

Working with design targets in Najda Under Control (NUC)

Introduction

Najda Under Control (A.K.A. NUC) is the graphical user interface used to program the Najda DSP crossover board. NUC allows the user to import SPL curves for individual drivers, FIR coefficients for filters and biquad coefficients for IIR filters. The filter coefficients mentioned here are typically produced by other programs; however, in this application note we will employ NUC's built-in filter utility, which is understood to be the tool most users will use.

Assumptions

This application note is not intended to be a tutorial on how to use NUC, as the program operation is fairly simple, in my opinion. It is assumed that the user is able to master NUC independently.

Managing displayed SPL levels in NUC

When deciding how to handle unpredictable SPL levels in driver measurement data (this is dependent on user measurement method), NUC software developers elected to normalize the SPL to 0dB at 1kHz, based on the SPL curve that has the highest amplitude at 1kHz. You can see this indicated by the red arrow in the screen shot in Figure 1 below. This is the SPL trace for the driver in Channel #2.



Figure 1.

Unchecking Channel #2 turns off its SPL trace; NUC selects the curve with the next highest SPL at 1kHz (see Figure 2). Note that the levels of the two remaining traces have shifted up. The user should be mindful of this behaviour during the design process, whenever a channel is deselected.

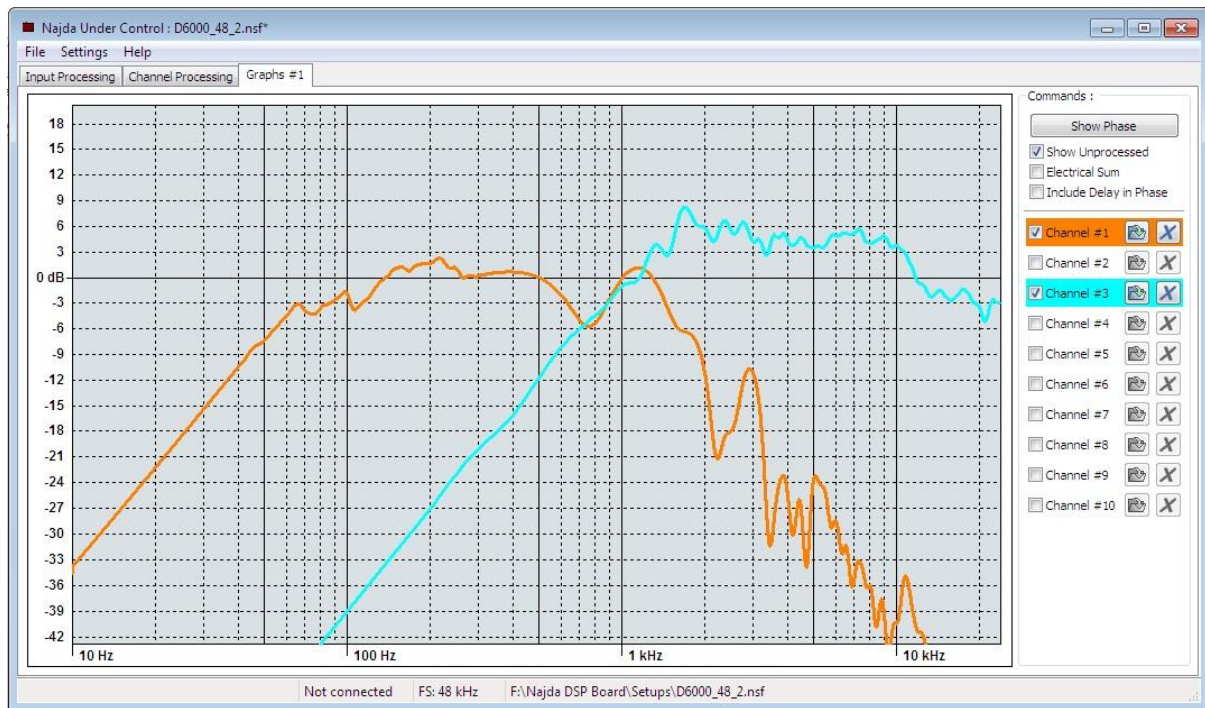


Figure 2.

The above described behaviour did trip me up, as I prefer to de-clutter the display by displaying only the particular channel that I am tuning at the time. I was unaware that this could cause the SPL level to shift, and it of course resulted in the incorrect final levels.

The Workaround

Objective: fix the SPL level after all SPL data has been imported, such that levels remain constant irrespective of any channels being de-selected.

Figure 3 shows a Notepad window titled 'Lister - [c:\Users\Shaun\Desktop\For Nick\SEASDXTD6.frd]' with a zoom level of 53%. The window contains a list of three columns of data: frequency (Hz), magnitude (dB), and phase (degrees). The data points are as follows:

Frequency (Hz)	Magnitude (dB)	Phase (degrees)
885.05	91.89	120.03
896.81	92.12	119.46
908.73	92.36	118.80
920.81	92.59	118.07
933.05	92.82	117.25
945.45	93.05	116.34
958.02	93.28	115.36
970.75	93.49	114.24
983.66	93.69	113.02
996.73	93.88	111.67
1009.98	94.02	110.23
1023.11	94.18	108.84
1037.01	94.21	107.55
1050.79	94.25	106.45
1064.76	94.27	105.66
1078.91	94.28	105.22
1093.25	94.29	105.17
1107.79	94.34	105.50
1122.51	94.43	106.17

Fig 3.

The fix: the fix is quite simple – in one of the unused channels, import a dummy SPL file that has a flat frequency response and magnitude higher than the highest of your design SPL files at 1kHz (a flat response will show up as a horizontal line, which will be least distracting, visually). Keep that channel selected, while you select/deselect your design channels at will.

To do so, you will have to view the SPL files in another application; something as simple as Windows Notepad will do. Note the highest SPL level in the 1kHz region (Figure 3). The first column contains the frequency points (Hz), the second column is magnitude (dB), and the last is phase (degrees). In the example, no point falls exactly on 1000Hz, but apparently the SPL around 1kHz is about 94dB. In the case of my own set of files, the highest level at 1 kHz was 97.07dB, so the dummy file will have to have all magnitude points above this level.

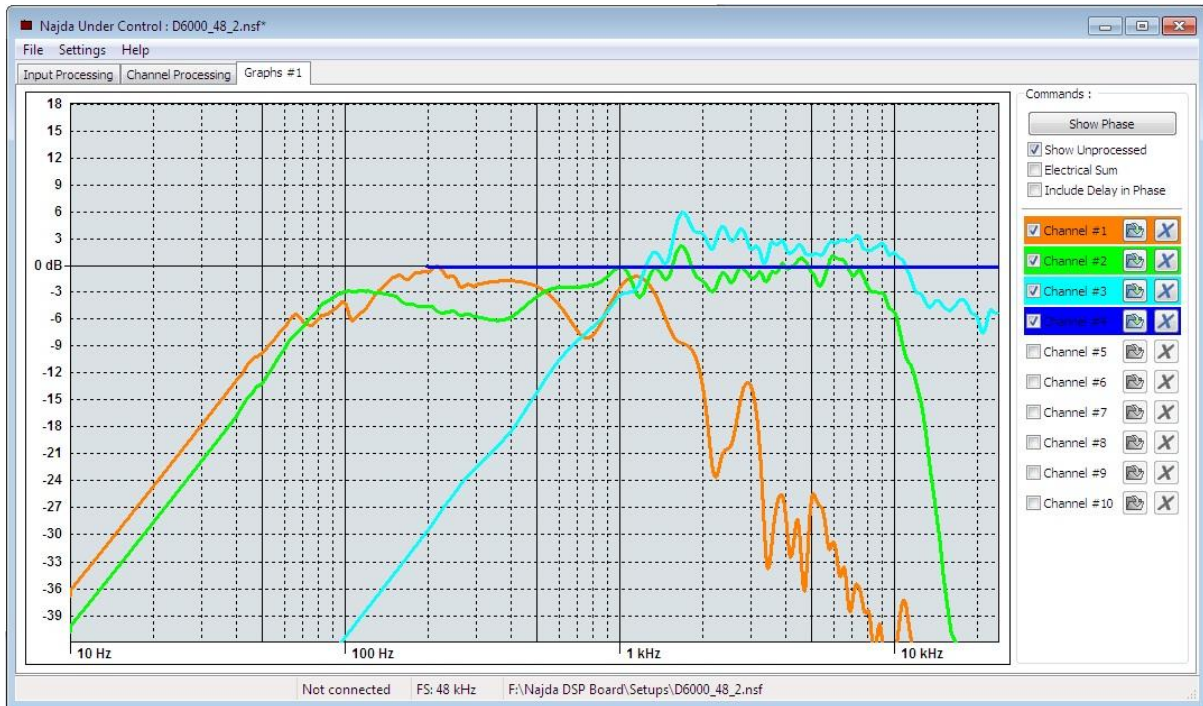


Figure 4.

As I only discovered this fix after having completed my design, for demonstration purposes, I chose a level only slightly higher at 97.1dB so as not to upset the work done already. You might choose to select a level around 10dB or 15dB higher, to move the dummy trace up and out of the way. Figure 4 shows the flat SPL file imported into Channel #4. Now de-selecting Channel #2 does not cause the levels to shift (Figure 5).

PS. I do not know why the dummy trace does not extend all the way to the left of the graph. It may have something to do with the reduced amount of data points that I used in the file when compiling it. Nevertheless, it does the job.

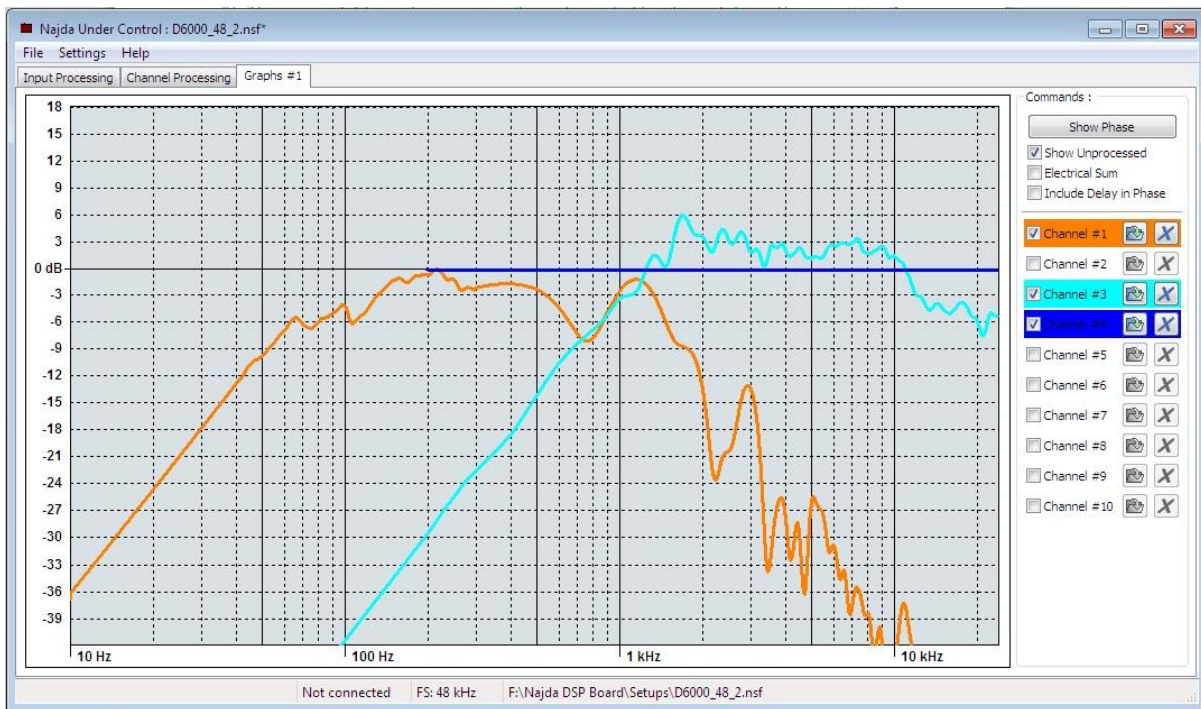


Figure 5.

Designing with target response curves

Tuning a crossover network is greatly simplified by having available an overlay of the target response curves for the respective drivers, to which the processed curves can be graphically matched. While NUC does not have an explicit target response function, the functionality does exist. It is done by setting up the desired response in some unused channels, using the standard filters available in the Channel Processing tab.

For my own project, the targets were as follows:

Bass: -3dB low frequency limit 30Hz; low-pass 200Hz Linkwitz-Riley 4th order

Midrange: band-pass symmetrical Linkwitz-Riley 4th order; 200Hz/2300Hz

Tweeter: high-pass 2300Hz Linkwitz-Riley 4th order

Channel assignment (left):

Bass: Channel #1; Target in Channel #8

Midrange: Channel #2; Target in Channel #9

Tweeter: Channel #3; Target in Channel #10

Figure 6 shows the targets and raw (unprocessed) driver responses. Target levels were set at -6dB, coinciding with the lowest in-band point amongst the drivers, so as to minimize the need for applying boost (as the woofer already requires a significant amount of EQ boost to meet the target, eating up valuable headroom). That point is indicated by the red arrow.

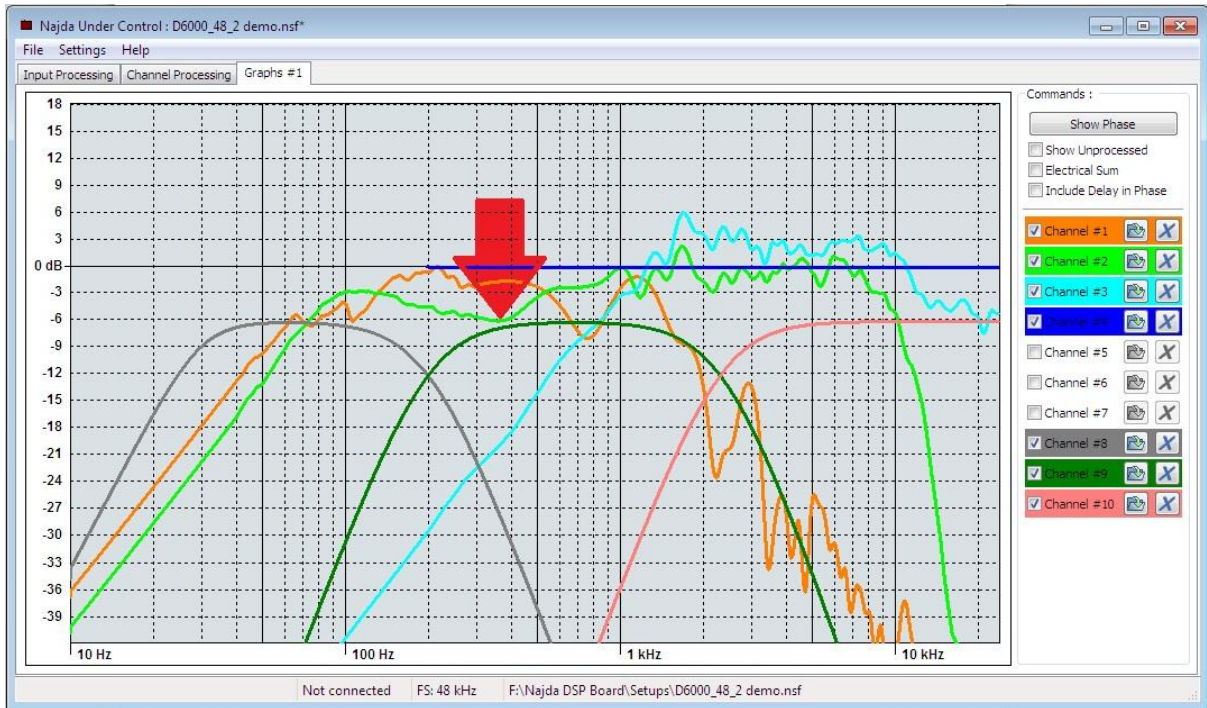


Figure 6.

I'll show how the midrange driver response was derived. The approach was to first apply EQ to flatten response, then the appropriate high-pass/low-pass filters. Figure 7 shows the midrange unequalised. Figure 8 shows the response flattened, and Figure 9 shows low-pass L-R 24dB/octave @2300Hz and high-pass L-R 24dB/octave @190Hz filters applied.

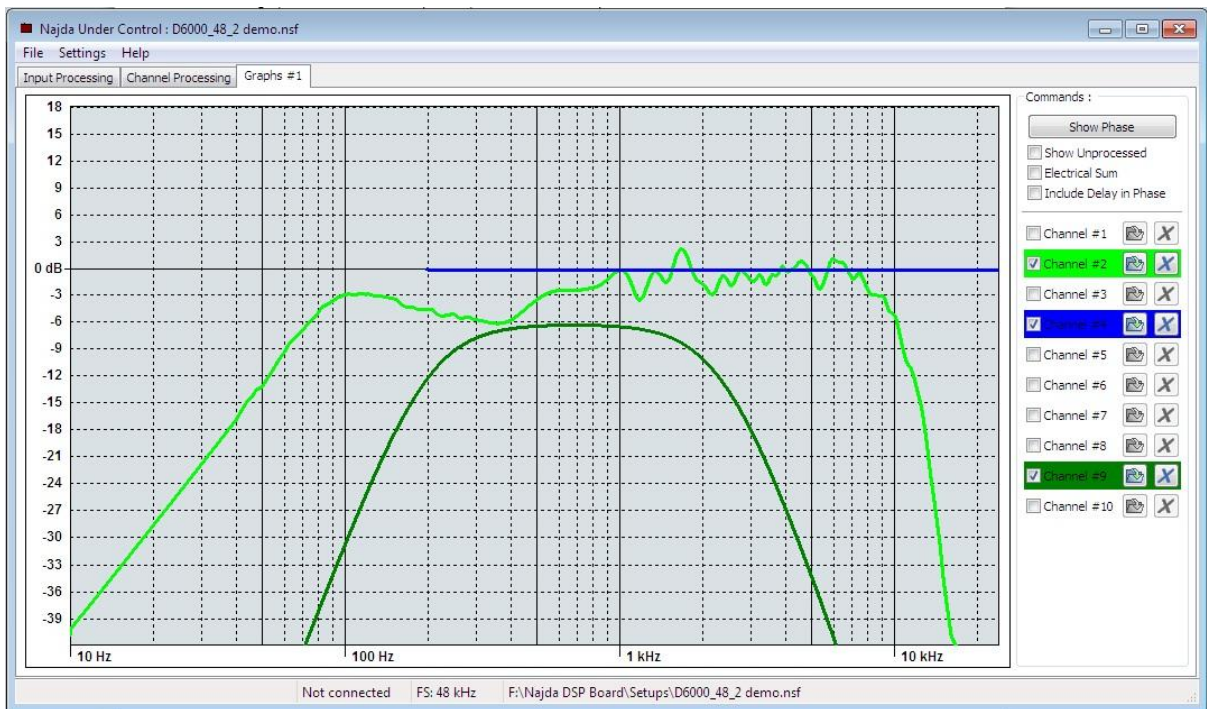


Figure 7

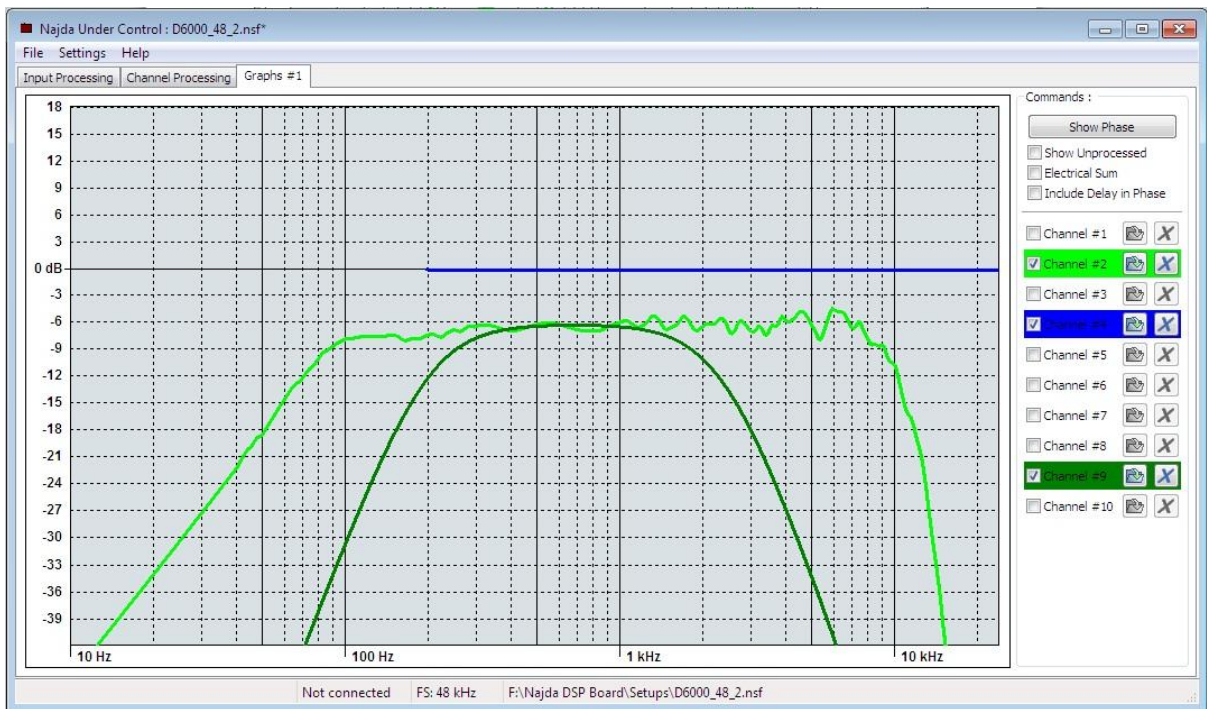


Figure 8.

The filters used will not necessarily be the same as that which describes the target response, as final response is a product of the driver's measured in-box acoustic response plus whatever filters and equalization are required to get it to meet the target response.

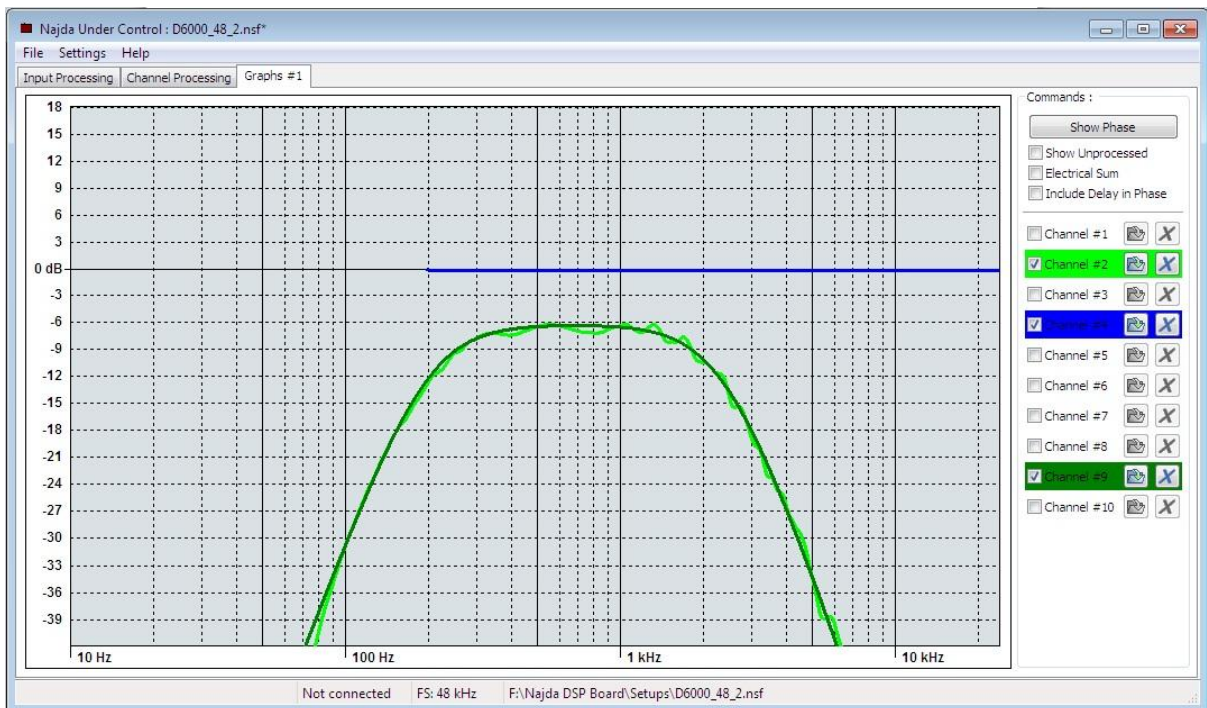


Figure 9.

Figure 10 shows the total EQ applied to the driver in order to align it to the target response (purple trace).

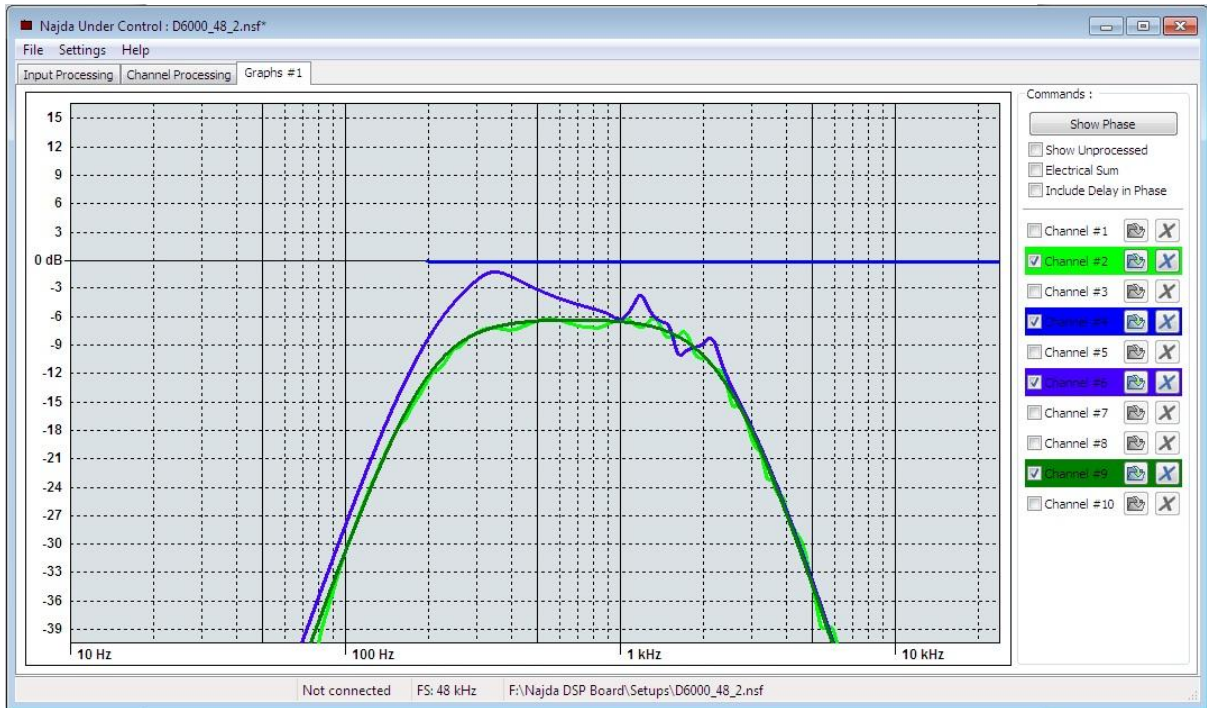


Figure 10

There are many ways to achieve the target response. The approach I have demonstrated may not be the best, though I do think it simplifies matters. One could, alternatively, apply the high-pass/low-pass filters first, though it then becomes trickier to EQ the drivers, due to some of the EQ points then falling on a slope, making it somewhat less easy to judge the amount of adjustment required.