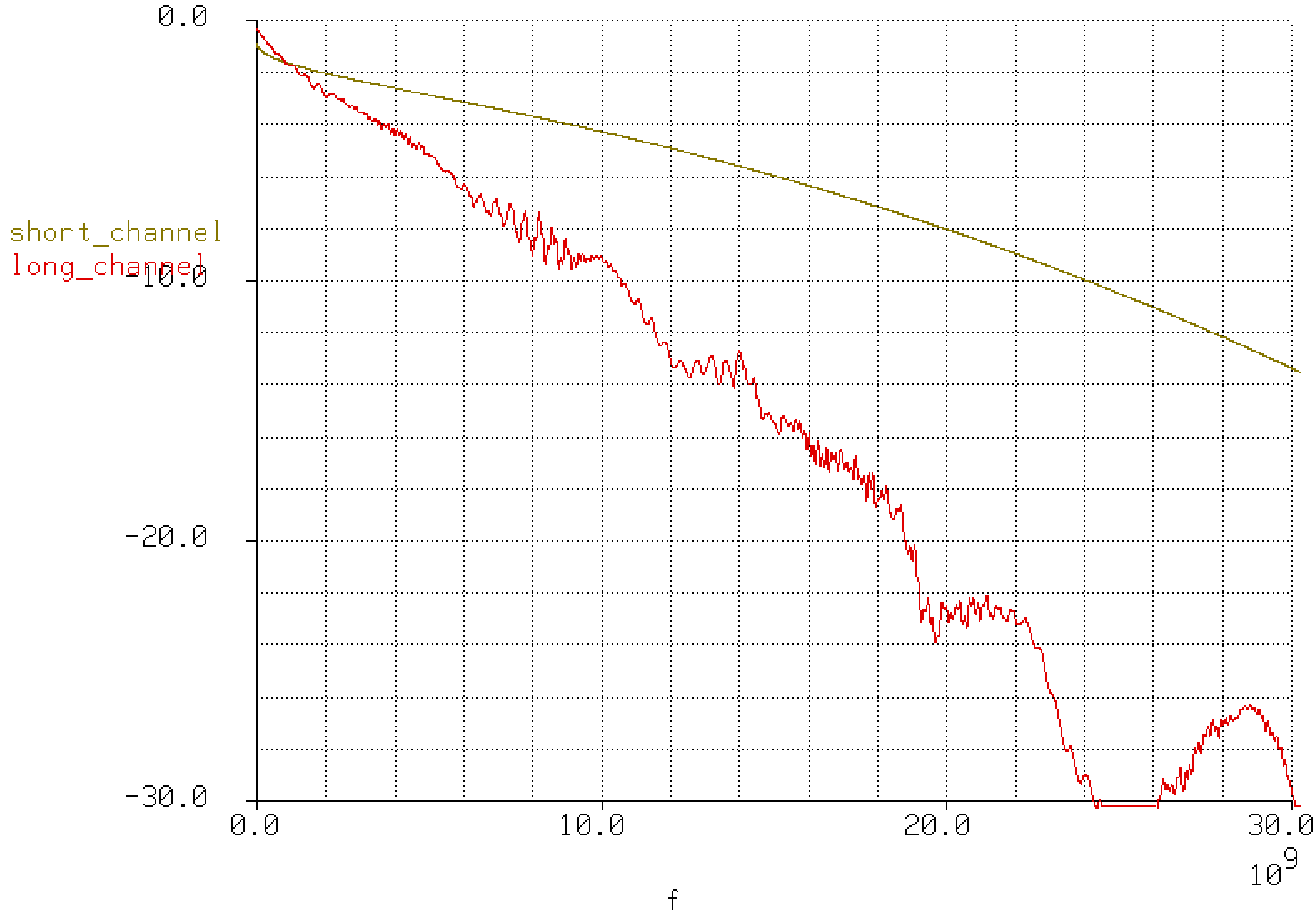
Measuring Transmitter equalization filter coefficients at the end of a channel with loss

Charles Moore Avago Technologies

IEEE802.3ba Clause 85.8.3.3 specifies a way of measuring transmitter transversal filter tap coefficients at TP2 after some amount of TP0 to TP2 loss, which could be up to 6.5dB at Nyquist, plus some chip and package attenuation. My simulations show that at these levels of loss the Clause 85 method gives reasonably accurate results although there is some error especially for c(-1). On the other hand for the Clause 92 TP0-TP2 with higher loss at Nyquist, simulations indicate significant inaccuracy especially for c(-1).

My simulations are done with 2 artificial channels, one using the synthesis technique described in benartsi_3bj_02_0513 but 154 mm long, which I will call the short model and is supposed to represent about loss of a typical Clause 85 TP0-TP2 channel but scaled to 25.78125Gb data rate. The other consists of the package model from Annex 93A attached to the host model: Quadra_8p25in_Pair8_9_THRU.s4p provided by Megha Shanbhag of TE Connectivity. I will call this the long model, and it is supposed to represent a Clause 92 TP0-TP2 channel.

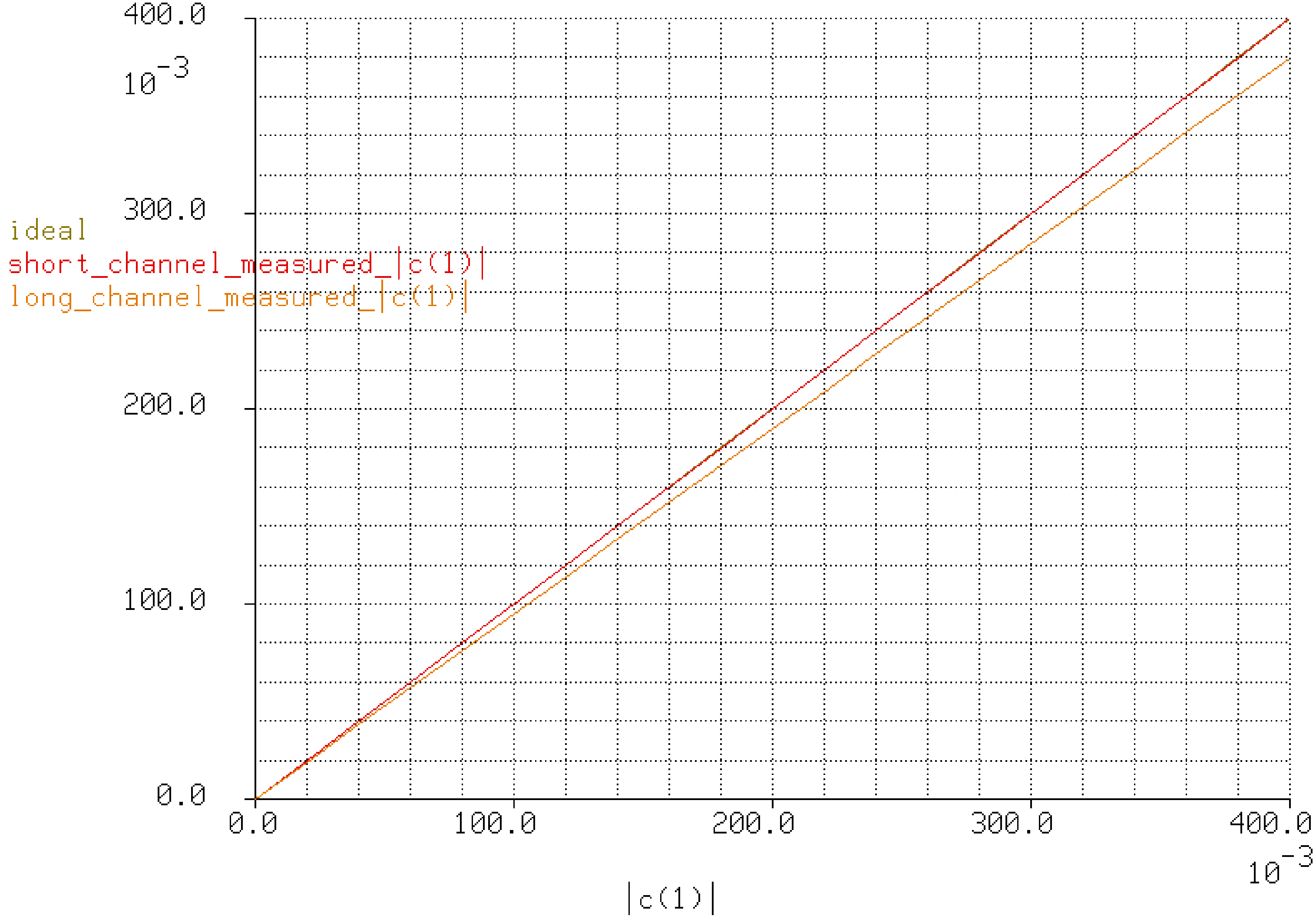


Channels used

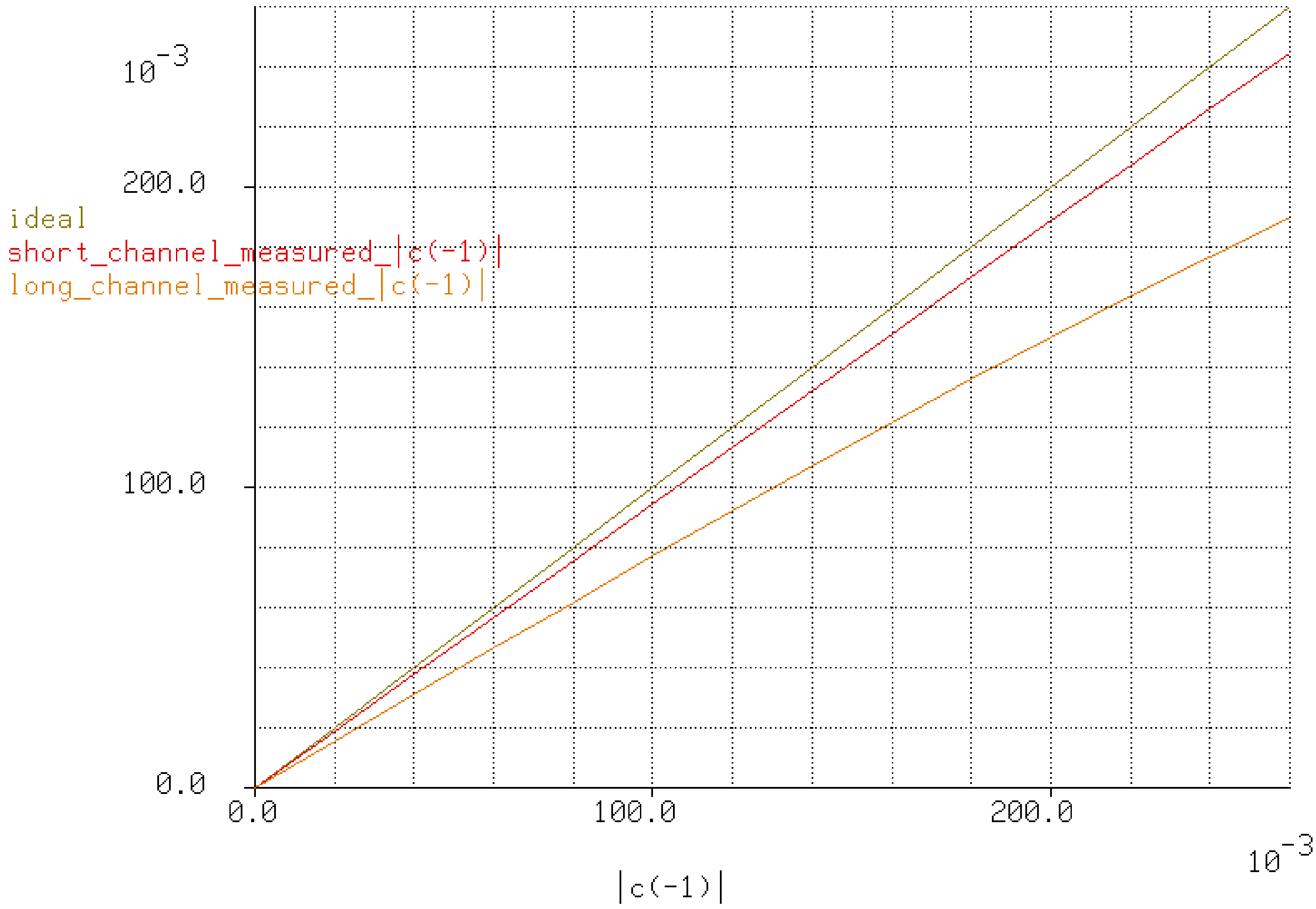
I simulated a PRBS9 pattern through the long and the short channel, with either precursor de-emphasis running in steps of -0.02 from 0 to -0.26 (ie (c(0) – c(–1))/(c(0) + c(–1)) running from 1 to 2.1 or postcursor de-emphasis running in steps of -0.02 from 0 to -0.40 ((c(0) – c(1))/(c(0) + c(1)) running from 1 to 5.0). I used a data rate of 25.78125Gb and 32 times oversampling.

I used the octave program “LinearFit” to extract the pulse response of the channel from the simulated PRBS response and “EqualizePulse” to compute the cursors. “LinearFit” and “EqualizePulse” were provided by Adam Healey of LSI to the members of the BA group in August 2009.

The following slides show plots of actual and “measured” values of precursor and postcursor taps vs actual.



Postcursor (c(1)) measured vs actual



Precursor (c(-1)) measured vs actual

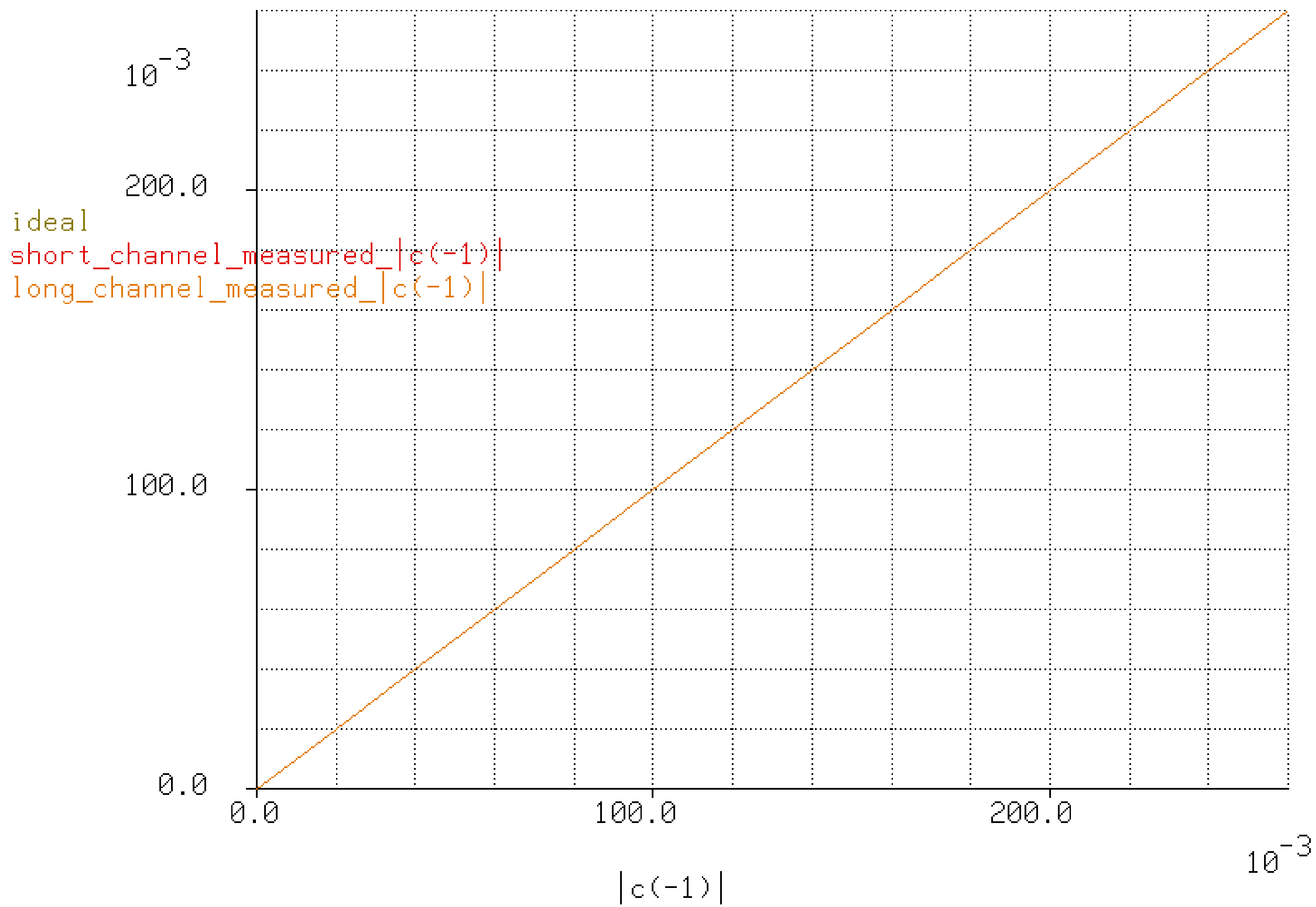
The post cursor measurement looks pretty good with a maximum error of only about 0.02 for the long channel. Pre cursor measurement is not too bad for the short channel, maximum error <0.02, which means that the clause 85 method is good at the loss level it was intended for, but the error is substantial for the long channel, >0.07. We need something better for Clause 92.

The problem comes about because the method uses the rising edge as a timing reference. Pre cursor equalization moves the rising edge. When symbol space samples are made to create the sampled pulse, the reference, un-equalized pulse and the equalized pulse are sampled at different times and the equalization of the sampled pulse is not really valid and gives inaccurate results.

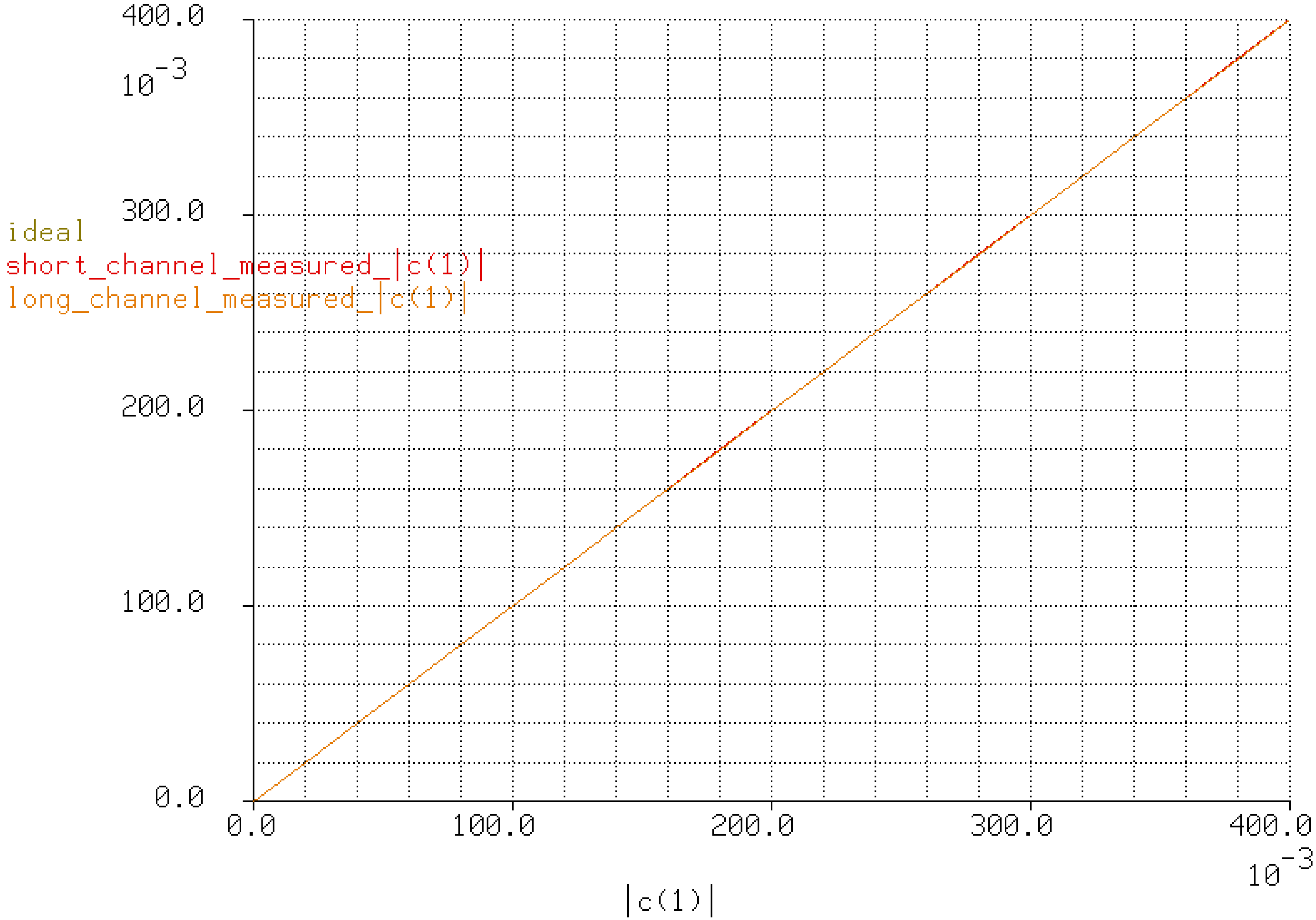
I can think of 2 ways to improve the calculation of the taps:

1. Find a better timing reference than the rising edge half height crossing point.
2. Find the best LMS fit of the equalized pulse to the sum of 3 symbol time spaced, weighted reference (unequalized) pulses, adjusting the timing of the start of the pulses to give the best LMS fit. The three weights are c(-1), c(0), c(1).

I have tried method 2 on my simulated data and find the error small. Unless someone is ready to state that method 1 is workable and gives good results I recommend using method 2 and making suitable changes to Clause 92.8.3.7.



Precursor (c(-1)) measure vs actual
Using method 2



Postcursor (c(1)) measure vs actual
Using method 2

If we use method 2, the process will converge much faster if we have a fast, closed form way to find the best LMS fit of the equalized pulse at any timing offset. Such a way exists let

$$b[i] = \text{base unequalized pulse } for \quad 0 < i \le MN \text{ , } 0\, othewise$$
$$p[i] = \text{equalized pulse } for \quad 0 < i \le MN \text{ , } 0\, otherwise$$
$$p_{fit}[i, j_{offset}] = c(-1) \cdot b[i - M + j_{offset}] + c(0) \cdot b[i + j_{offset}] + c(-1) \cdot b[i + M + j_{offset}]$$
$$\epsilon[o]^2 = \sum (p[i] - p_{fit}[i, o])^2$$
$$\epsilon[o]^2 = \sum p[i]^2 - 2 \cdot \sum p[i] \cdot p_{fit}[i, o] + \sum p_{fit}[i, o]^2$$
$$\begin{aligned}
\epsilon[o]^2 = &\sum p[i]^2 - \\
&2 \cdot \sum p[i] \cdot (c(-1) \cdot b[i - M + o] + c(0) \cdot b[i + o] + c(-1) \cdot b[i + M + o]) + \\
&\sum (c(-1)^2 \cdot b[i - M + o]^2 + c(0)^2 \cdot b[i + o]^2 + c(1)^2 \cdot b[i + M + o]^2) + \\
&2 \cdot \sum (c(-1) \cdot c(0) \cdot b[i - M + o] \cdot b[i + o]) + \\
&2 \cdot \sum (c(-1) \cdot c(1) \cdot b[i - M + o] \cdot b[i + M + o]) + \\
&2 \cdot \sum (c(0) \cdot c(1) \cdot b[i + o] \cdot b[i + M + o])
\end{aligned}$$

If we call

$$B0 = \sum b[i]^2 = \sum b[i+o]^2 = \sum b[i \pm M + o]^2$$
$$B1 = \sum b[i] \cdot b[i+M] = \sum b[i+o] \cdot b[i \pm M + o]$$
$$B2 = \sum b[i] \cdot b[i+2 \cdot M] = \sum b[i-M+o] \cdot b[i+M+o]$$
$$A_{-1}(o) = \sum b[i-M+o] \cdot p[i]$$
$$A_0(o) = \sum b[i+o] \cdot p[i]$$
$$A_1(o) = \sum b[i+M+o] \cdot p[i]$$

Then:

$$\begin{aligned}\epsilon[o]^2 = \sum p[i]^2 - \\ & 2 \cdot (c(-1) \cdot A_{-1}(o) + c(0) \cdot A_0(o) + c(1) \cdot A_1(o)) + \\ & B0 \cdot (c(-1)^2 + c(0)^2 + c(1)^2) + \\ & 2 \cdot (c(-1) \cdot c(0) \cdot B1 + c(-1) \cdot c(1) \cdot B2 + c(0) \cdot c(1) \cdot B1)\end{aligned}$$

To minimize with respect to c(n), let

$$\frac{d\ \epsilon^2}{d\ c(n)}=0$$

$$0=\frac{d\ \epsilon^2}{d\ c(-1)}=2\cdot(-A_{-1}(o)+c(-1)\cdot B0+c(0)\cdot B1+c(1)\cdot B2)$$

$$0=\frac{d\ \epsilon^2}{d\ c(0)}=2\cdot(-A_{0}(o)+c(-1)\cdot B1+c(0)\cdot B0+c(1)\cdot B1)$$

$$0=\frac{d\ \epsilon^2}{d\ c(1)}=2\cdot(-A_{1}(o)+c(-1)\cdot B2+c(0)\cdot B1+c(1)\cdot B0)$$

Giving 3 linear equations in 3 unknowns (c(n))

$$A_{-1}(o)=c(-1)\cdot B0+c(0)\cdot B1+c(1)\cdot B2$$
$$A_{0}(o)=c(-1)\cdot B1+c(0)\cdot B0+c(1)\cdot B1$$
$$A_{1}(o)=c(-1)\cdot B2+c(0)\cdot B1+c(1)\cdot B0$$

With BN being constant with o and A_n and c(i) changing with o.