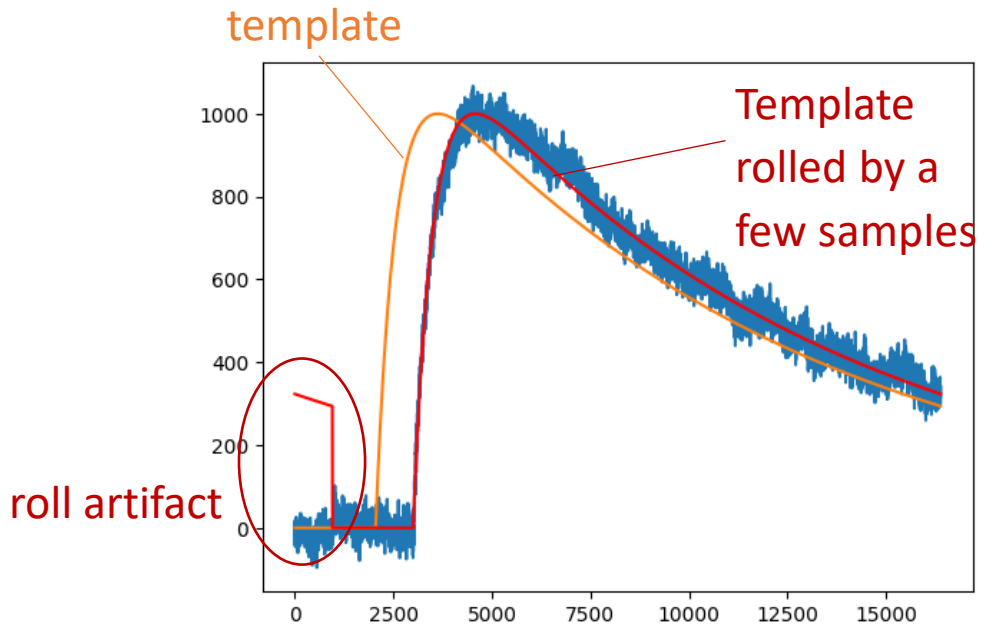
Optimum Filter: rolling vs sliding

- QP template can have a very long tail
- But you don't need the tail to do a fit. Including the entire tail would make the fit window too large, which would make the fit too computationally expensive
- Problem: currently, when OF does the fit, it does not slide the template over the trace. Instead, it performs a roll operation. If the template doesn't go to 0 at the edge of the template, the rolling operation creates an artificial step in the beginning

signal amplitude Time shift (roll) template

$$\chi^2 = \int_{-\infty}^{\infty} \frac{|v(f) - Ae^{-i\omega t_0} s(f)|^2}{J(f)} df$$

Noise PSD



Optimum Filter: rolling vs sliding

- QP template can have a very long tail
- But you don't need the tail to do a fit. Including the entire tail would make the fit window too large, which would make the fit too computationally expensive
- Problem: currently, when OF does the fit, it does not slide the template over the trace. Instead, it performs a roll operation. If the template doesn't go to 0 at the edge of the template, the rolling operation creates an artificial step in the beginning
- Solution: use a template longer than the trace and precalculate its Fourier transforms in a sliding window equal to the event's length.

$$\chi^2 = \int_{-\infty}^{\infty} \frac{|v(f) - Ae^{-i\omega t_0} s(f)|^2}{J(f)} df \quad \Rightarrow \quad \chi^2 = \int \frac{|v(f) - As(f|t_0)|^2}{J(f)} df$$

scipy.signal.ShortTimeFFT