Estimating V_{clip} and gain values of the Fireface 802

Author: Thejasvi Beleyur, Active Sensing Collectives group, Uni. Konstanz, May 2025.

I've had this suspicion that the Fireface 802 gain knob on channels 9-12 is *not* linear, and some pilot data of mine has already shown this. This notebook will now do a thorough job and calculate the gain at each of the knob positions, but also measure the V_{clip} of each channel.

How to measure V_{clip} and gain values?

The idea is that we feed in at least two different V_{rms} signals for each knob position to estimate both Vclip and gain value in a channel.

The max possible value of measured RMS on a channel is fixed: ($\frac{\sqrt{(2)}}{2}$~0.707) - and is hit when we reach the $Vclip$. We can thus calculate the RMS re max each time and use it to simplify our math.

$RMS_{norm} = \frac{V_{rms\,in}}{V_{rms-clip}}$ (eq. 1)

And when we take measurements with a known $V_{rms\,in}$, then the normalised RMS we get is:

$RMS_{norm} = \frac{V_{rms\,in}}{V_{rms-clip}} \times gain$ (eq. 2)

In principle we could already use this itself to perform parameter estimation with multiple $V_{rms\,in}$ values at each gain position, however I chose to convert this equation into a linear form in the decibel scale:

$dB(RMS_{norm}) = dB(V_{rms\,in}) - dB(V_{rms-clip}) + dB(gain)$ (eq. 3)

And by comparing the $dB(RMS_{norm})$ observed and expected for various values of $dBgain$ and $V_{rmsclip}$ we can then find the best estimates.

The setup

An oscilloscope + signal generator running at 1 Mhz/s sampling rate (Keysight INfiniiVision DSO-X-2004A was used to generate 20 kHz sinusoids with varying Vpp using a BNC-audio jack (6.3mm) cable.

The sinusoid voltage signal was fed into the Instrument line of channels 9-12 with differing Vpp and gain positions on the knobs in 'Lo gain' mode. The voltage signals had the following Vpp values: 100, 200, 500 and 1000 mVpp.



After multiple rounds of data collection, I realised only signals $\geq$ 50 mVpp were recorded and amplified properly - and thus only 100mVpp and above were used. *This meant that we DON'T have an estimate of position 8* - which is supposed to be +60 dB gain.

The gain positions

The gain positions were labelled 0-8 in a clockwise fashion.

The analysis

The estimation of gain for each position was done in two steps.

1. The *relative* gain of each position with reference to position 0 was calculated. The dBrms at each position was subtracted from that of the 0th position for a given input Vpp.
2. The *absolute* gain of the 0th position and $V_{rmsclip}$ of each channel was estimated using a least-squares fit with equation 3.

What the user-manual says

Gain values

The expected gain of the 0th position is +6 dB, and of the 4th position is +30 dB. The 8th position is expected to be +60 dB, but this could not be verified in the experiment because low signal voltages were not getting picked up properly despite the claim that a Vrms of -33 dBu could be fed into the instrument jack in the user's manual.

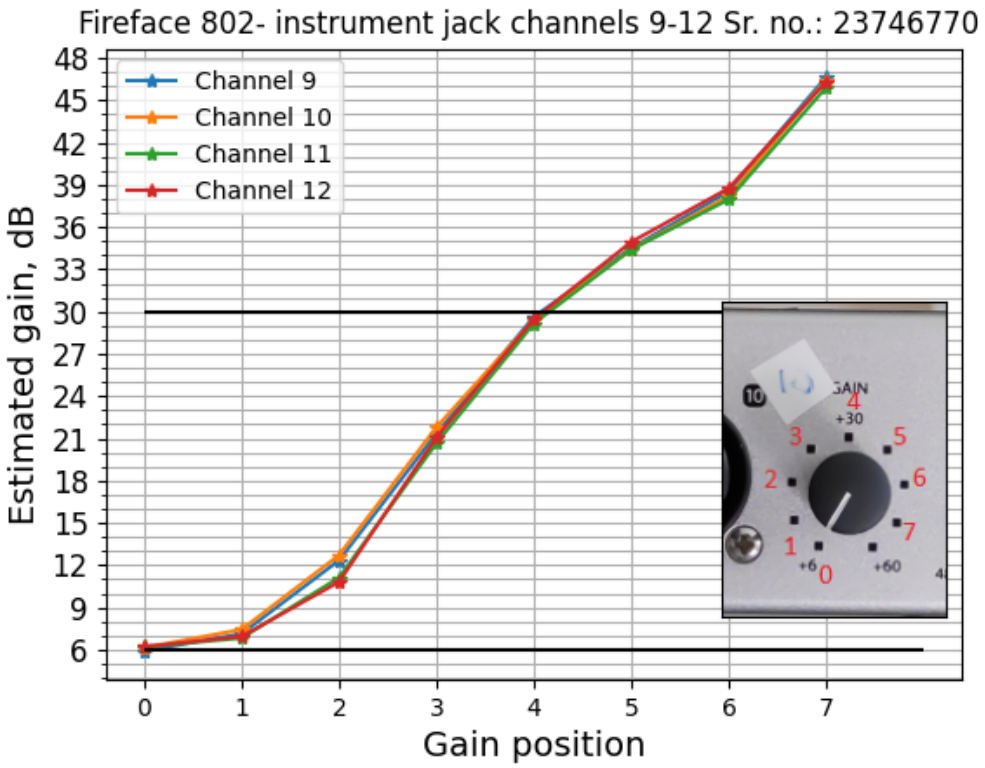
$V_{rmsclip}$

The $V_{rmsclip}$ is supposed to be 27 dBu, (the reference of 0 dBu is 0.775Vrms), which corresponds to 17.35 Vrms.

The results: Gain at various positions

The final results and the analysis workflow is outlined in more detail in the `ff802_gain_measurements.py` module. Here I will show only the results.

The +6 and +30 are correct ($\leq$ 1 dB difference), other positions show some channel-dependent variation



The estimated gain at various knob positions. Knob positions are numbered from 0-7 in a clockwise fashion (see inlay figure). Knob position 8 could not be estimated and is thus not labelled, even though measurements were carreid out.

In [1]:
```
import pandas as pd
import numpy as np
gain_at_positions = pd.read_csv('fireface802-srno_23746770_gainpositions.csv')
gain_at_positions.loc[:,['channel_num', 'gain_position', 'est_meangain_dB']]
```

Out[1]:

	channel_num	gain_position	est_meangain_dB
0	9	0	0.000000
1	9	1	1.235775
2	9	2	6.441169
3	9	3	15.448128
4	9	4	23.727640
5	9	5	28.621028
6	9	6	32.629014
7	9	7	40.751653
8	10	0	0.000000
9	10	1	1.287188
10	10	2	6.563361
11	10	3	15.648068
12	10	4	23.270305
13	10	5	28.278972
14	10	6	32.047875
15	10	7	40.319142
16	11	0	0.000000
17	11	1	0.624771
18	11	2	4.914910
19	11	3	14.462425
20	11	4	22.958737
21	11	5	28.210304
22	11	6	31.759413
23	11	7	39.713869
24	12	0	0.000000
25	12	1	0.733191
26	12	2	4.606063
27	12	3	14.858007
28	12	4	23.259190
29	12	5	28.741064
30	12	6	32.546739
31	12	7	40.106101

$V_{cliprms}$ and gain at position 0 across channels

Below is the estimate of Vcliprms and gain at position 0 across channels.

In [2]:
```
gain0_and_vclip = pd.read_csv('estimated_gain_pos0_vcliprms_fireface802.csv')
summary = gain0_and_vclip.loc[::2,['channel', 'avg_estgain_dB', 'avg_Vcliprms']]
summary
```

Out[2]:

	channel	avg_estgain_dB	avg_Vcliprms
0	9	5.903669	18.827784
2	10	6.148709	18.783548
4	11	6.204425	18.772865
6	12	6.209803	18.773275

$V_{cliprms}$ are $\leq$ 1 dB than company specs and very consistent across channels.

For the channels 9-12 instrument jack at 'Lo Gain', the Vcliprms is supposed to be 27 dBu. How different are the Vcliprms that we get?

The difference between company specs and our measurements are $\leq$ 1 dB, and the difference across channels is negligible (in the first and second decimal points).

In [3]:
```
summary['Vclip_dbu'] = 20*np.log10(summary['avg_Vcliprms']/0.775)
summary['Vclip_difference'] = summary['Vclip_dbu'] - 27
summary.loc[:,['channel','avg_Vcliprms','Vclip_dbu', 'Vclip_difference']]
```

Out[3]:

	channel	avg_Vcliprms	Vclip_dbu	Vclip_difference
0	9	18.827784	27.709950	0.709950
2	10	18.783548	27.689518	0.689518
4	11	18.772865	27.684577	0.684577
6	12	18.773275	27.684767	0.684767