Electronics and Pulse Shaping for DRIFT II

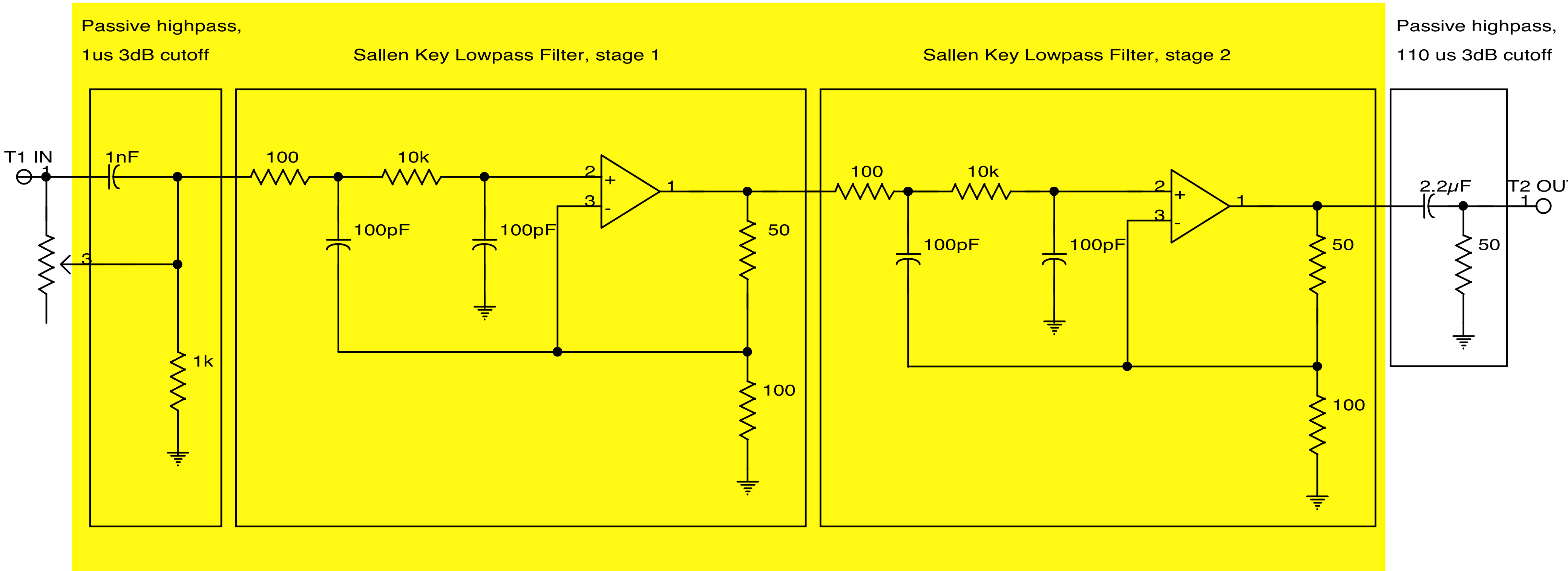
Ed Daw

CYGNUS workshop, Boulby, 24th July 2007

University of Sheffield

DRIFT II collaboration

Current Shaping Electronics



Gaussian function. The purposes of this are to filter much of the noise from the signal of interest and to provide a quickly restored baseline to allow high counting rates. The CR-200 is available in 4 different

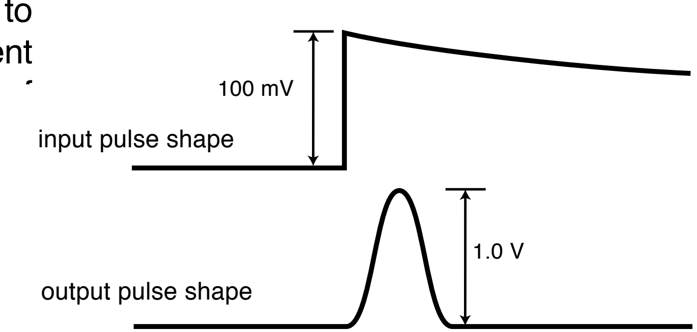
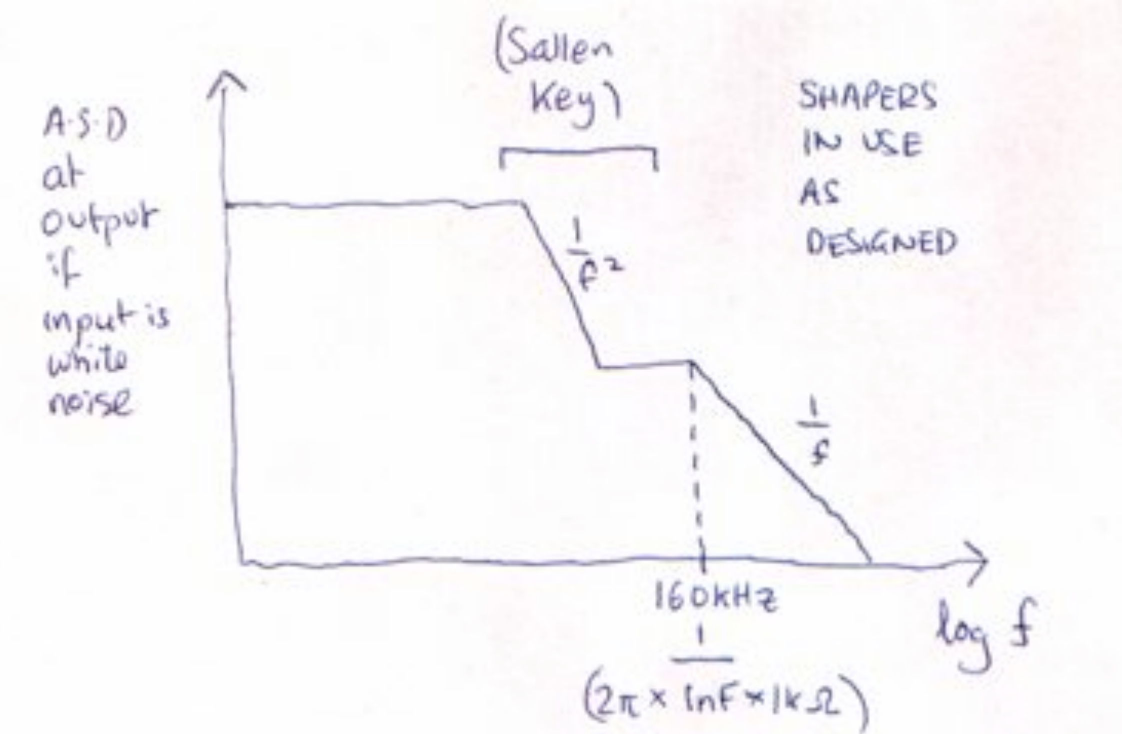
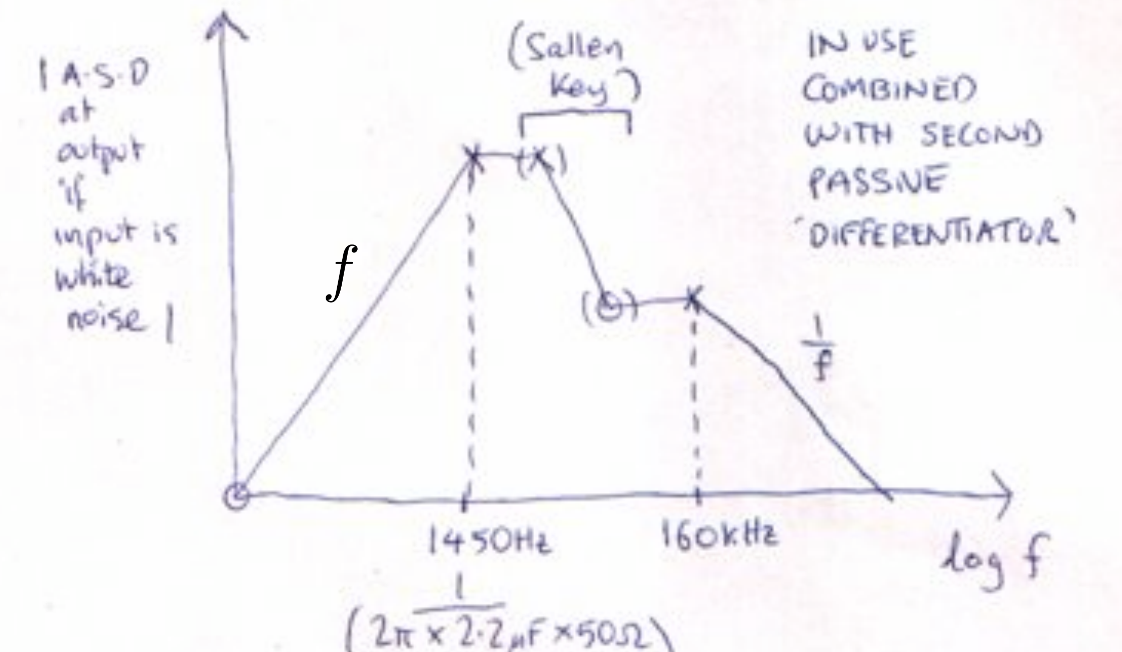
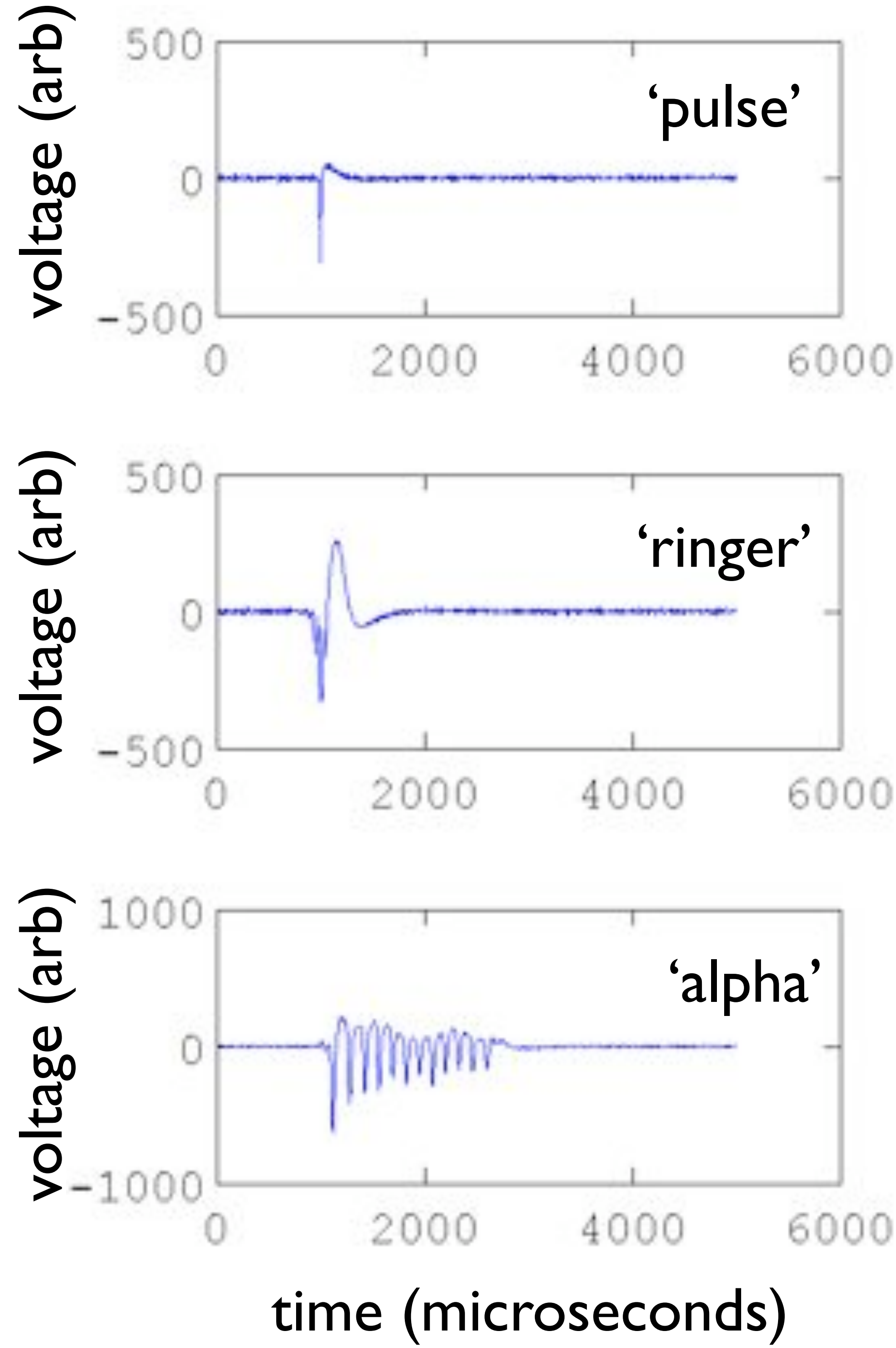


Figure 1. Comparison of sample input and output pulse shapes

Some issues with the current electronics.







Circuit Model and Inversion to remove overshoot in data already acquired with current electronics

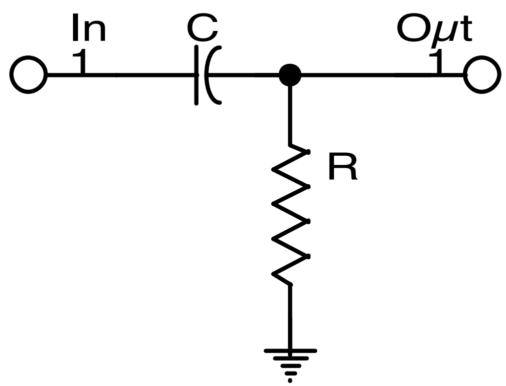
Models in terms of transfer functions of circuit elements. Three elements: (a) Input RC ‘differentiator’, (b) Sallen-Key lowpass filter (two in series) (c) Output RC ‘differentiator’

The Sallen Key stages are lowpass filters designed to remove high frequency noise and, along with it, the sharpness of the leading edge of fast-risetime signals. These filters are NON invertible.

However, we can invert the two ‘differentiator’ stages. If we do this, we should see the overshoots disappear, and the exponential decay time produced by the charge amps re-appear. We will not however recover the fast leading edge of pulses.

Digital Inverse Filter

Consider a single passive 'differentiator' circuit element



$$s = 2\pi i f$$

Impedance of resistor: R

Impedance of capacitor: $\dfrac{1}{sC}$

The transfer function H(s) is the (complex) ratio of the amplitude and phase of a sine wave at the output to that at the input, at frequency $-is/2\pi$

$$H(s) = \frac{V_O(s)}{V_I(s)} = \frac{R}{R + \frac{1}{sC}} = \frac{sCR}{1 + sCR}$$

To invert, take the reciprocal, equivalent to swapping the input and output ports.

$$H^{-1}(s) = \frac{V_I(s)}{V_O(s)} = \frac{R + \frac{1}{sC}}{R} = \frac{1 + sCR}{sCR}$$

Now take this filter and implement in software.

Filter Implementation

$$H^{-1}(s) = \frac{V_I(s)}{V_O(s)} = \frac{R + \frac{1}{sC}}{R} = \frac{1 + sCR}{sCR}$$

The general method is to take the coefficients of the polynomials in the numerator and the denominator of this equation, and perform a 'bilinear' or 'Tustin' transformation on them, yielding numbers that can be used in a numerical algorithm implementing the filter. MATLAB is the industry standard tool for doing this.

```
% construct filters
        rpos=1/polepos;
b=[rpos 1]; a=[rpos 0];
rpos2=1/p2pos;
b2=[rpos2 1]; a2=[rpos2 0];
tf1=tf(b,a);
tf2=tf(b2,a2);
tftot=tf1*tf2
tfdig=c2d(tftot,1e-6,'tustin');
[bd,ad]=tfdata(tfdig,'v');
```

Limitations of this method - (a) not ALL filters can be inverted. The resultant inverse may diverge to infinity in a controlled (slow) or uncontrolled (virtually instantaneous) way.

The above filter is unfortunately infra-red divergent. Very low frequencies cause the output to blow up for a finite input. However, maybe we can still learn something from what happens before it diverges.

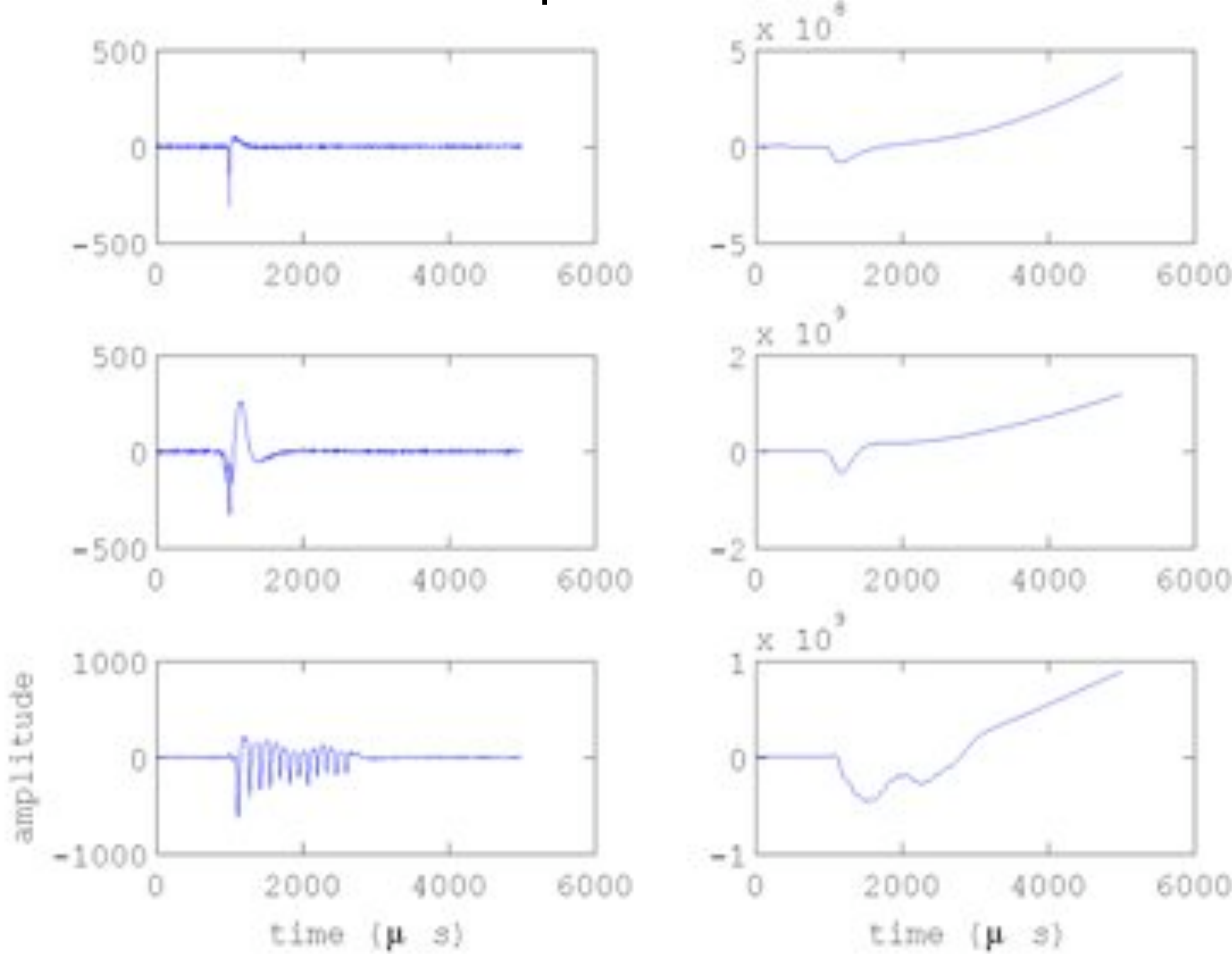
Inverse Filtering on Pulses from DRIFT II

Some DRIFT II outputs After the ‘inverse’ filter

pulse

ringer

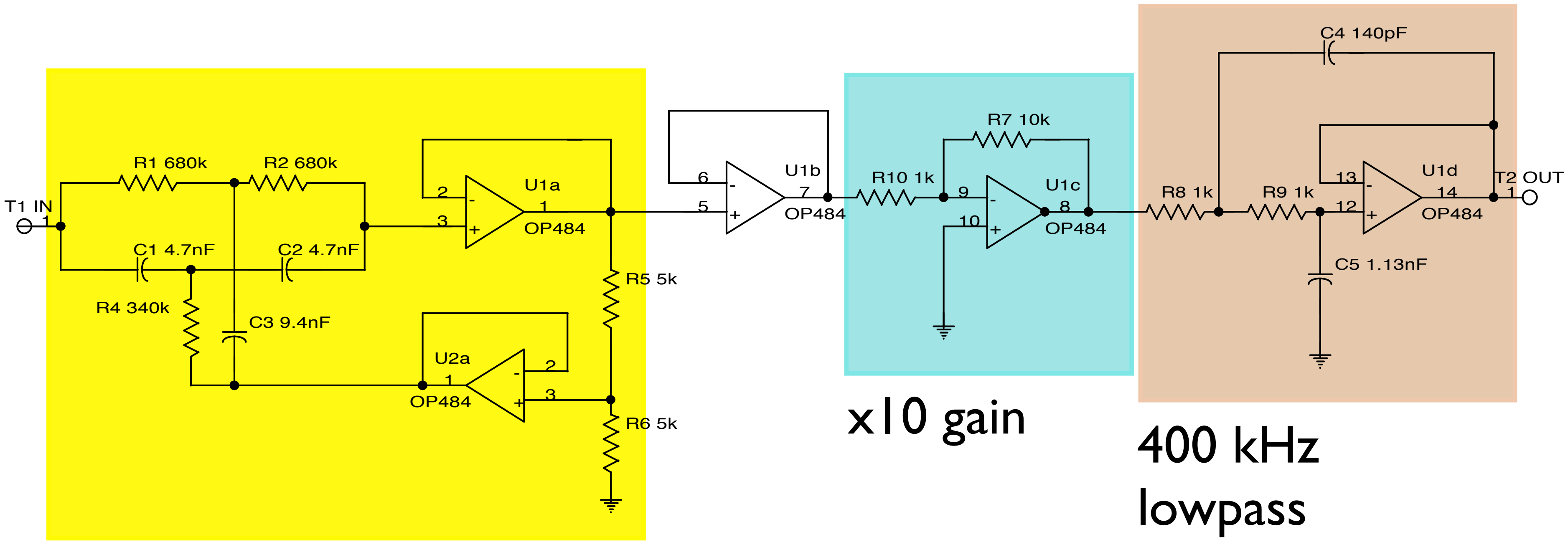
alpha



Encouraging can remove ringing as I desired, but
‘inverse’ filter unstable on long timescales due to pole at DC

Possible Replacement for the Shapers and Passive Highpass Filter

Implemented using 2 Analog Devices OP484 op-amps per channel



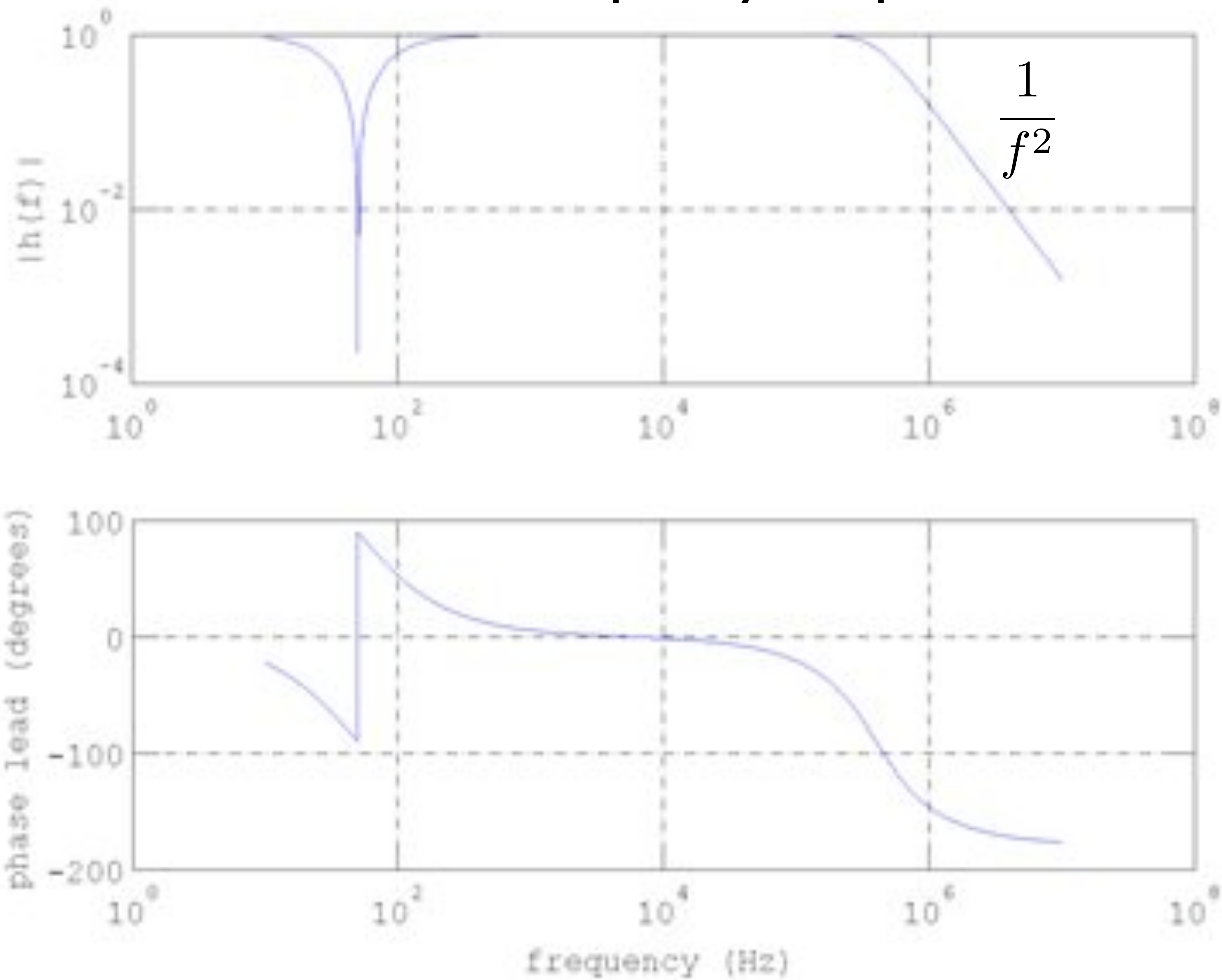
Twin T 50Hz notch filter

x10 gain

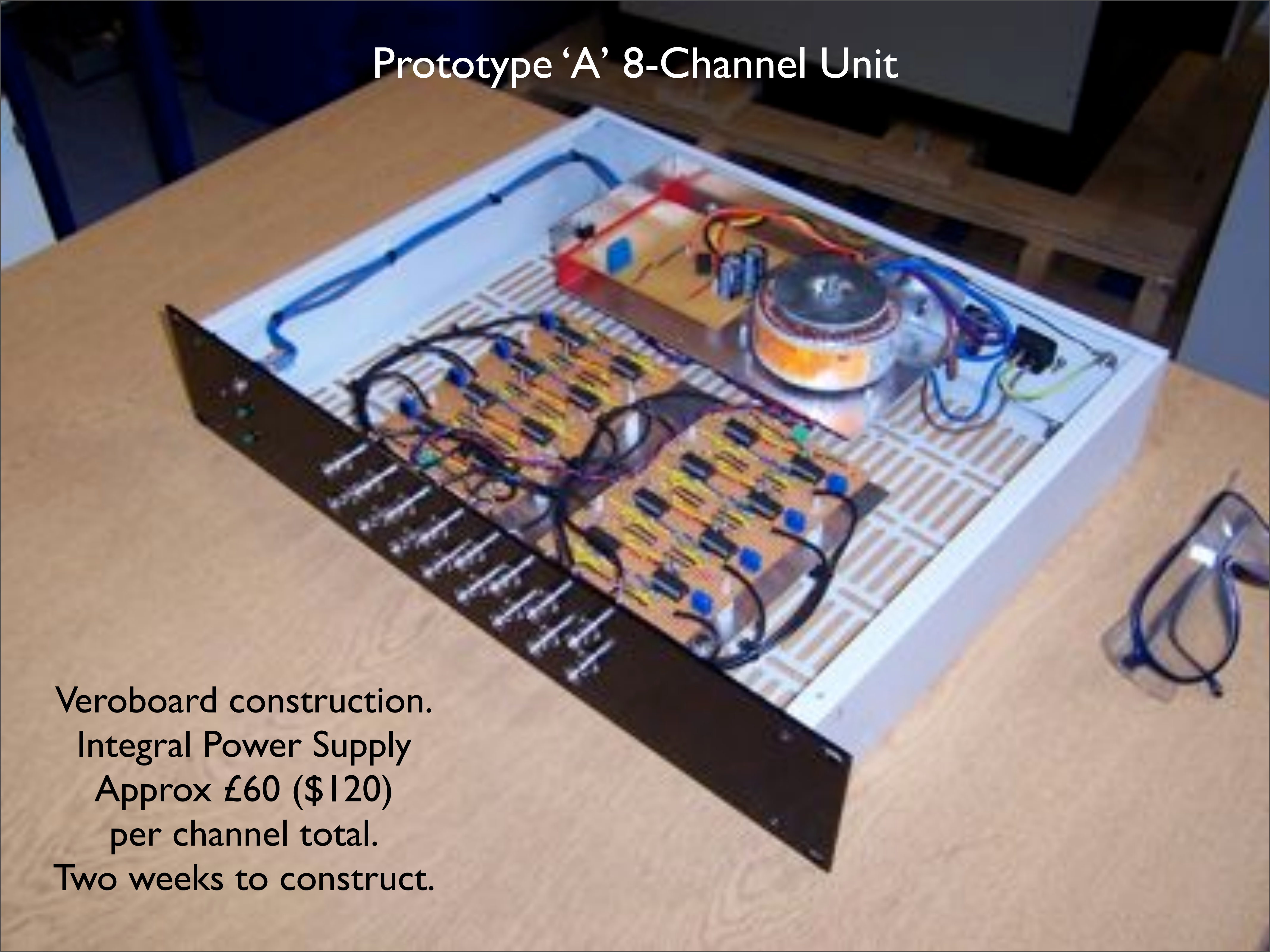
400 kHz lowpass Butterworth anti-aliasing

Designed to do as little as possible to the signal, just remove 50Hz noise , add broadband gain, and suppress high frequency noise that might alias into the Nyquist bandwidth of DC - 500kHz

Predicted Frequency Response



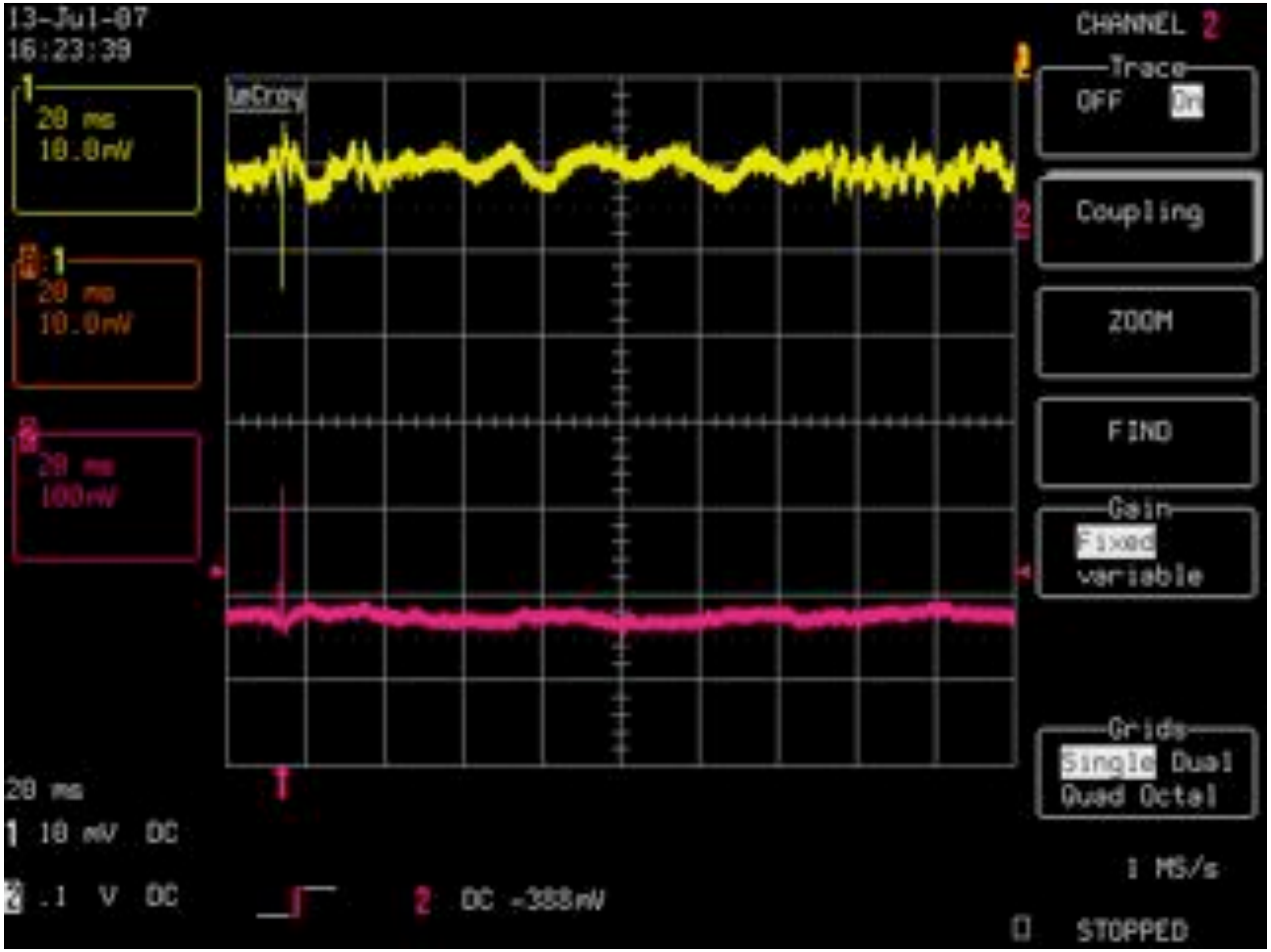
NOTE that the phase delay between about 1kHz and about 100kHz is less than about 20 degrees and reasonably flat. Therefore this electronics should distort pulses minimally, apart from removing 50Hz and f>400kHz.


Prototype ‘A’ 8-Channel Unit
Veroboard construction.
Integral Power Supply
Approx £60 ($120)
per channel total.
Two weeks to construct.

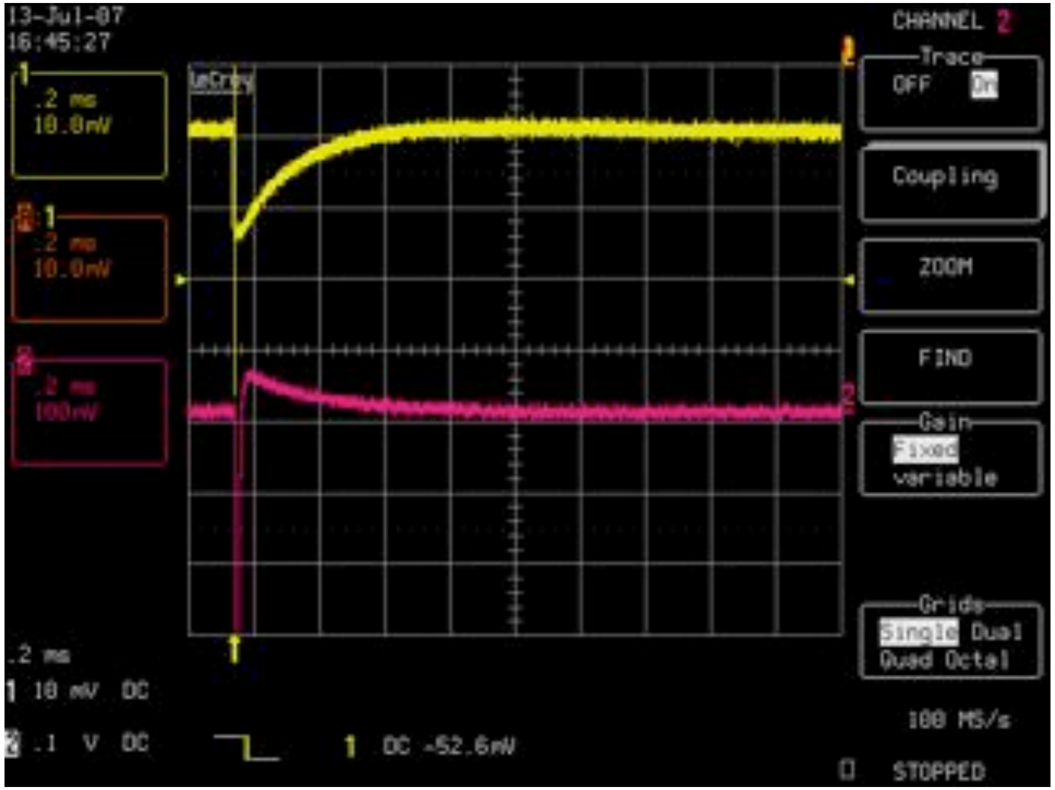
First Underground Test

Hardware installed at Boulby on 12th - 13th July.
Calibration data and 1.5 hour run with ^{252}Cf neutron source.
(Thanks Sean, Johanna)

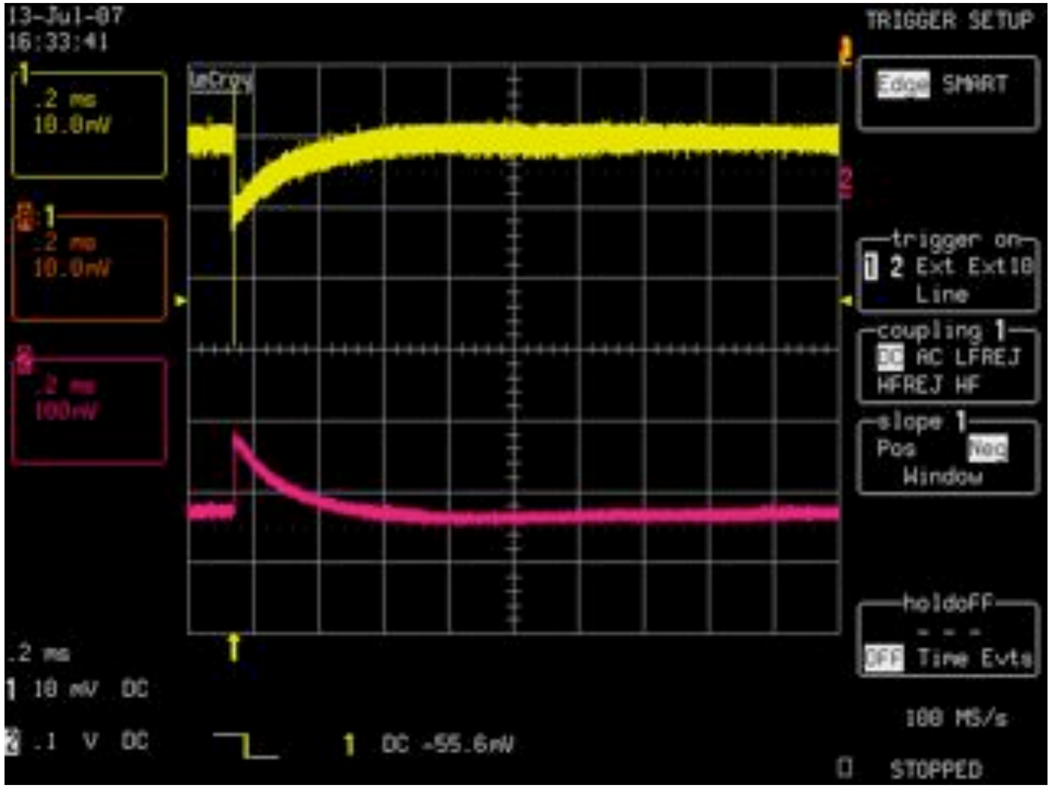
noise mitigation



Spark Response

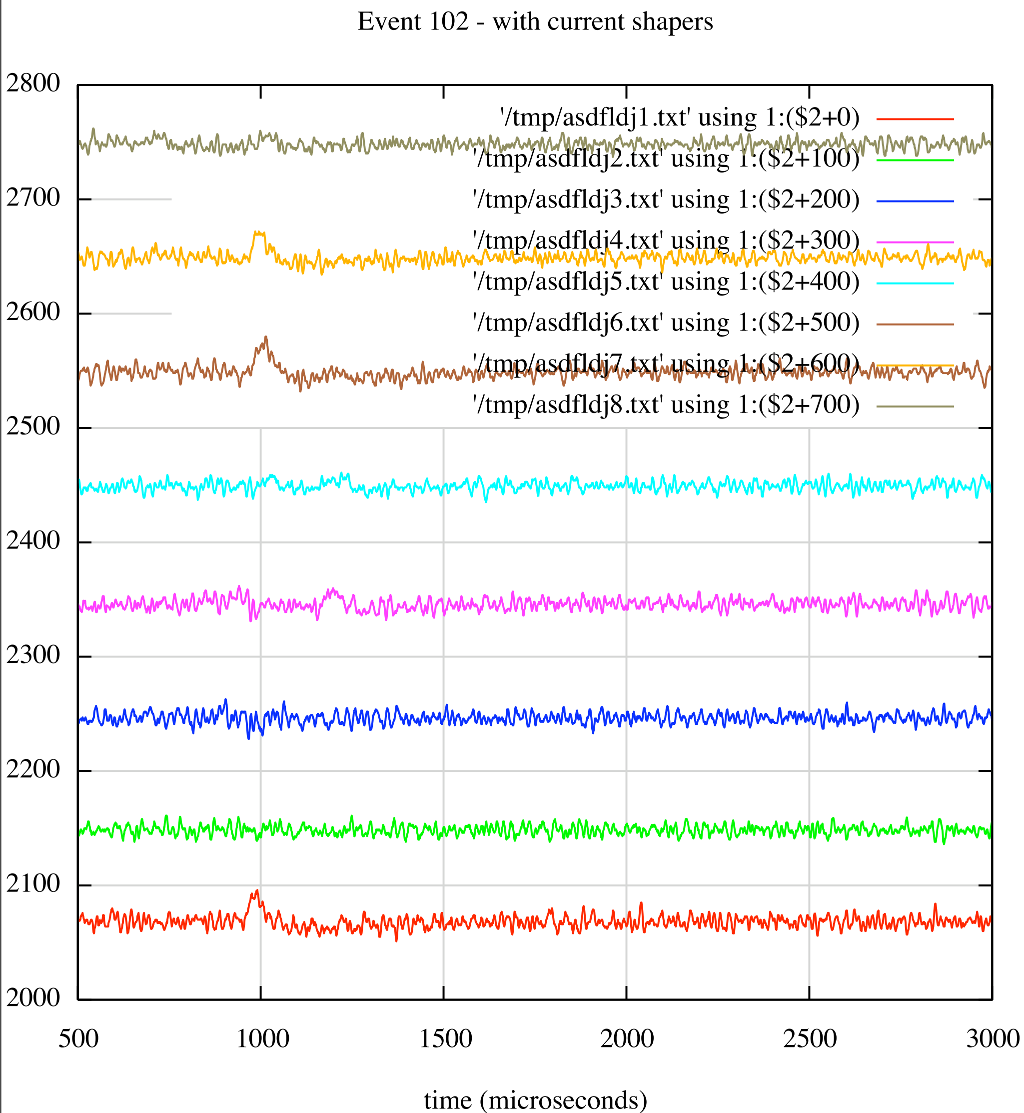


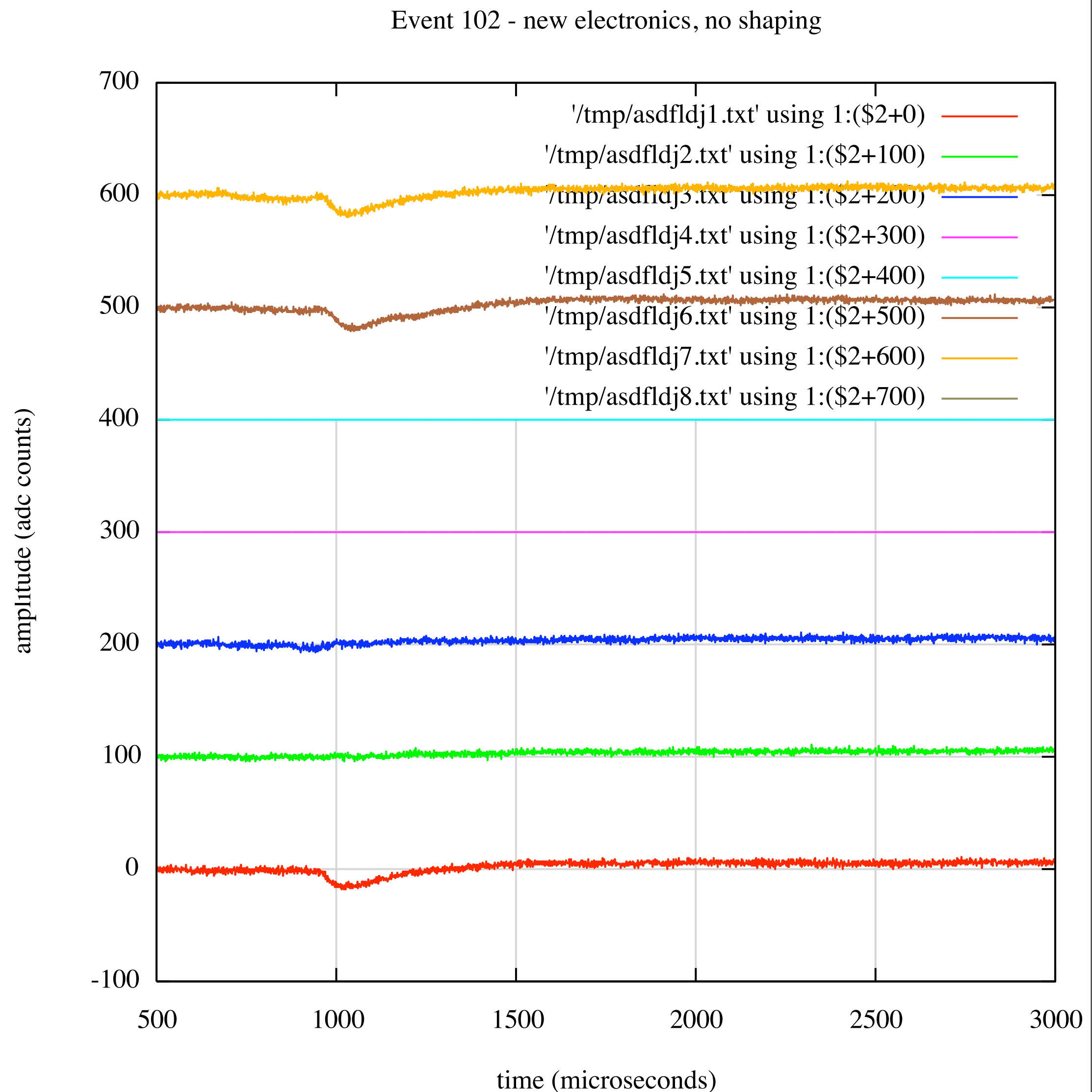
Cremat shaper + passive highpass



Prototype ‘A’

Neutron recoil-like event from 13th July run



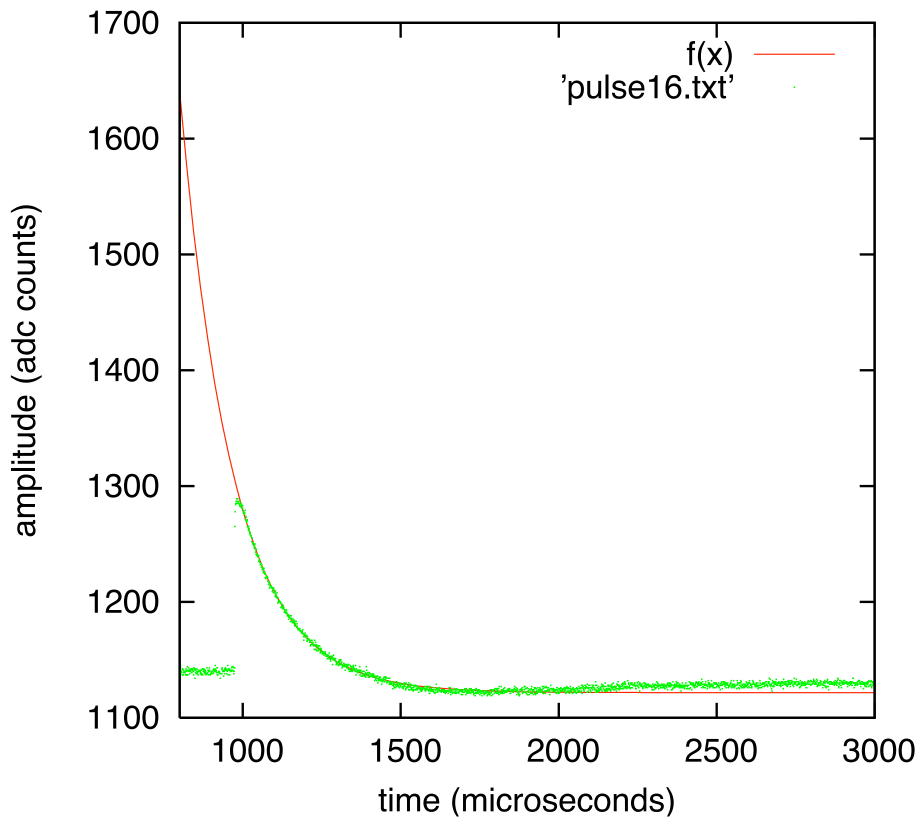


Some (3) channels flatlined due to DC offsets out of ADC range. Electronics now modified with tunable offsets to move all traces within dynamic range of ADCs.

Signal to noise ratio improved in the prototype, no discernable overshoots (none in this event type in the current electronics either), some interesting structure in the event tails is suggested.

Software shaping

It should be possible to use a filter to nullify the effect of the tail on all events introduced by the charge amplifier. This filter can be implemented digitally, as for the ‘inverse shaping’ filter described earlier.



Fit an event with a very fast risetime (event 16 from 13th July neutron run) to a decaying exponential. This event is a spark, so the charge pulse to each wire is effectively an impulse

Equivalent circuit for an exponential impulse response is an ‘integrator’ or more precisely an ‘exponential averager’ having time constant 167.6 us.

$$H(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} = \frac{1}{1 + sRC}$$

The inverse of this filter has the transfer function

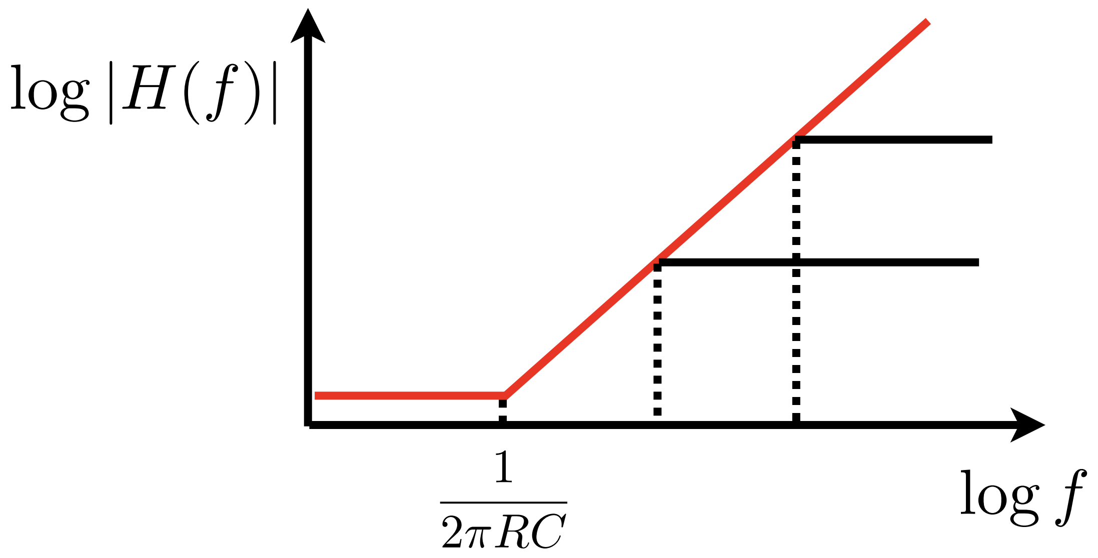
$$H^{-1}(s) = 1 + sRC$$

a single zero at s=-1/RC

Software zero-pole shaper

$$H^{-1}(s) = 1 + sRC$$

This transfer function rises as f at high frequencies, so the response diverges at high frequencies - an 'ultraviolet' divergence. To fix this, insert a pole at a higher frequency, between the zero frequency and the Nyquist cutoff.

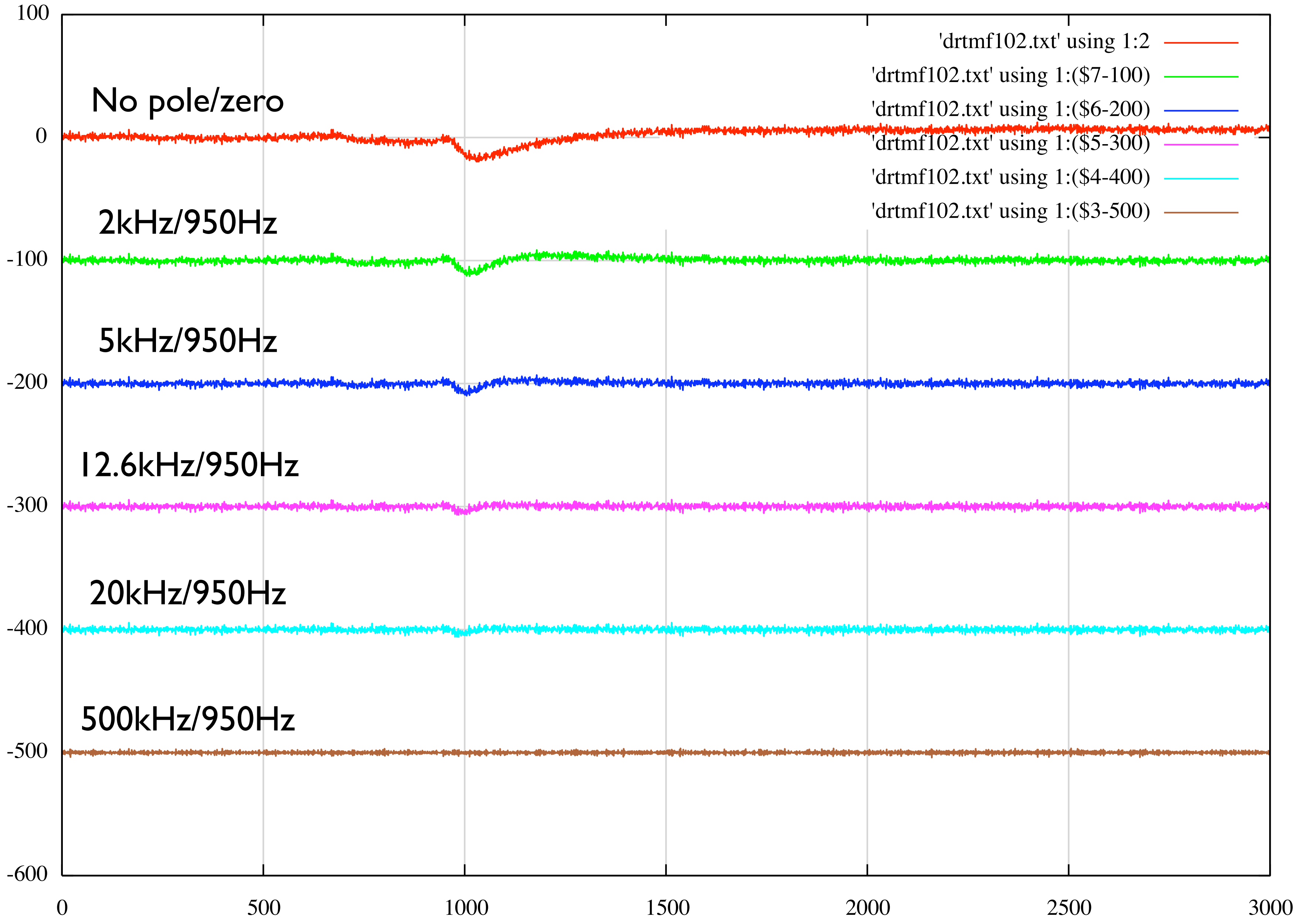


$$H_{PZ}(s) = \frac{1 + sRC}{1 + \frac{s}{2\pi f_P}}$$

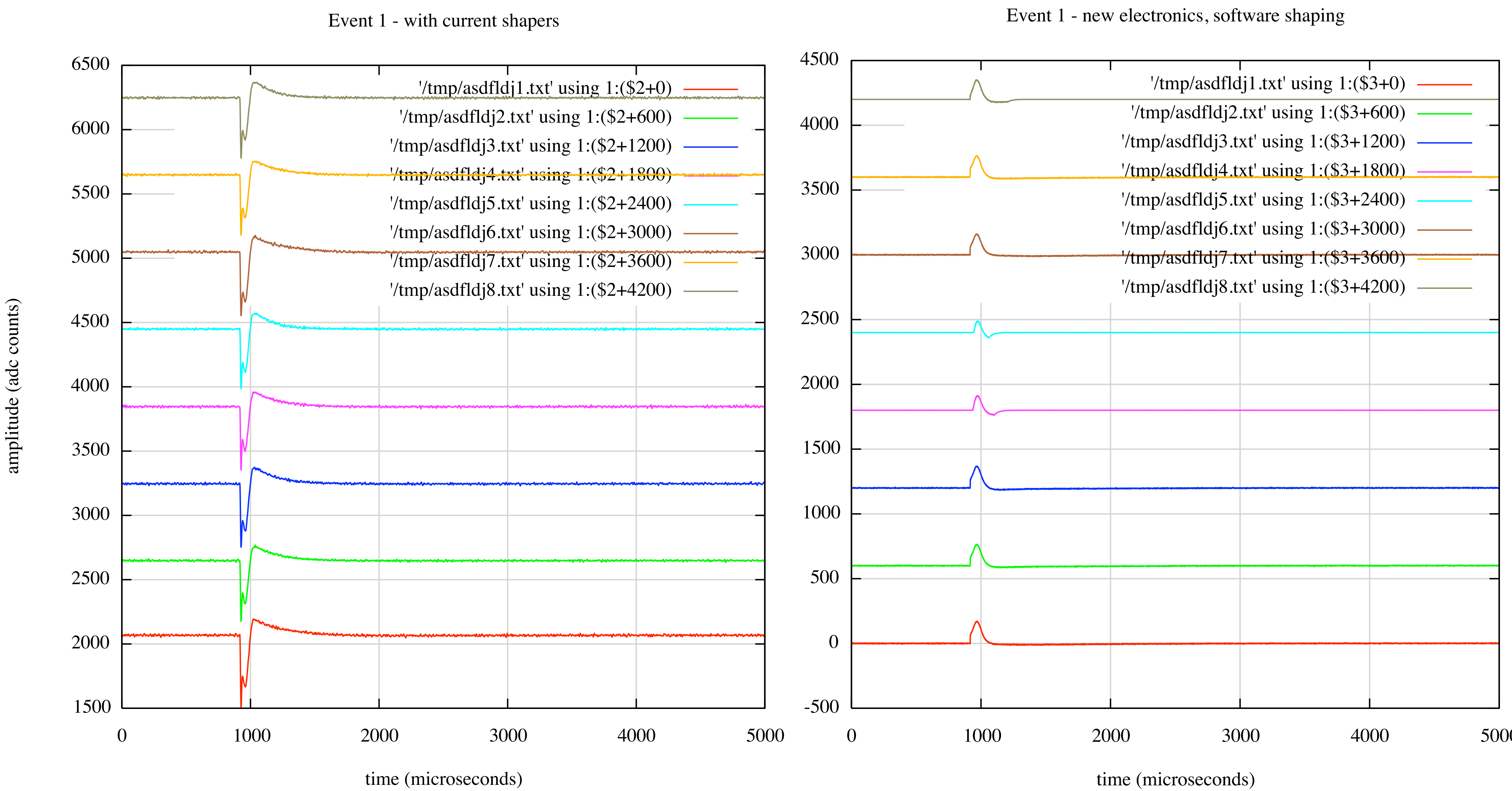
A higher pole frequency will more perfectly remove the exponential tail, BUT at the price of possibly boosting high frequency noise with respect to the signal, reducing the signal to noise ratio. This is why the unshaped output of my prototype has higher signal to noise ratio than the existing shapers, but software shaping will also run into this same issue.

Noise - Tail removal trade-off

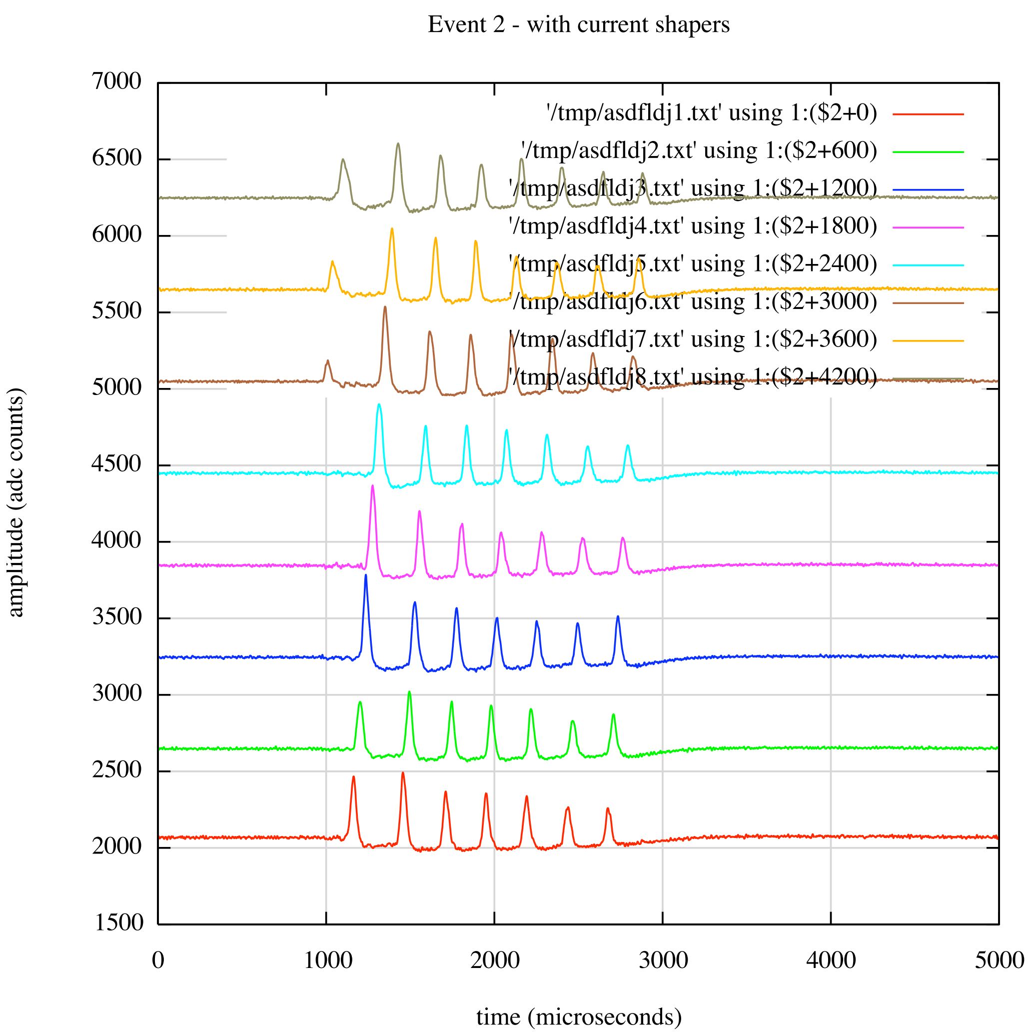
Different pole frequencies

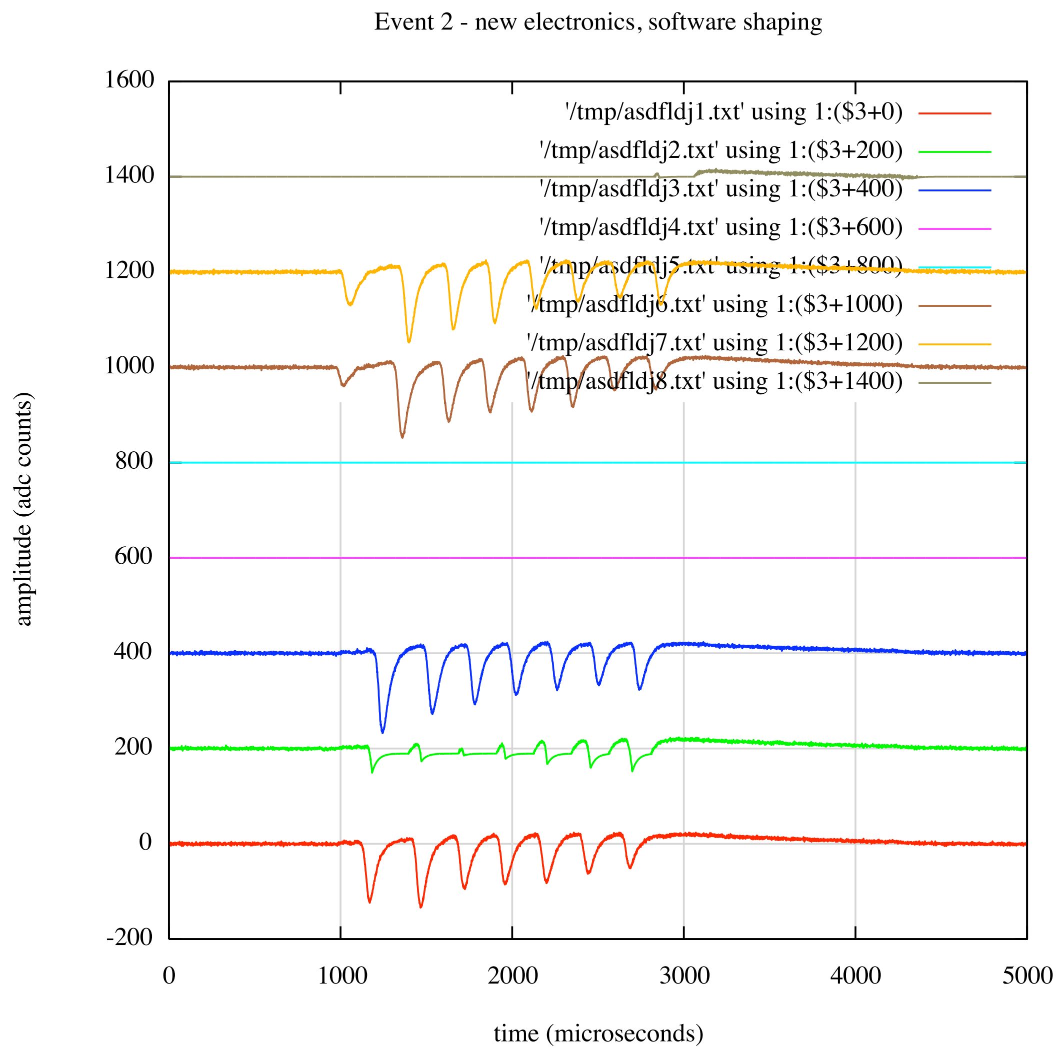


Pulse-to-ground Event



5kHz pole, 950Hz zero, on an Alpha Event





Conclusions

Prototype A electronics succeeds in removing overshoot on events studied without compromising SNR on neutron-like events

Software shaping can be run in real time so that either the raw data or filtered versions of it can be used with the software trigger

Second test of the prototype electronics on Friday - channel-specific offsets tuned out, triggering on output of prototype 'A', longer neutron run, higher pulse gain.

For the future - reduce the noise! Suggest differential readout of each channel to break ground loop between detector and input to electronics.

Calibration studies ! Directional studies !