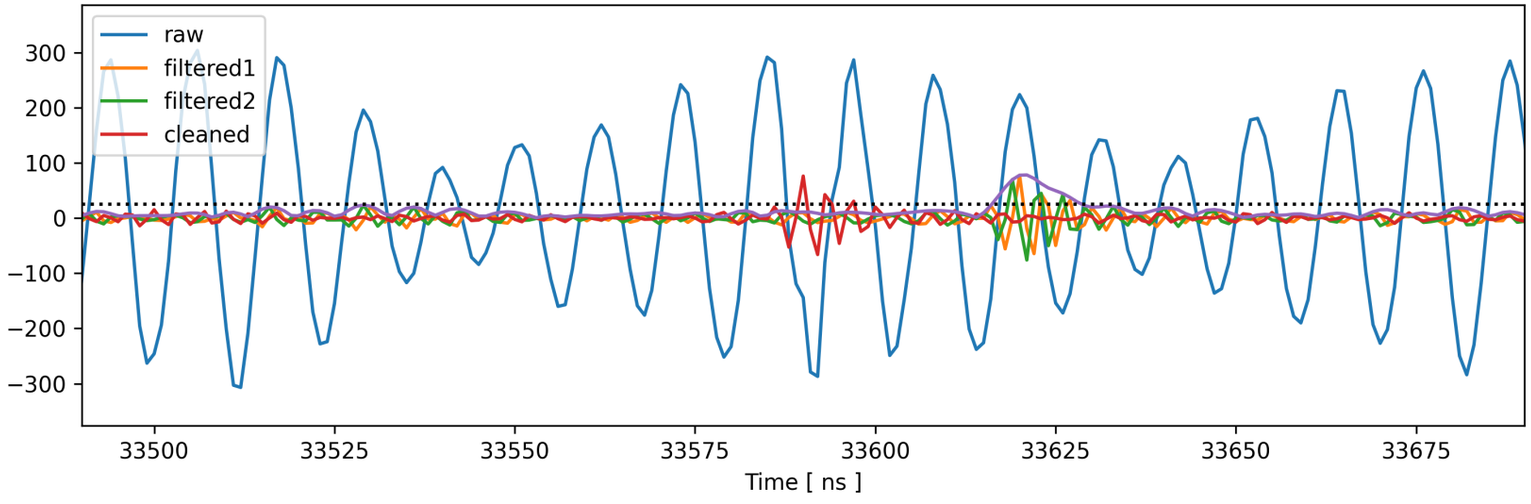


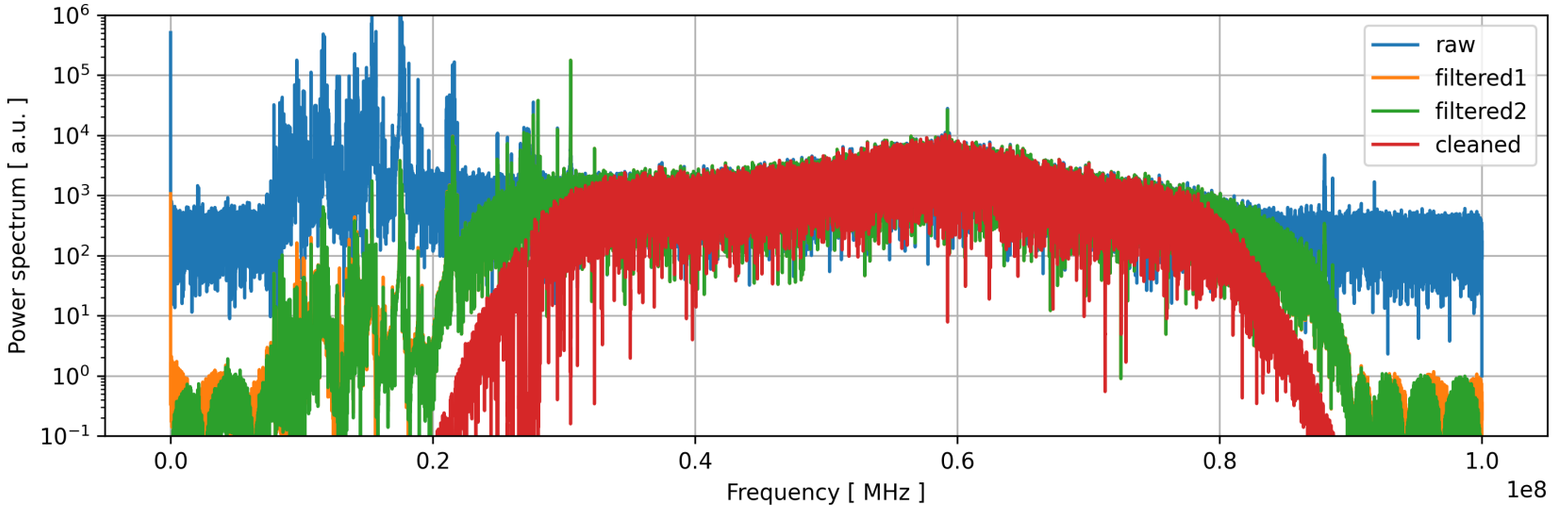


Bandpass filter with 16 taps

TBB Pulse finding

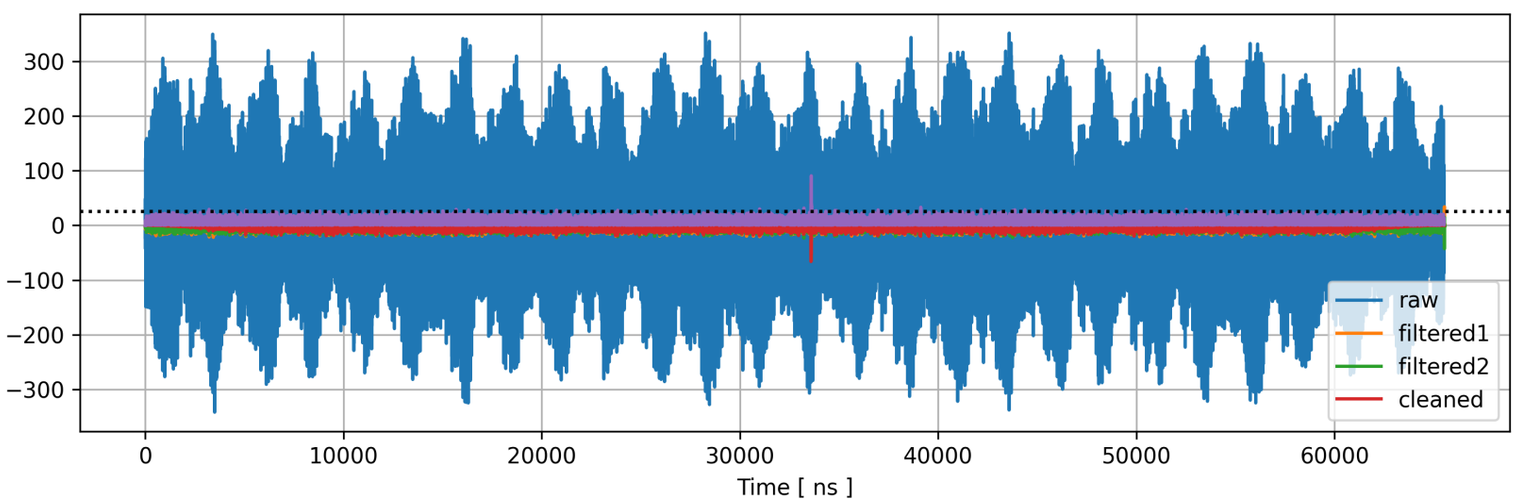


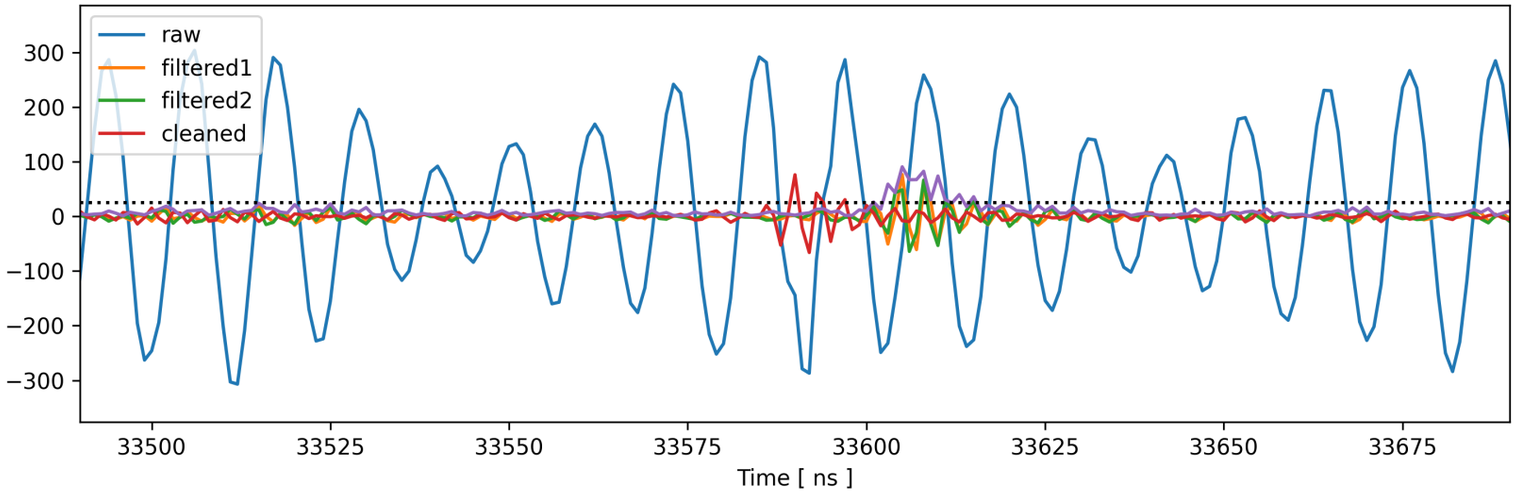


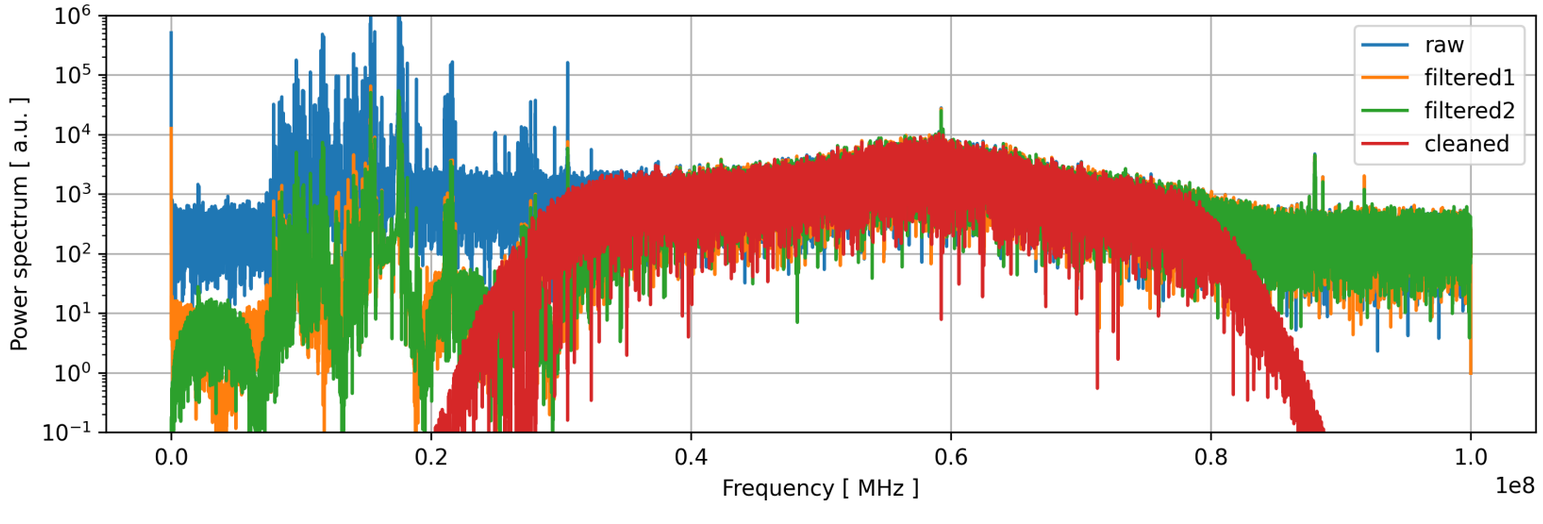


Bandpass filter with 60 taps

TBB Pulse finding

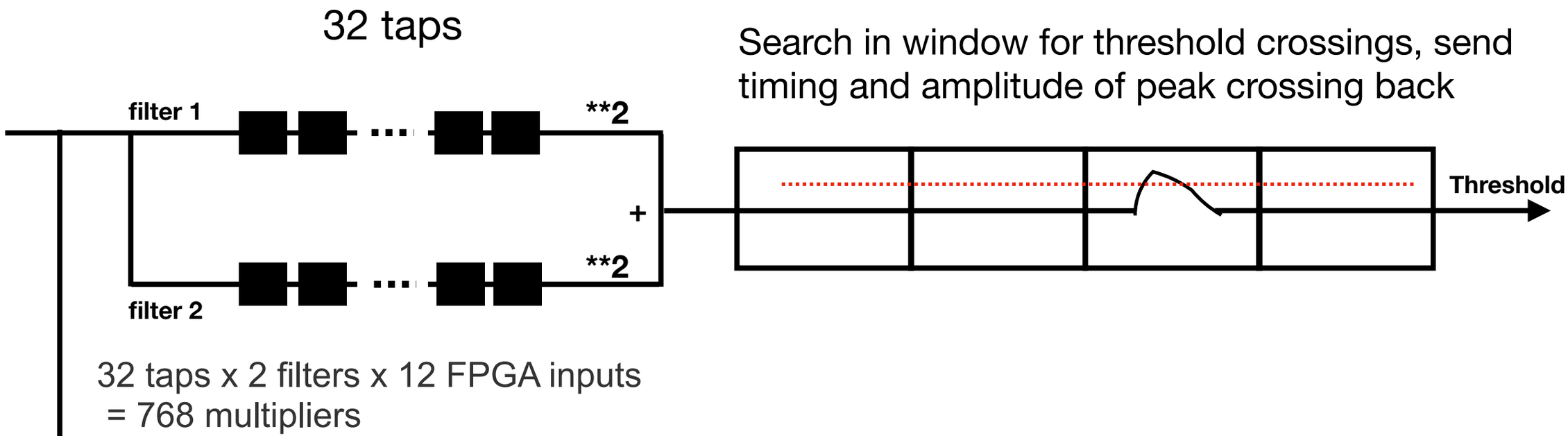






Highness filter with 32 taps

TBB Pulse finding



- Fixed filter taps?
- Do we need to change threshold (requiring a control point)?
- How to send information back (many windows at once?)
- How many antennas to monitor?

For CRs, this can be simple. We need to optimize with LIFT