

USER MANUAL

Station Board Filter Shapes

User Manual Document: A25044N0001

Revision: 1.0

D. Fort, 01Dec2011

*National Research Council Canada
Herzberg Institute of Astrophysics
Dominion Radio Astrophysical Observatory*

*P.O. Box 248, 717 White Lake Rd
Penticton, B.C., Canada
V2A 6J9*

Table of Contents

1	REVISION HISTORY	3
2	INTRODUCTION.....	4
2.1	STAGE 1	4
2.2	STAGES 2 WITHOUT MIXER	4
2.3	STAGE 2 WITH MIXER	4
2.4	STAGE 3	5
2.5	STAGE 4	5
3	OVERVIEW	6
4	CALIBRATION.....	7
5	PLOTS	8
5.1	STAGE 1 FILTERS	8
5.2	STAGE 2 FILTERS	8
5.3	STAGE 2 MIXER FILTERS	8
5.4	STAGE 3 FILTERS	8
5.5	STAGE 4 FILTERS	8
5.6	STAGE 4 FILTERS USING 512 TAPS.....	9

1 Revision History

Revision	Date	Changes/Notes	Author
1.0	01Dec2011	First Draft	D. Fort

2 Introduction

There are 36 Filter FPGAs on each Station Board. Each FPGA excepts one wide band input with a band width of 2048 MHz down to 128 MHz and can produce one narrow band output with a band width of 128 MHz down to 31.25 kHz. The EVLA input bandwidths are 1024 MHz (8-bit samplers) and 2048 MHz (3-bit samplers). The eMERLIN input band with is 512 MHz (8-bit samplers) where half of the samples belong to another band (the other polarization). The VLBA might have used a 128 MHz input band width with 15/16ths of the samples belonging to other bands (assuming sub-banding is done at the antenna).

The Filter FPGA consists of a 4 stages of FIR filtering in series. Each stage has a maximum of 512 taps set by the hardware resources. The hardware allows a lot of flexibility in the use of the 512 taps but one way was selected and is encoded into the existing file set, the CMIB software and Station Board GUIs. Normally, the filter shapes normalized by band width are the approximately the same for all bandwidths. An odd number of taps was chosen for all the filters because the pipeline delay through and odd number of taps is an integral number of clocks whereas an even number of taps produces a non-integral pipeline delay. Even though the number of taps is given as the full power of 2, the actual number of taps is one less.

2.1 Stage 1

The output band width of stage 1 is always 128 MHz. If the input is 8-bit samples then the number of taps in stage 1 drops to 256. If the input band width is 512 MHz and consists of 8-bit samples (eMERLIN), then only 128 taps are used. It is possible to use 256 taps but this is not supported by the files available at this time nor by the software interface. Because of this choice, the band shapes are similar to the 2048 and 1024 MHz input band width cases.

2.2 Stages 2 without mixer

Stage 2 always starts with a band width of 128 MHz and can be decimated by 2, 4, 8 or 16 yielding output band widths of 64, 32, 16 or 8 MHz using 64, 128, 256 or 512 taps respectively.

2.3 Stage 2 with mixer

It is possible to operate stage 2 in such a way that the input 128 MHz band can be shifted (rotated) in frequency by using a mixer phase model whose phase and rate can be updated every 10 milliseconds. This is useful for getting the area of interest away from slot boundaries. The number of taps available for filtering is effectively reduced by a factor of 2. The FPGA code is written and the coefficients are derived for a phase offset of +45 and -45 degrees (symmetry!) to produce the single side band mixer. It is likely possible to produce FPGA code and coefficients to do 0 and 90 with an increase in effective taps by a factor of 2 (or 4?). This upgrade was started but abandoned due to other pressures – something for the future. Because of the mixer it is possible to get away with only one filter per band width reduction because the area of interest can be shifted anywhere. The slots were chosen to stay away from DC when possible and to not require the sub-band flipper (even slot). It is also possible to shift the 128MHz input band with out decimating,

2.4 Stage 3

Stage 3 starts with a band width of 8 MHz (stage 2 *must* decimate by 16) yielding output band widths of 4, 2, 1 or 0.5 MHz using 64, 128, 256 or 512 taps respectively. The stage 2 filters without mixer are used.

2.5 Stage 4

Stage 4 starts with a band width of 0.5 MHz (stages 2 and 3 *must* each decimate by 16) yielding output band widths of 250, 125, 62.5 or 31.25 kHz using 64, 128, 256 or 512 taps respectively. . Normally, the stage 2 filters without mixer are used but it is possible to use the s4_* filters to take advantage of the extra processing capacity of stage 4 to yield 512 taps for all 4 decimations. In this case, only the transition width was reduced but the flatness of the pass band or the attenuation in the stop band could be increased instead.

3 Overview

The filters presented in this document are derived using a terribly inefficient method that has the advantage that it can derive filters with a given phase response. This is useful for fractional bit (phase slope) filters (VLBI) and mixer filters where the phase is not 0 or 90 degrees. The software is simple and is 'described' in A25285N0000. A single script (UNIX) can be executed to produce all of the files in ~20 hours. The software allows the pass band and its flatness, the stop band and its attenuation, the transition region width and the roll-off rate to be specified like most (more efficient) methods. The stop band attenuation was chosen to be 60 dB down from the pass band by EVLA requirement. The transition band width was chosen to be consistent with the number of taps available, pass band flatness desired and approximately 6 dB down at the nominal band edge. The transition width can be specified independently on each side of the band edge so that more of the transition region is outside of the desired band thus allowing the filters to be only 3 dB down at the nominal band edges. This is OK to a large extent because signals adjacent to the desired band are 'washed' by the frequency offset (WIDAR) scheme. In addition, the pass band can be made flatter by increasing the transition region width.

The filters are closest to the 'equal-ripple' designs of other software with a roll-off rate specified. The roll-off rate was chosen to be 10 dB per octave and comes nearly for free but the number of bits determines whether there is much of a roll-off left, especially for stage 1 (12-bit products). The other stages use 16-bit data and 16-bit coefficients. In all stages no bits are lost until the final scaling and symmetrical rounding to produce a 16-bit output from each stage.

4 Calibration

The plots in this document are not meant to be used for calibration of the data. A program exists (see A25285N0000) which produces the amplitude response of a given filter for a given band width and a given spectral resolution so that there will be one calibration number for each spectral bin. One (cal.exe) does each stage separately while another (gain.exe) attempts to combine all stages together to give a final number. Another program (snr.exe) calculates the SNR across the band which is useful for the band edge area.

5 Plots

There are three sets of plots, one giving the overall shape, one showing the pass band and one showing the pass band ripple in greater detail.

5.1 Stage 1 filters

s1_fir_04_* divides a 512 MHz input into 4 sub-bands where * = 00 to 03.

s1_fir_08_* divides a 1024 MHz input into 8 sub-bands where * = 00 to 07

s1_fir_16_* divides a 2048 MHz input into 16 sub-bands where * = 00 to 15

5.2 Stage 2 filters

s2_fir_02_* divides the 128 MHz stage 1 output into 2 sub-bands (* = 00 to 01)

s2_fir_04_* divides the 128 MHz stage 1 output into 4 sub-bands (* = 00 to 03)

s2_fir_08_* divides the 128 MHz stage 1 output into 8 sub-bands (* = 00 to 07)

s2_fir_16_* divides the 128 MHz stage 1 output into 16 sub-bands (* = 00 to 15)

5.3 Stage 2 mixer filters

s2_mix_01_00 mixes (rotates) the 128 MHz stage 1 output

s2_mix_02_00 mixes and filters the 128 MHz stage 1 output to get slot 0 of 2

s2_mix_04_02 mixes and filters the 128 MHz stage 1 output to get slot 2 of 4

s2_mix_08_02 mixes and filters the 128 MHz stage 1 output to get slot 2 of 8

s2_mix_16_02 mixes and filters the 128 MHz stage 1 output to get slot 2 of 16

5.4 Stage 3 filters

stage 3 filters are identical to the stage 2 filters.

5.5 Stage 4 filters

stage 4 filters can be identical to the stage 2 filters or see next section.

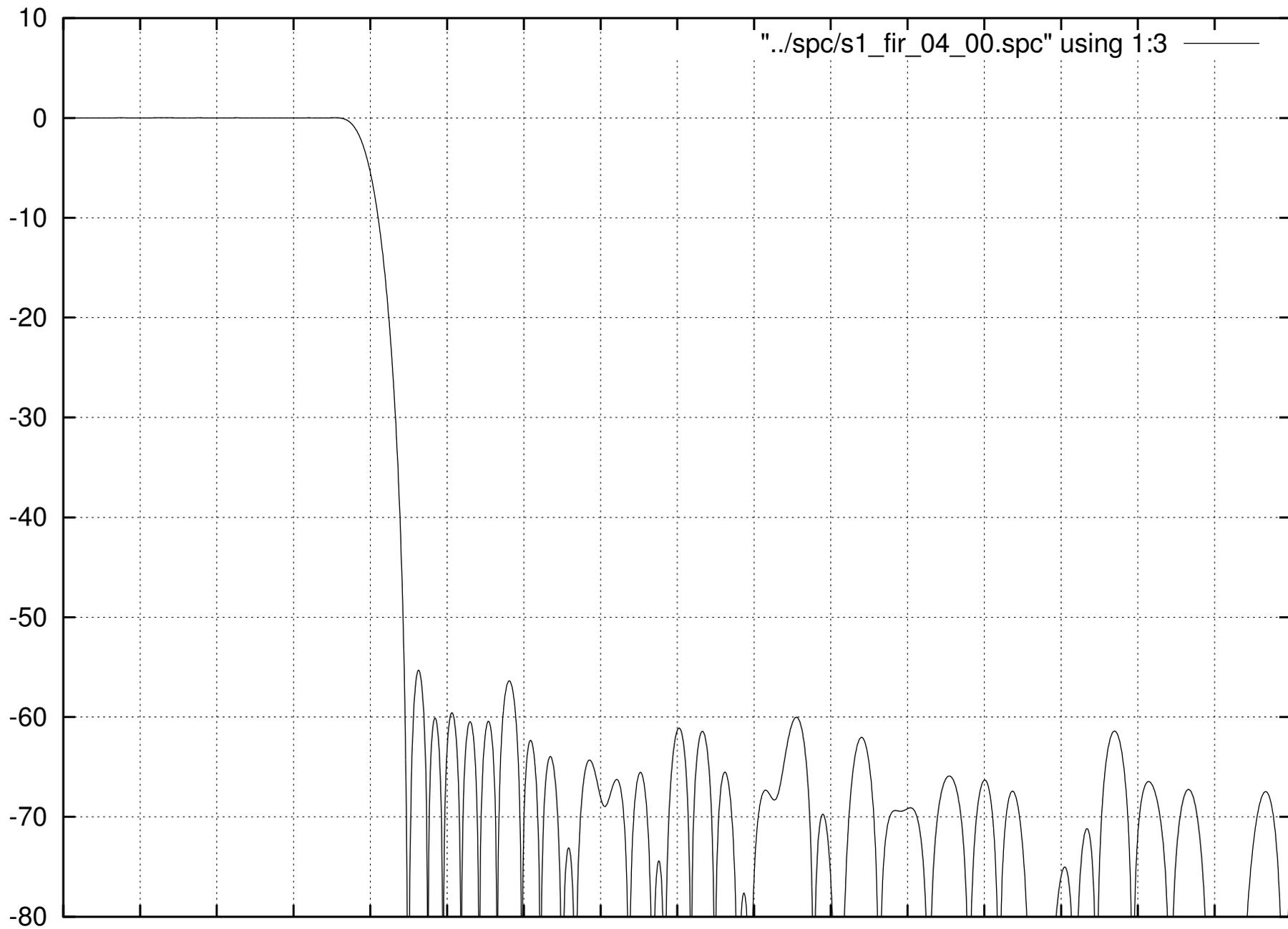
5.6 stage 4 filters using 512 taps

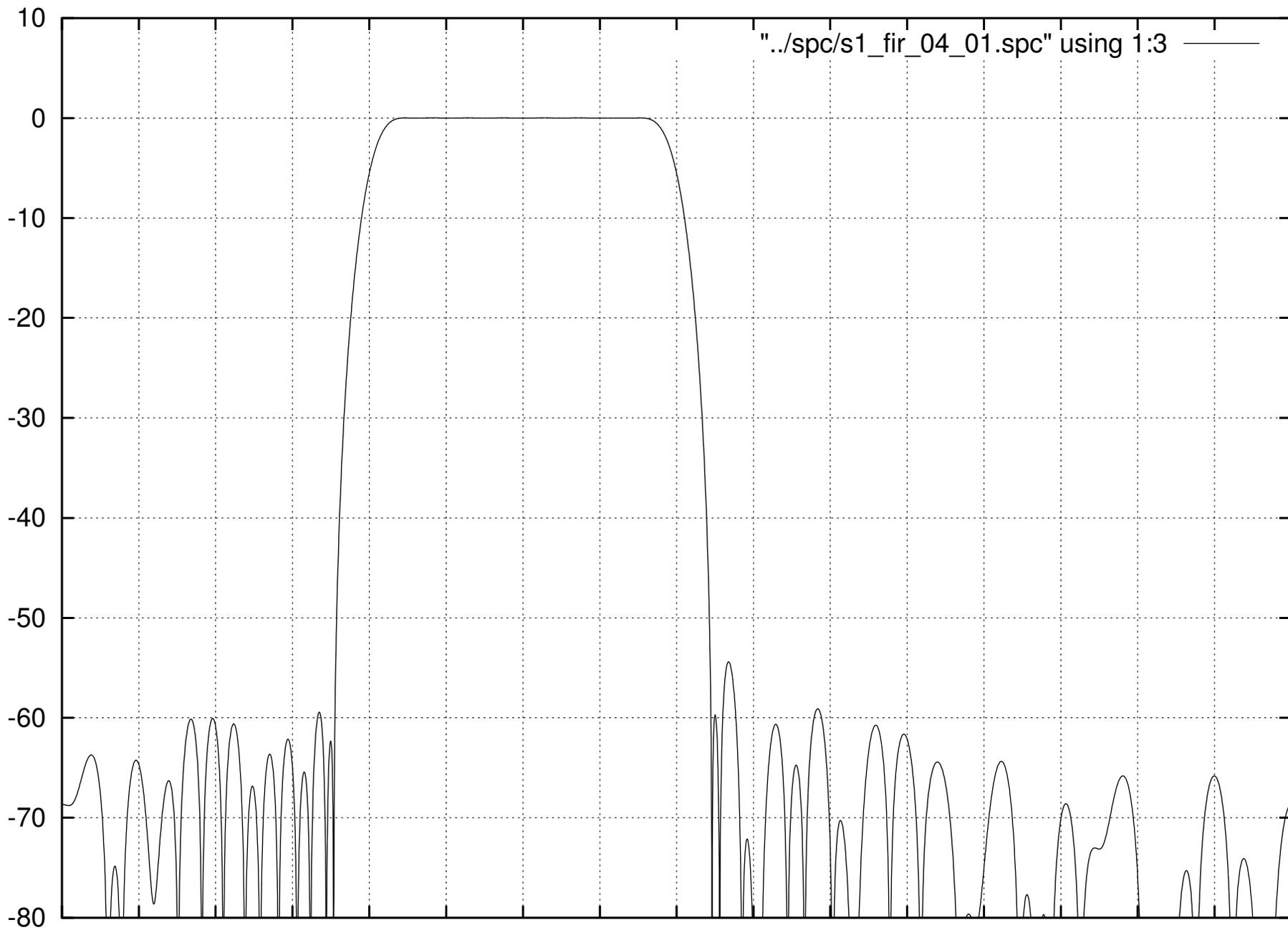
s4_fir_02_* divides the 0.5 MHz stage 3 output into 2 sub-bands (* = 00 to 01)

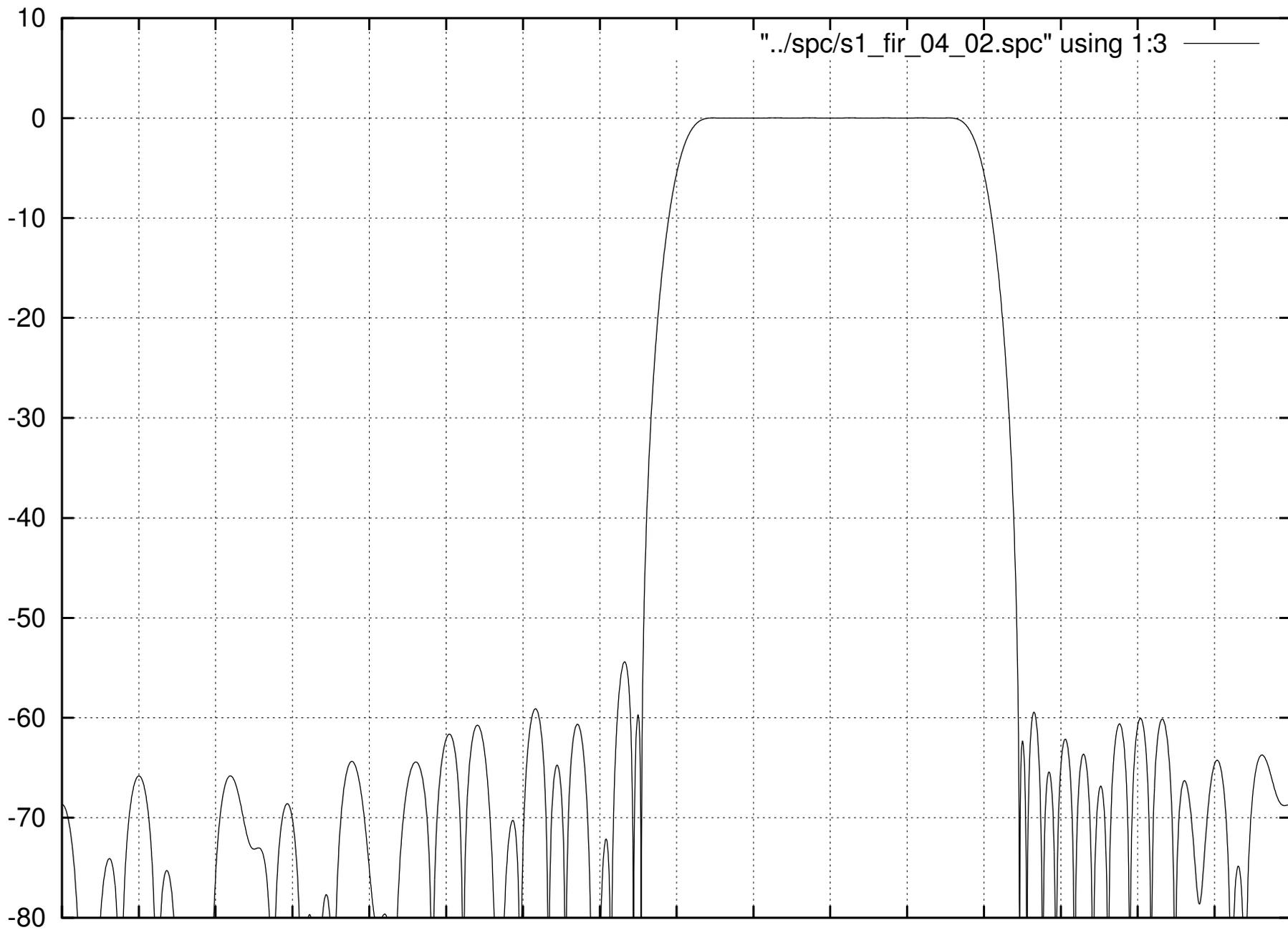
s4_fir_04_* divides the 0.5 MHz stage 3 output into 4 sub-bands (* = 00 to 03)

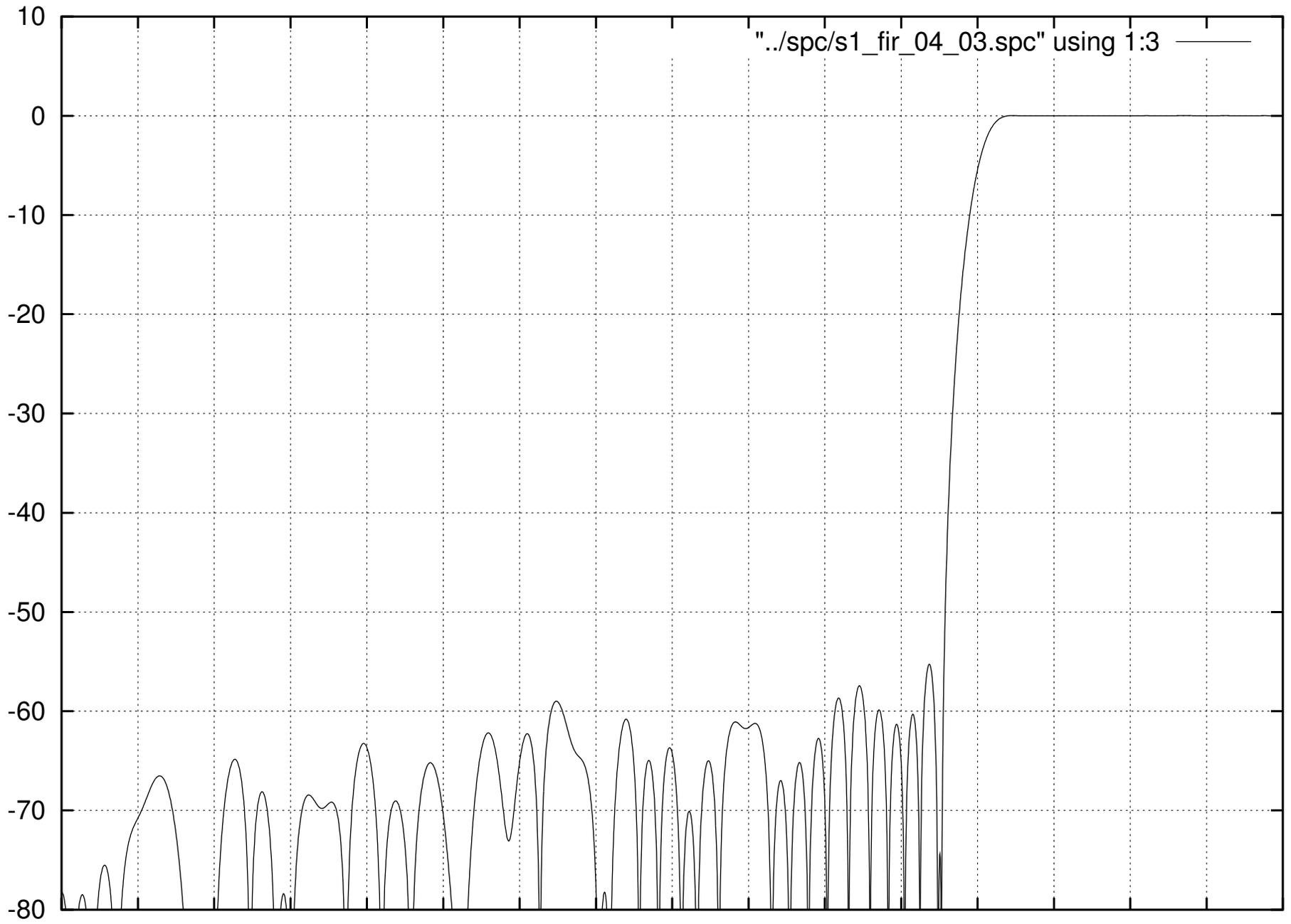
s4_fir_08_* divides the 0.5 MHz stage 3 output into 8 sub-bands (* = 00 to 07)

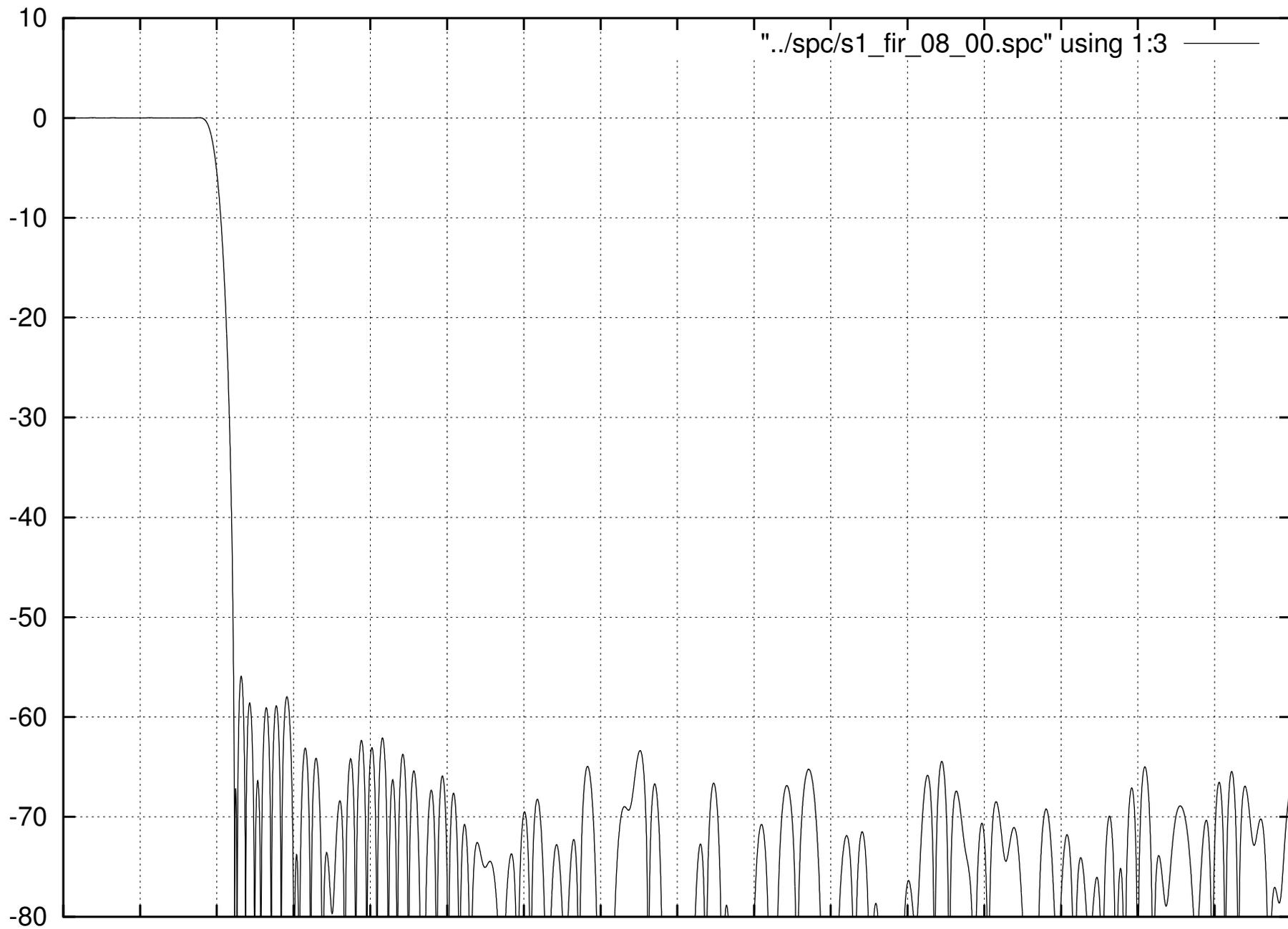
s4_fir_16_* divides the 0.5 MHz stage 3 output into 16 sub-bands (* = 00 to 15)

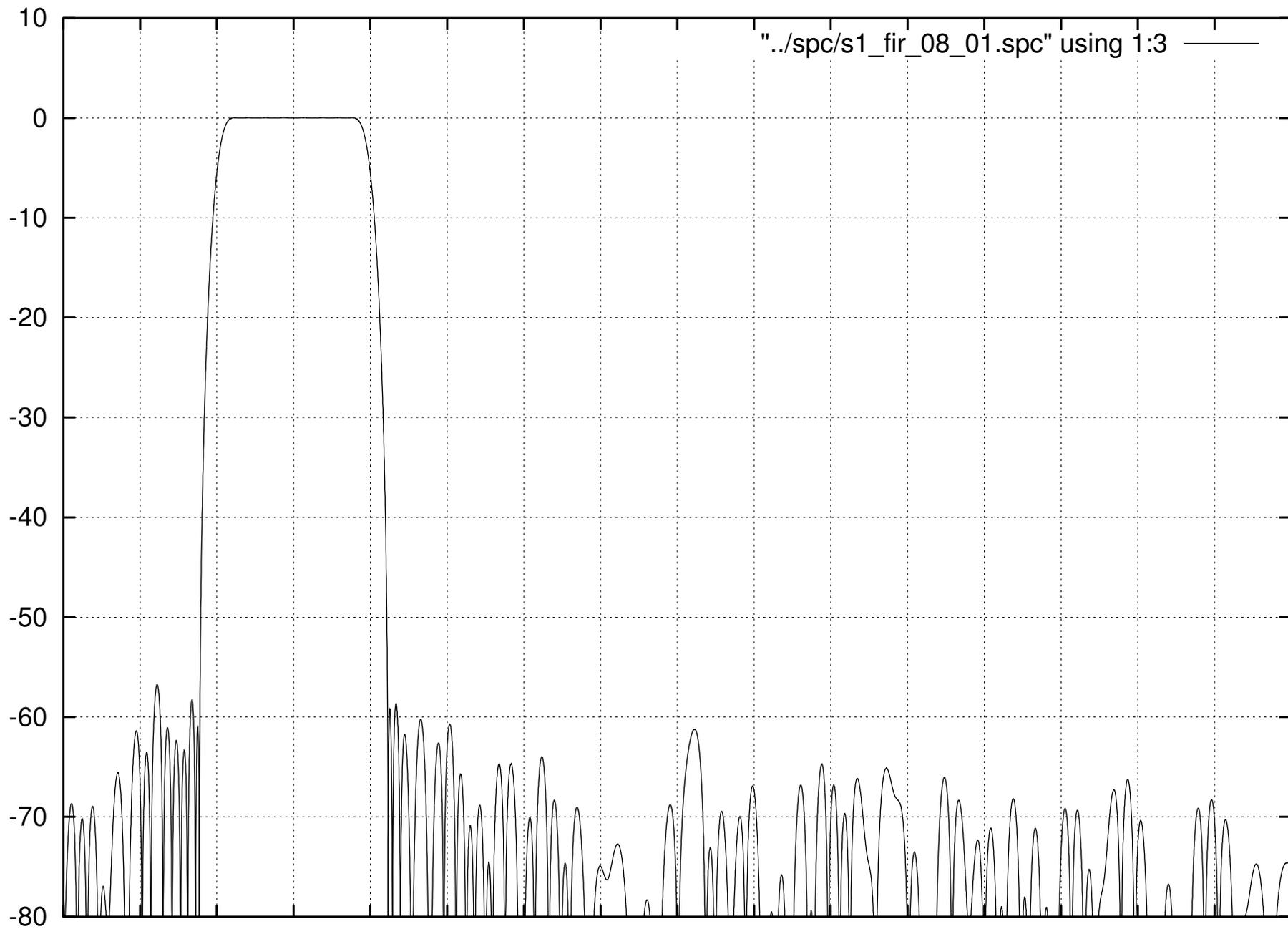


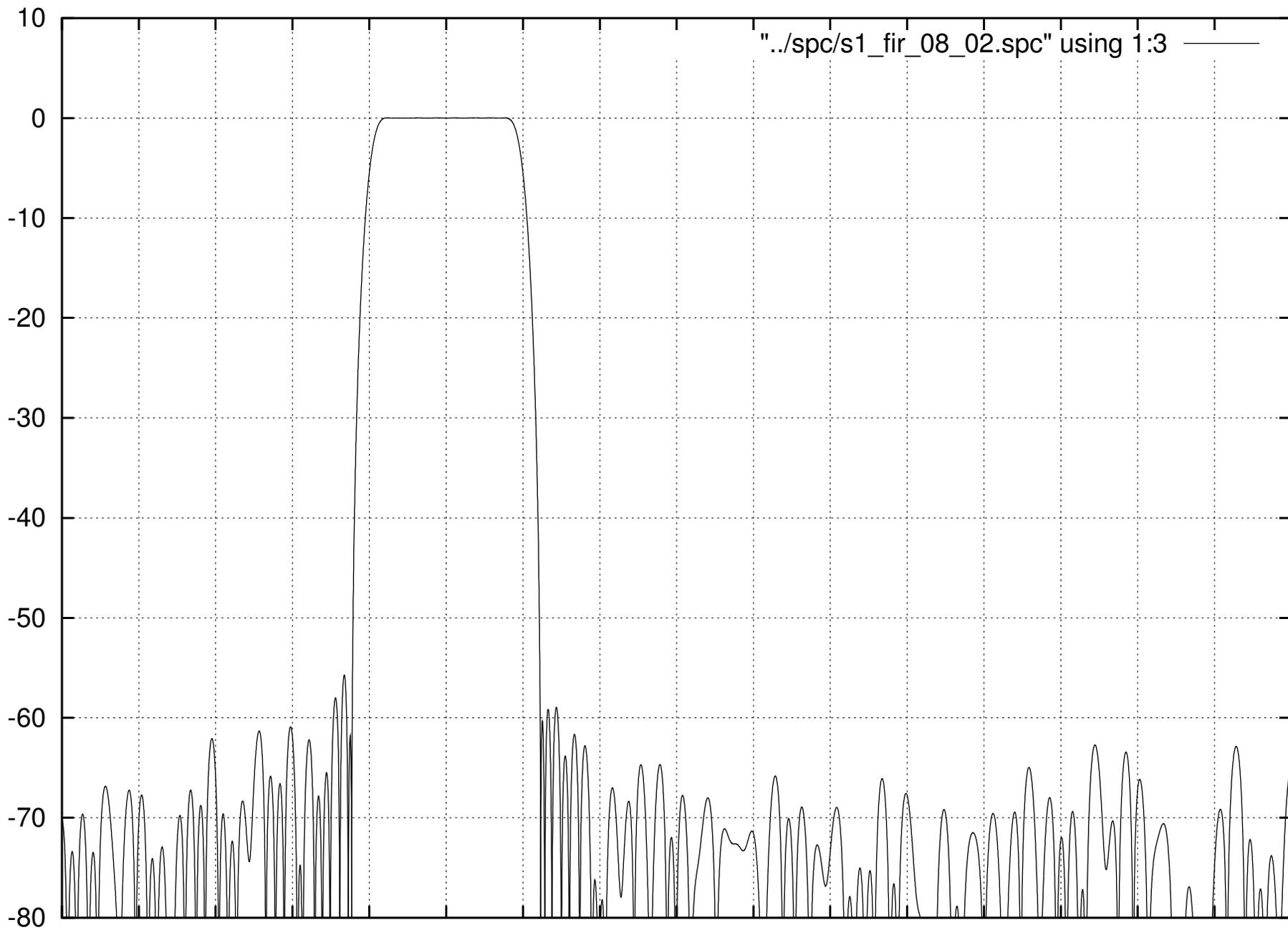


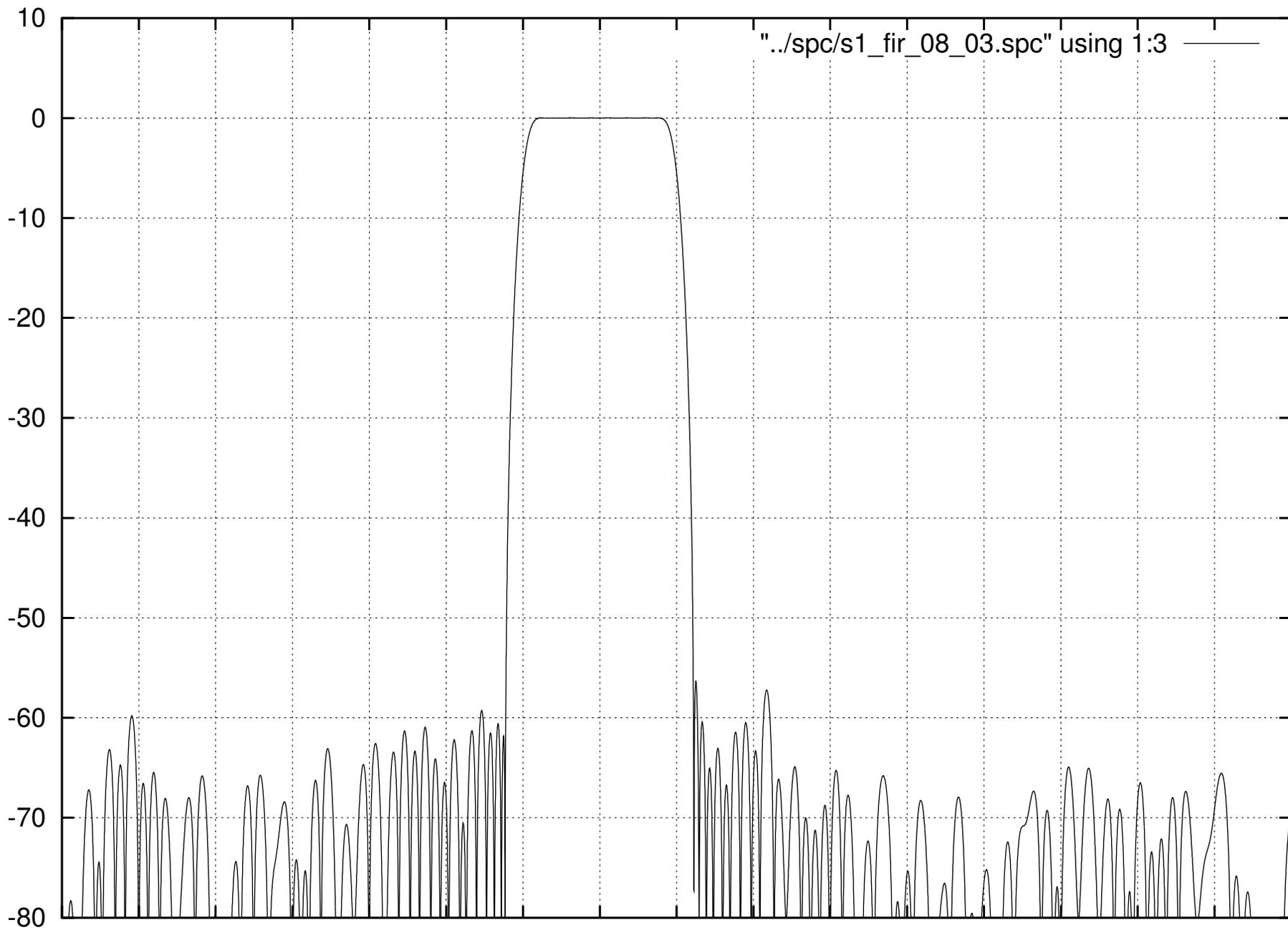


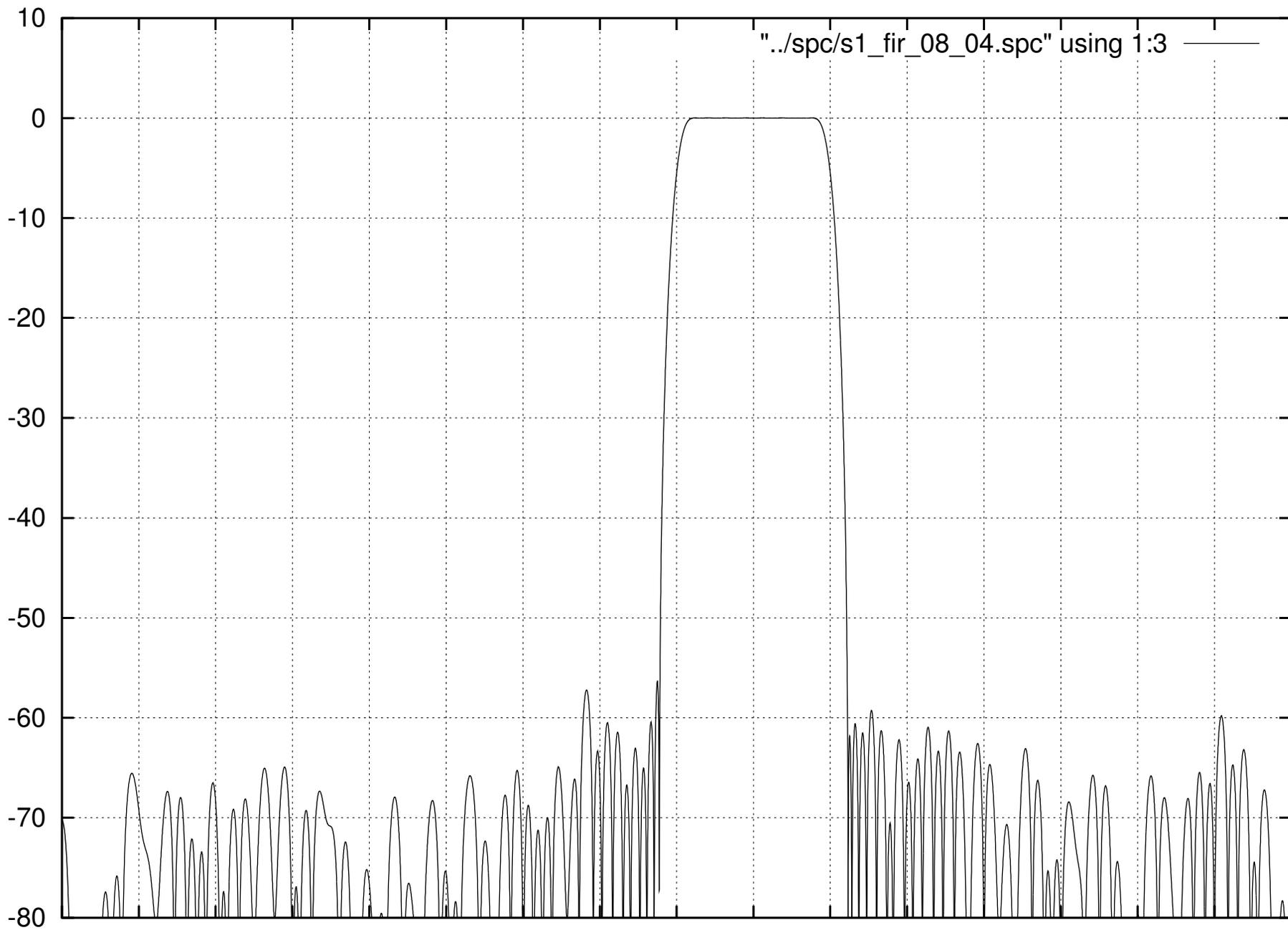


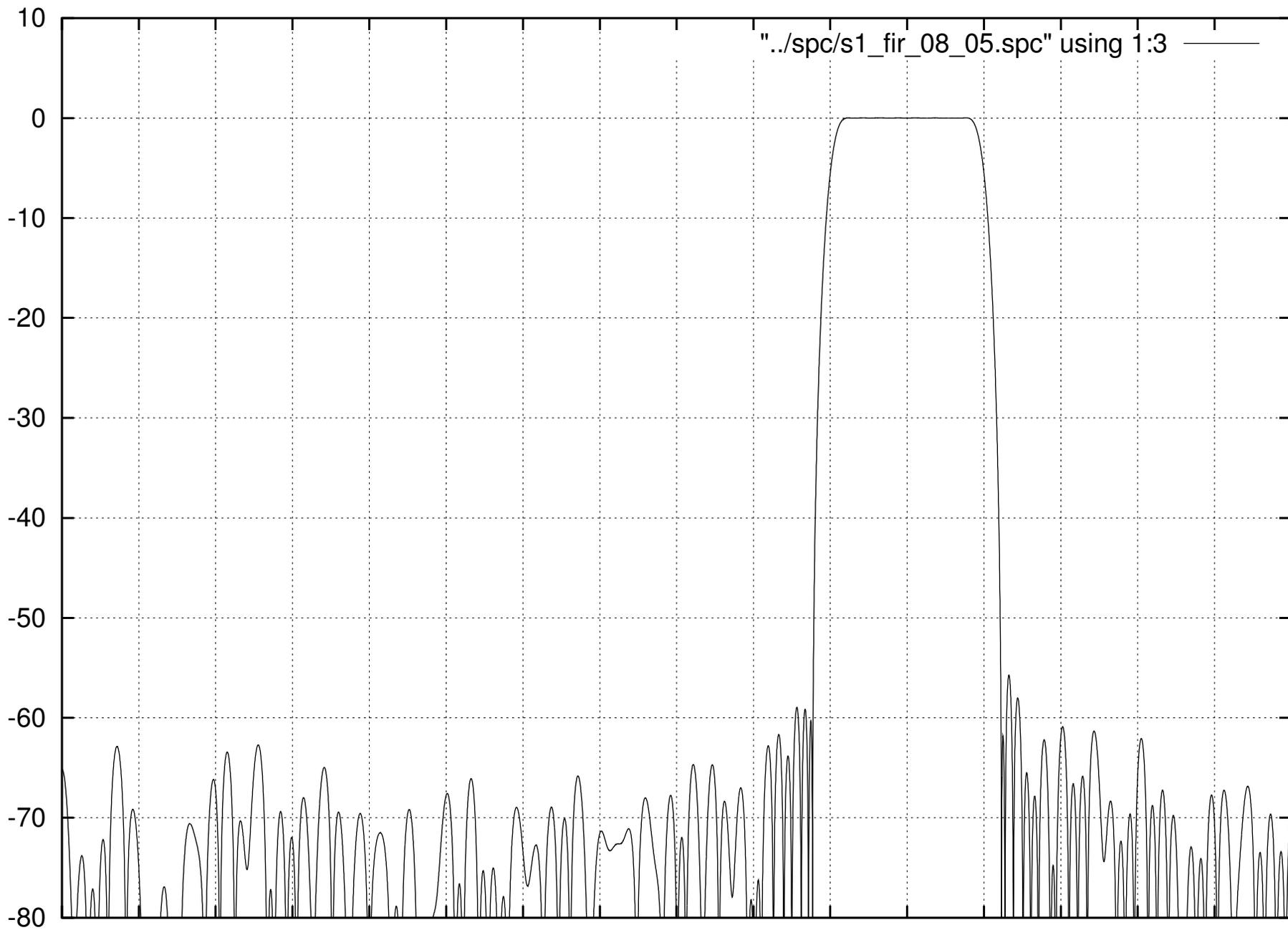


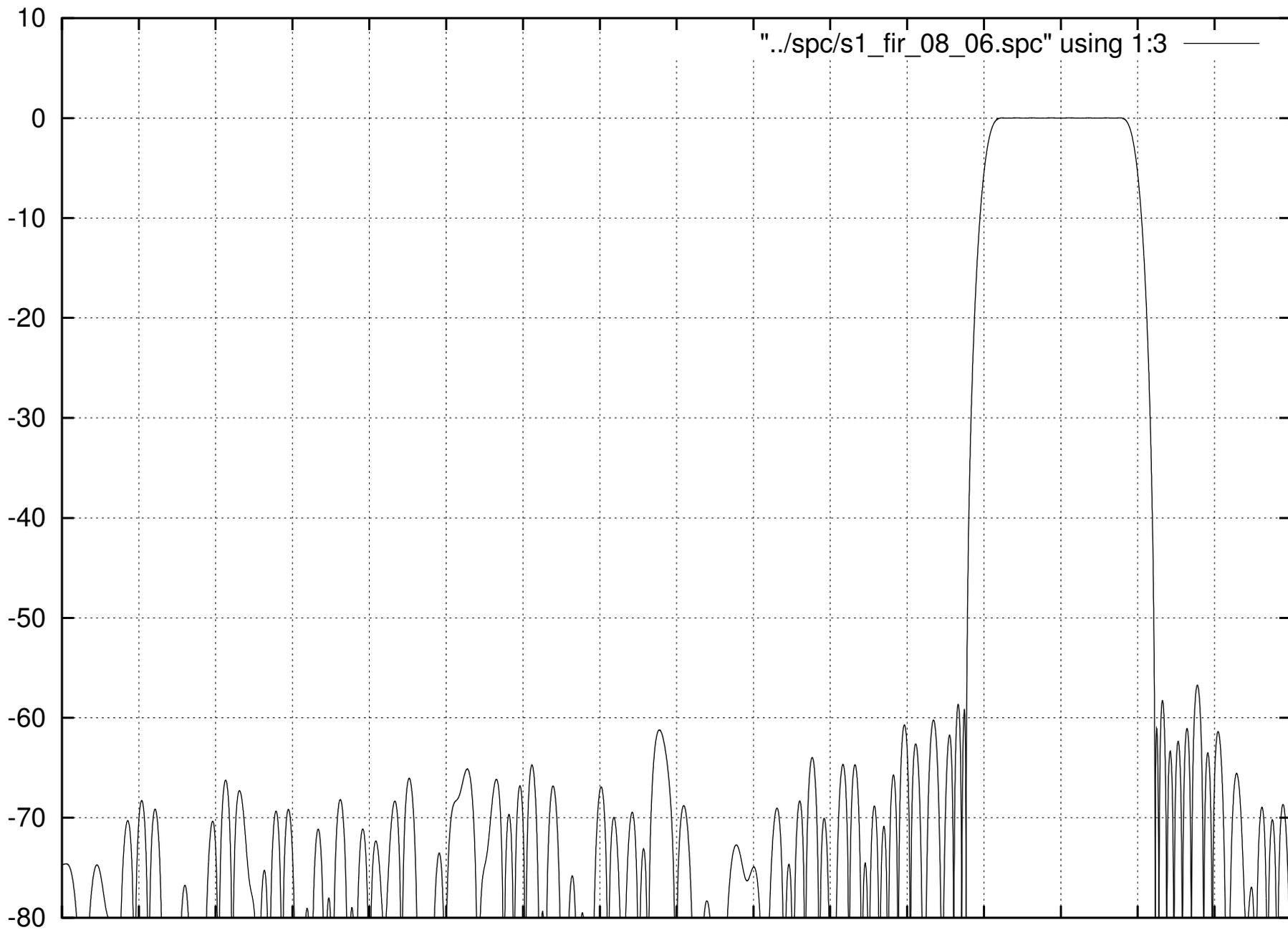


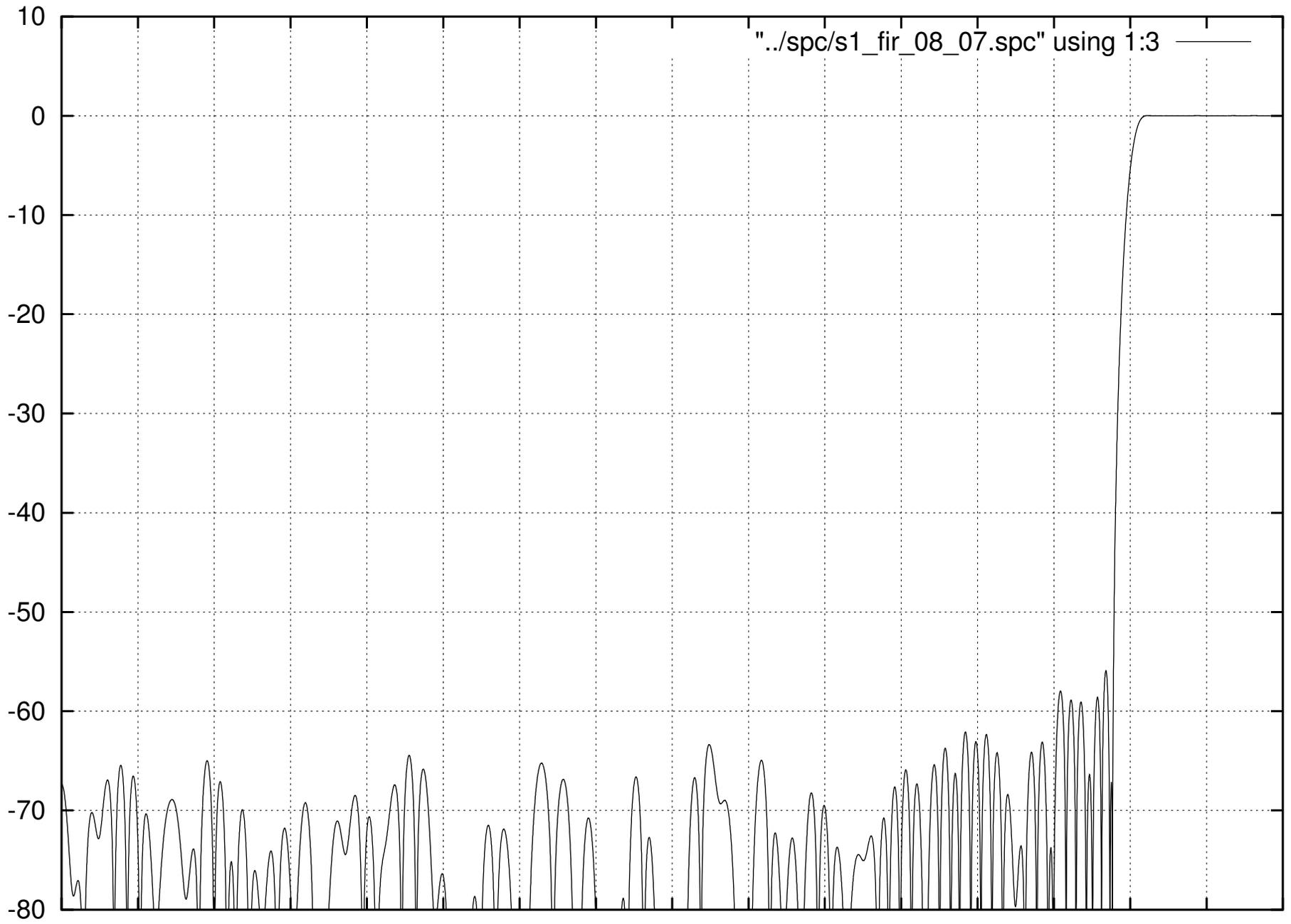


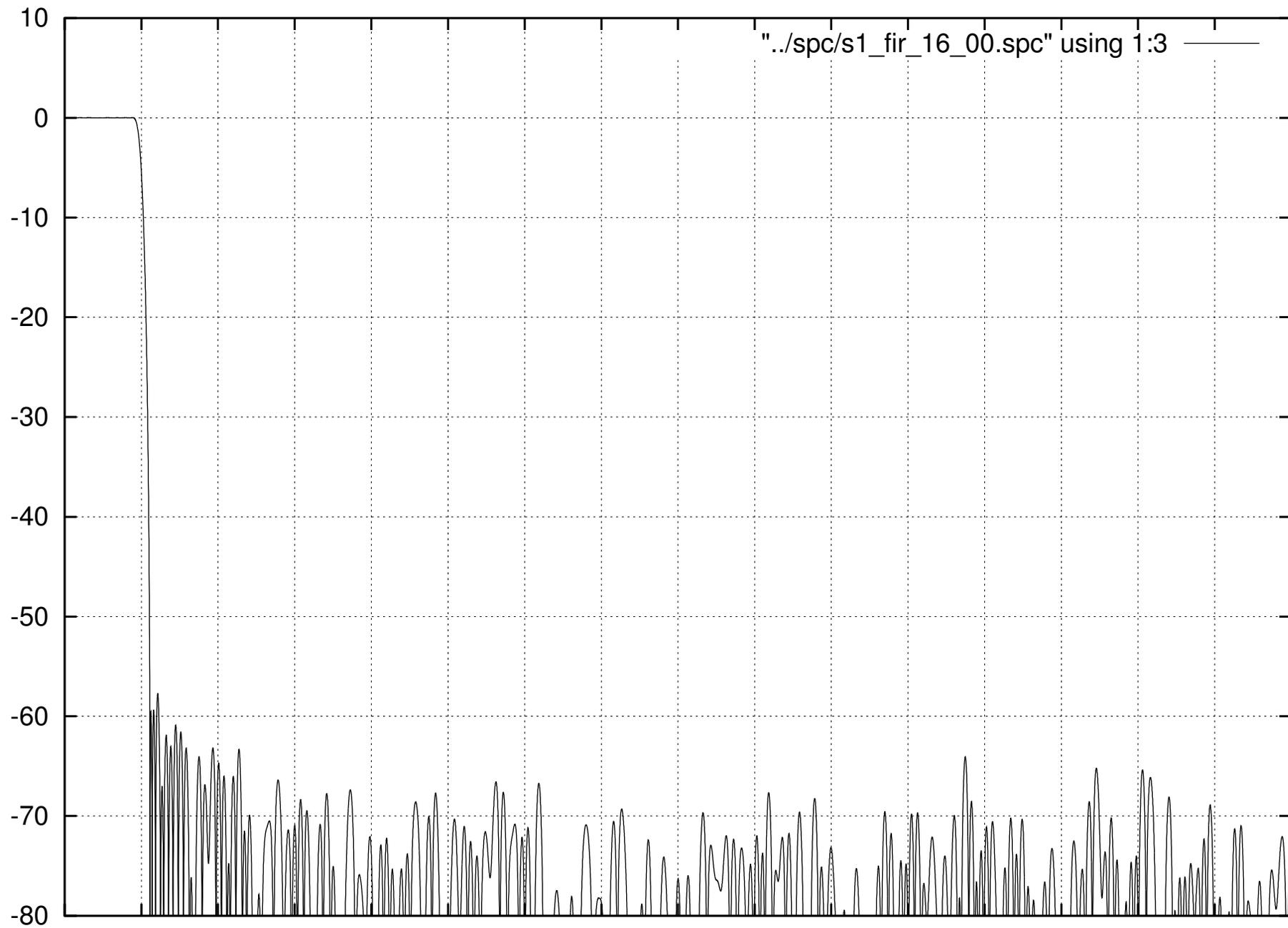


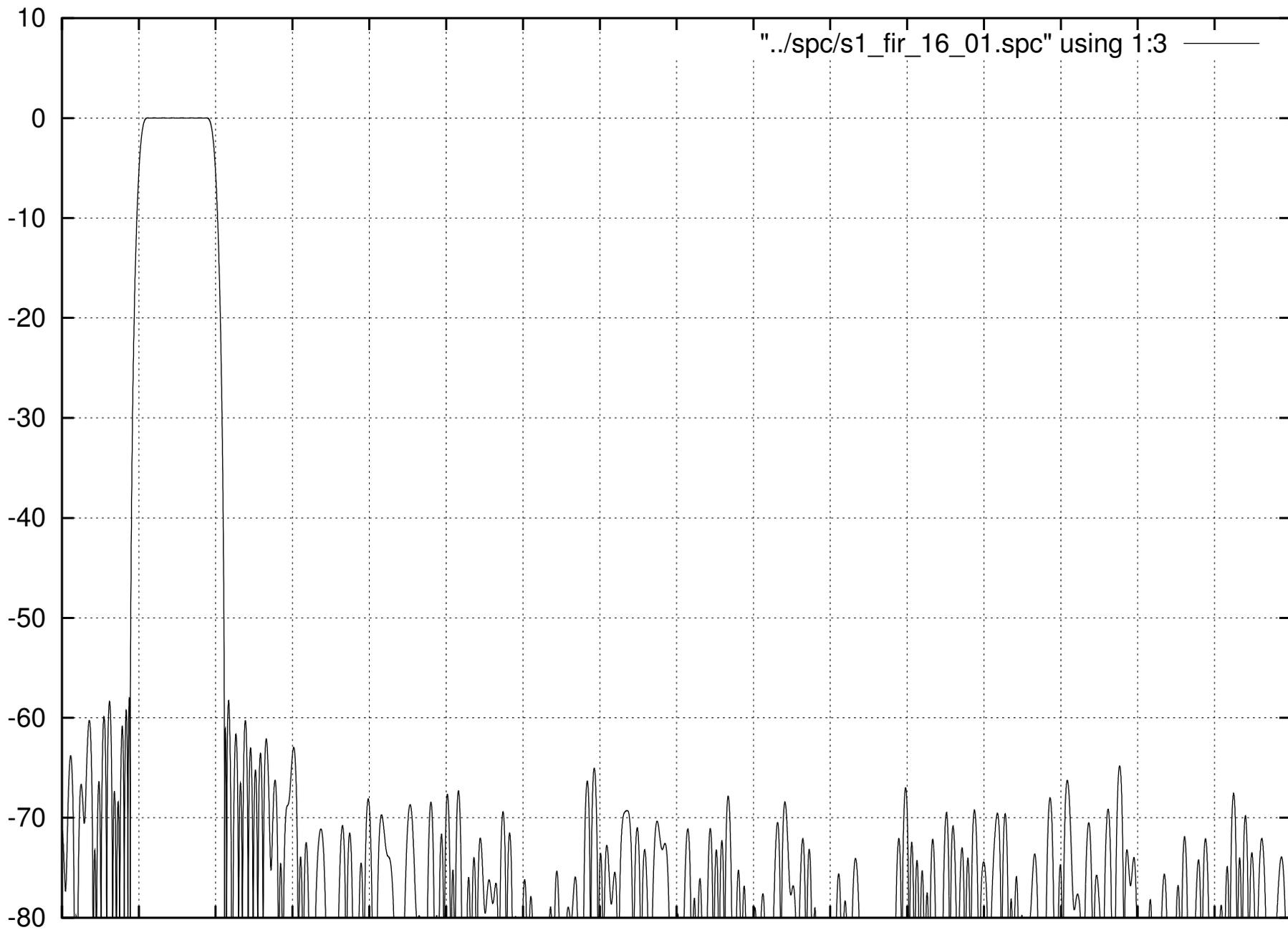


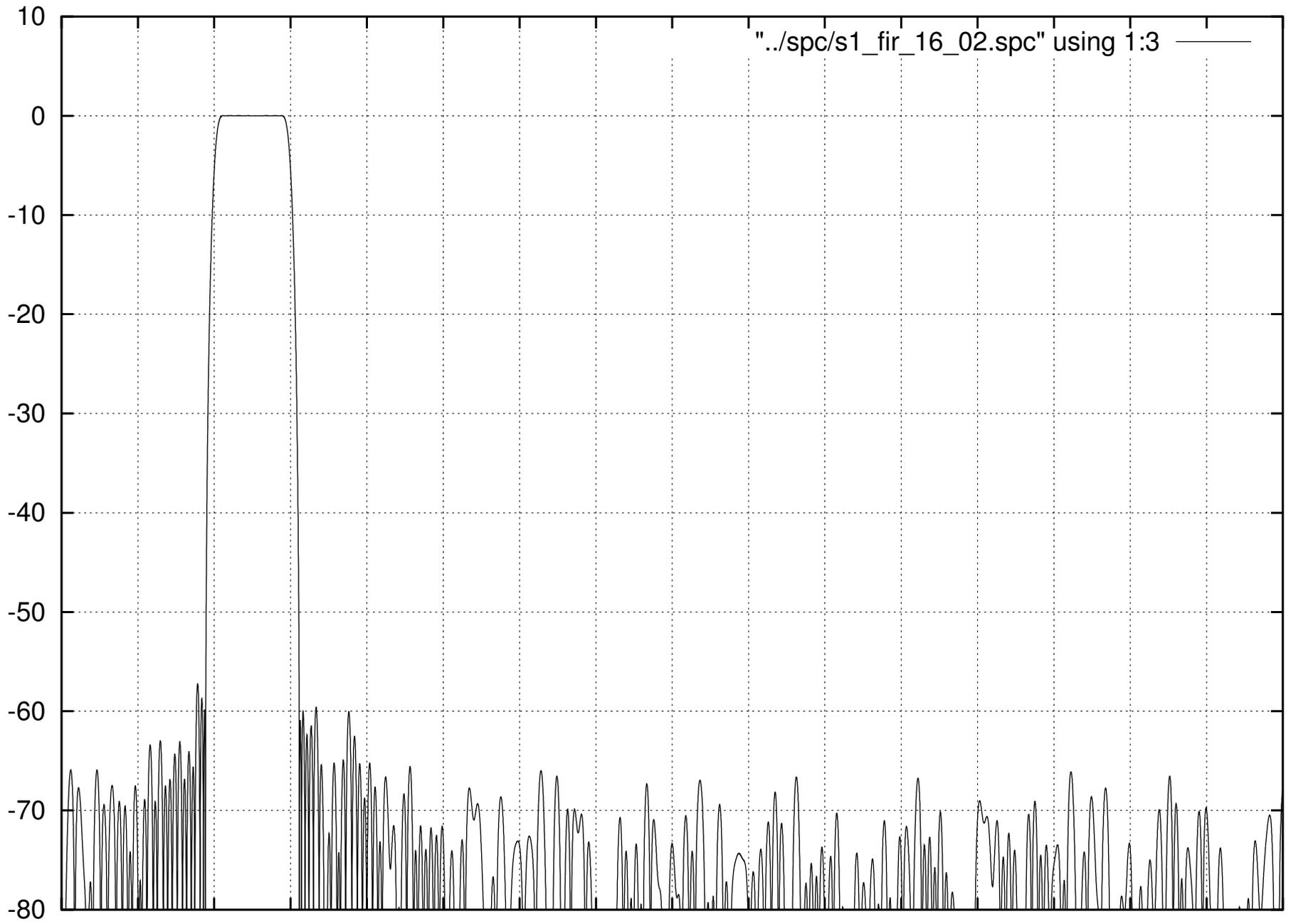


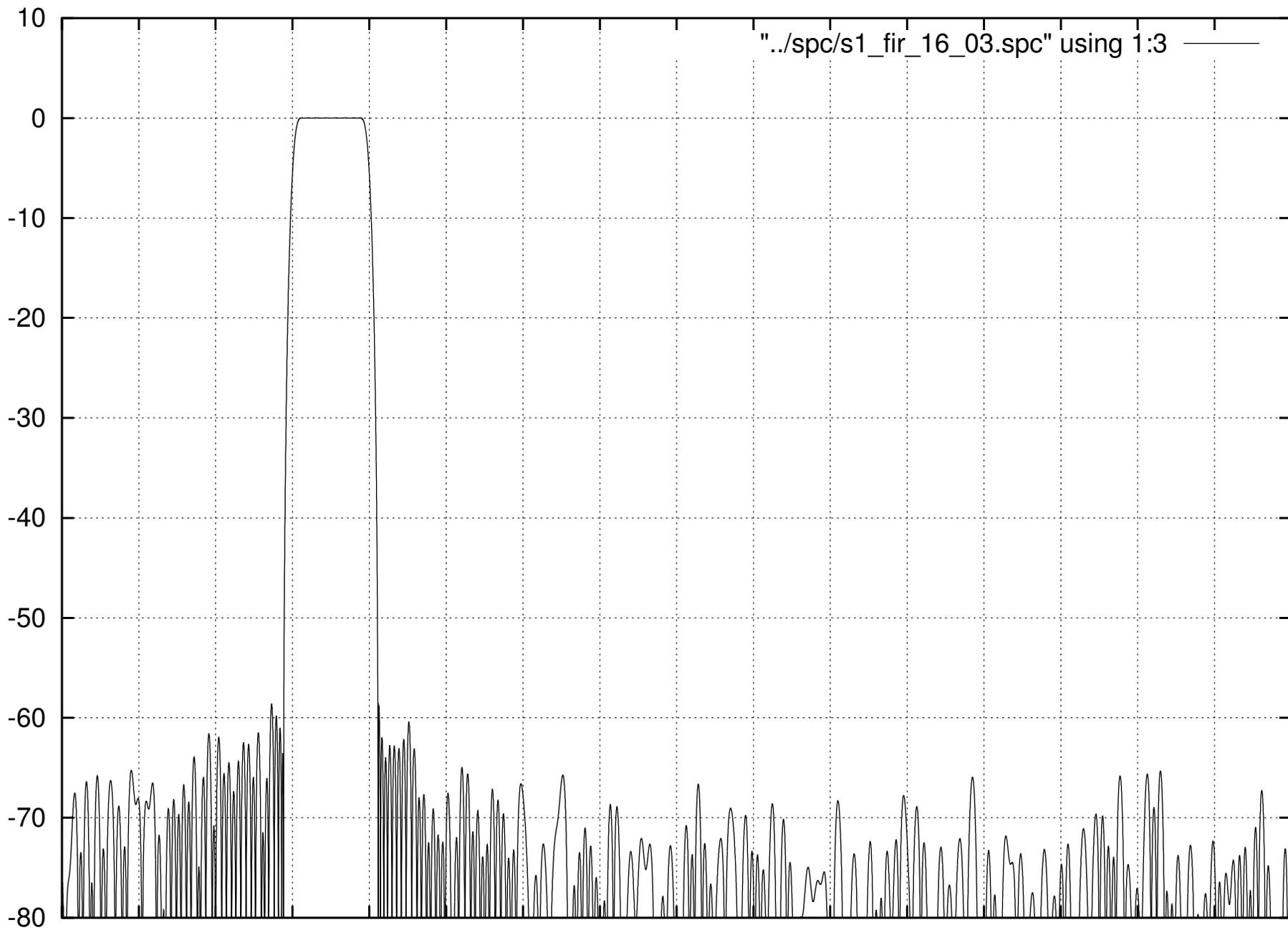


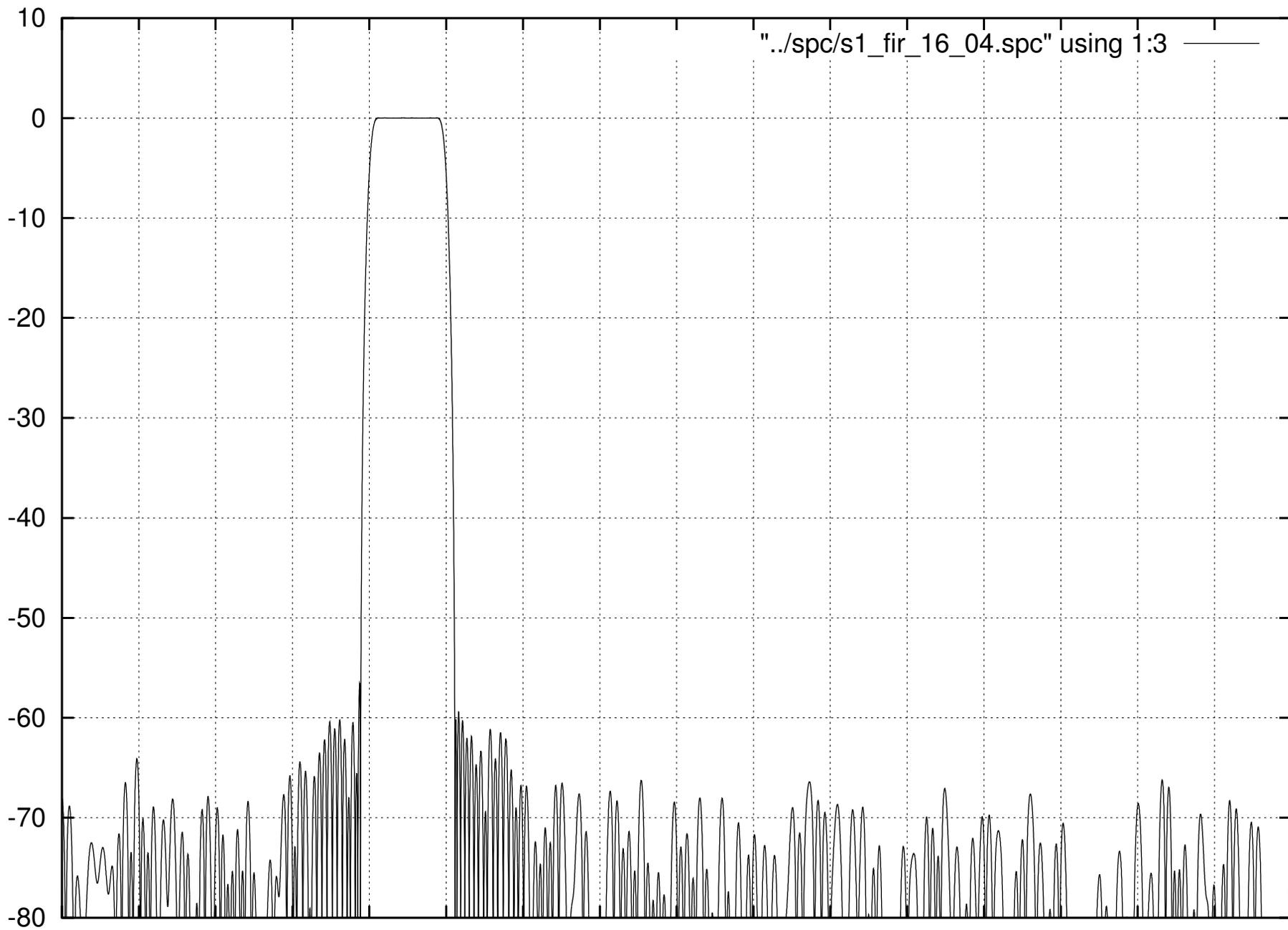


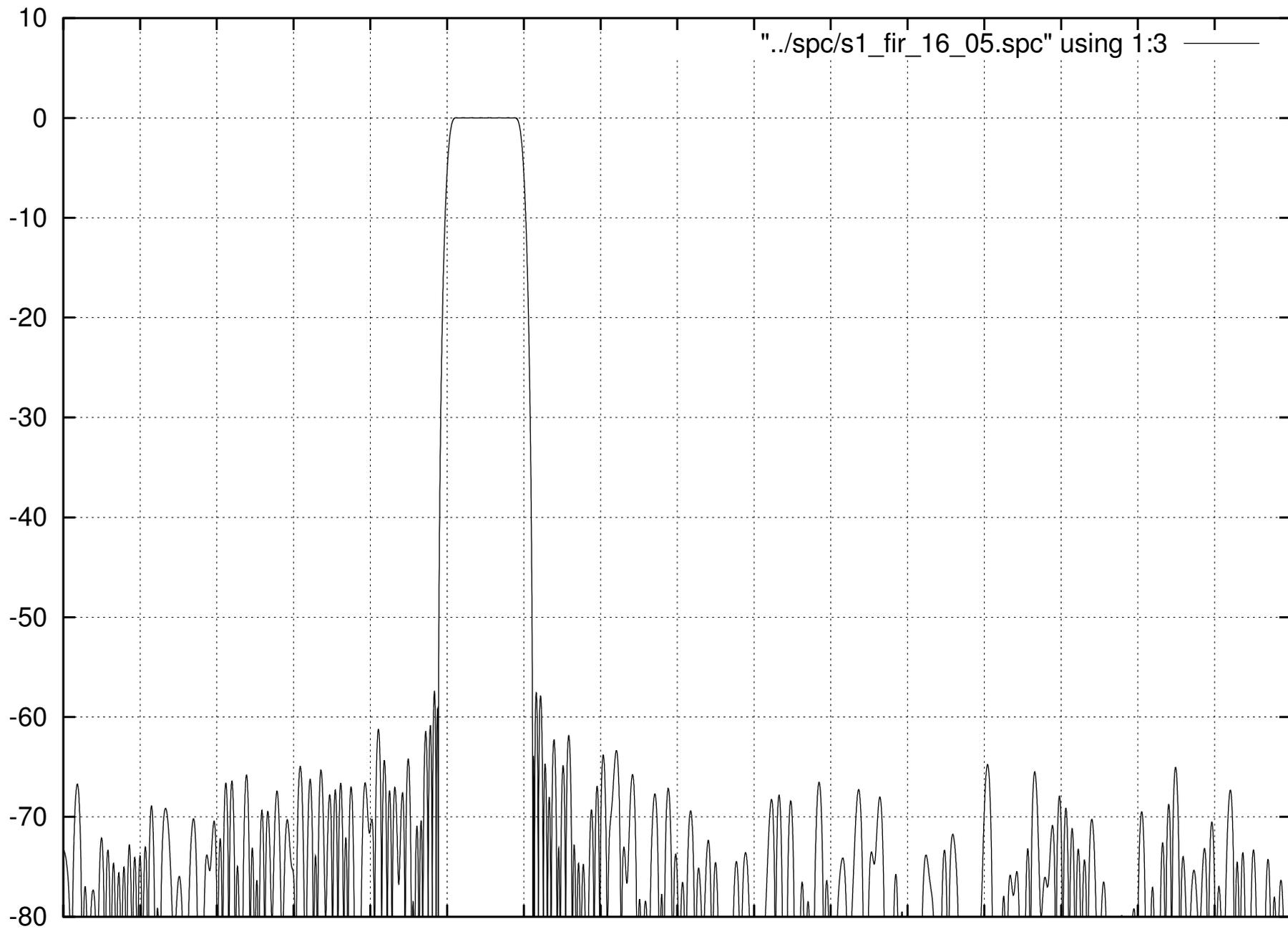


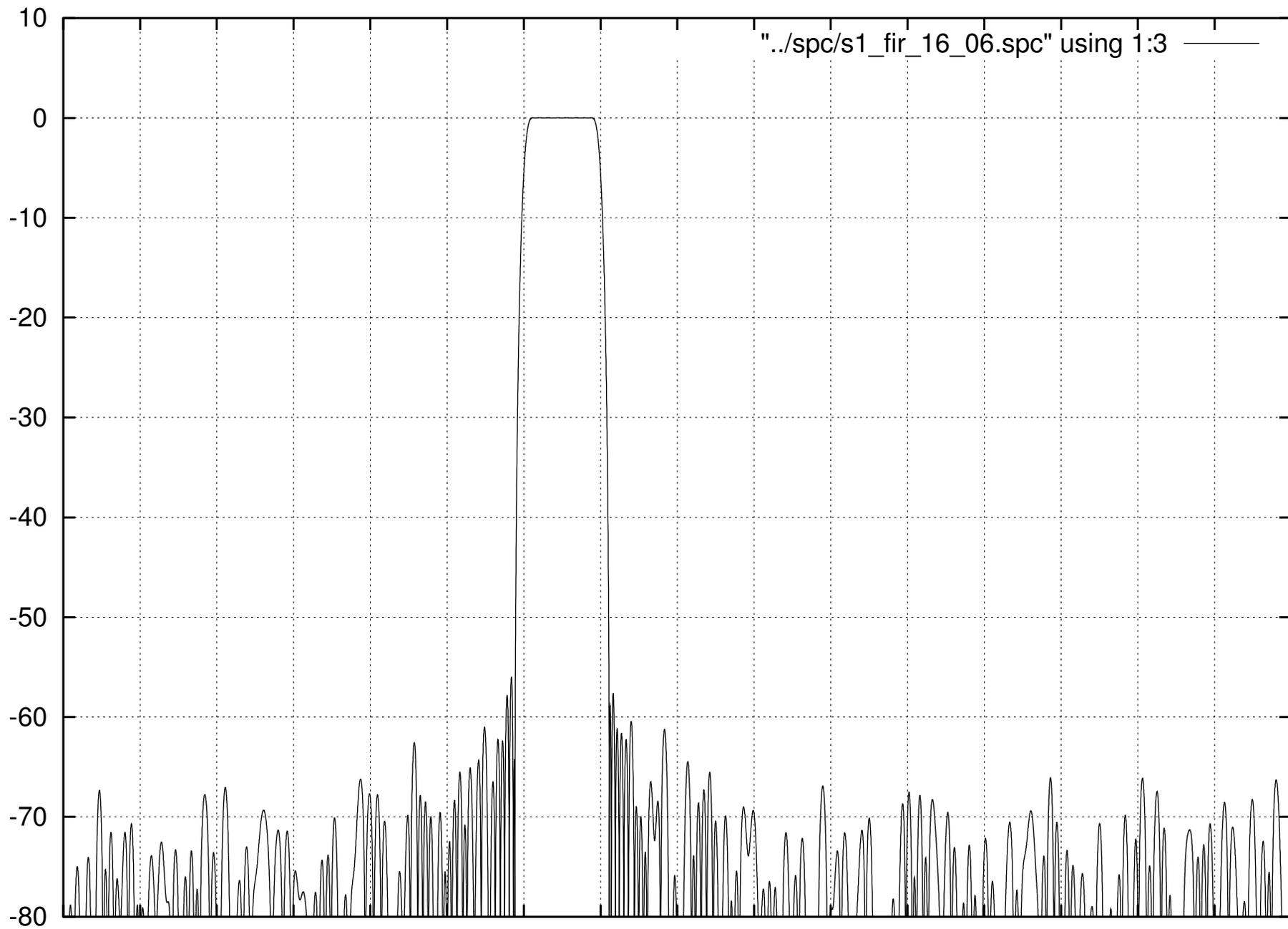


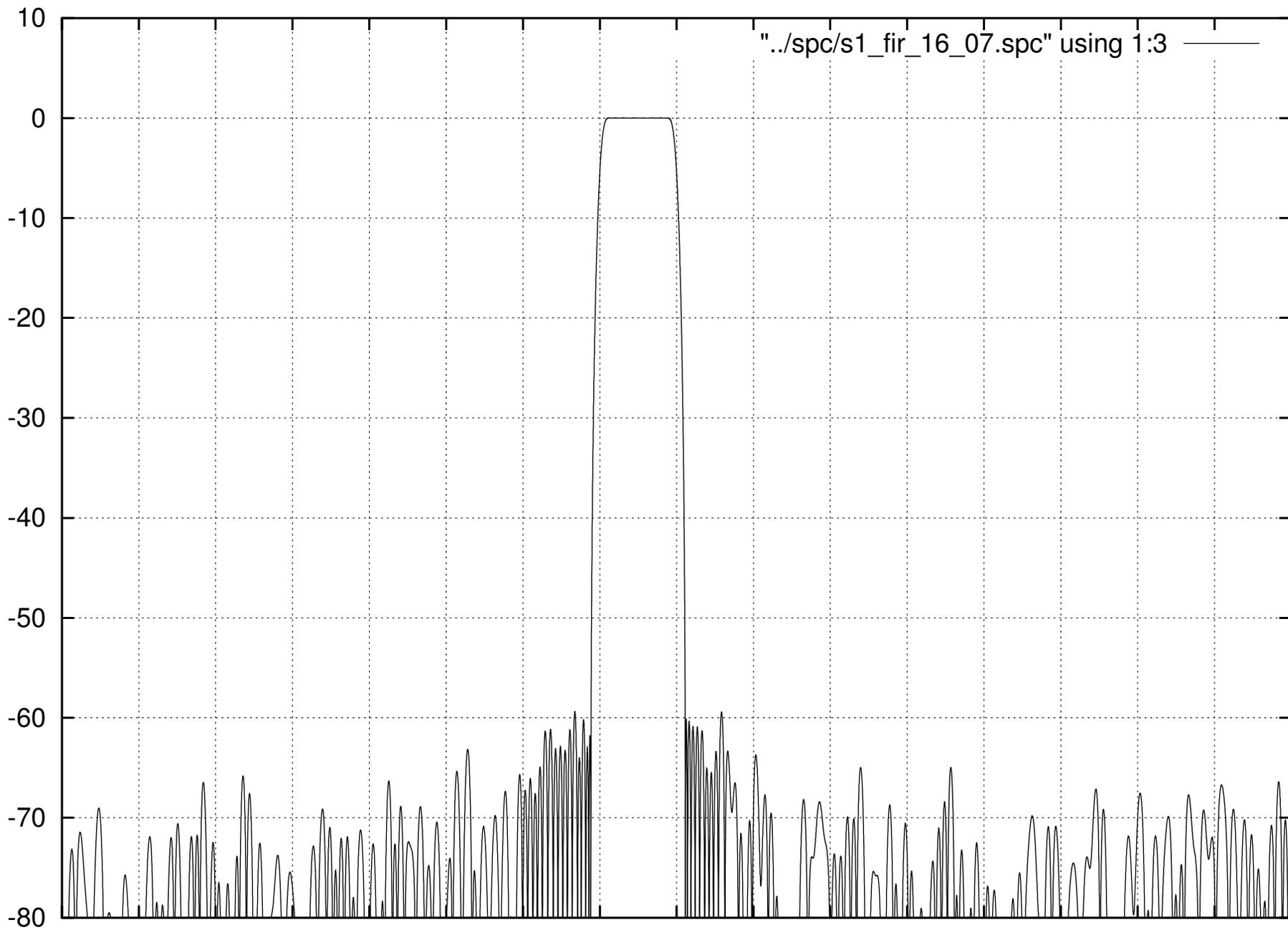


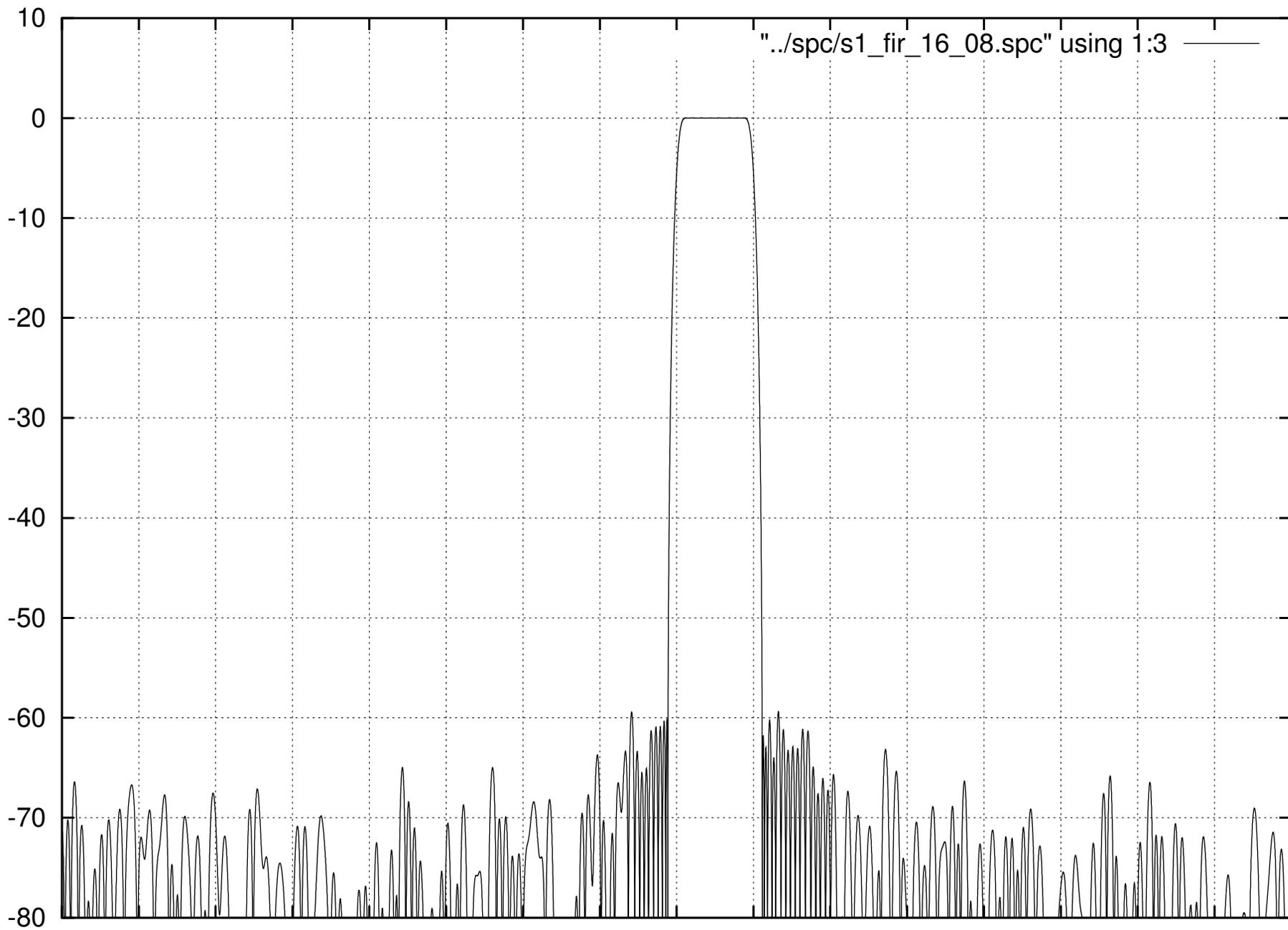


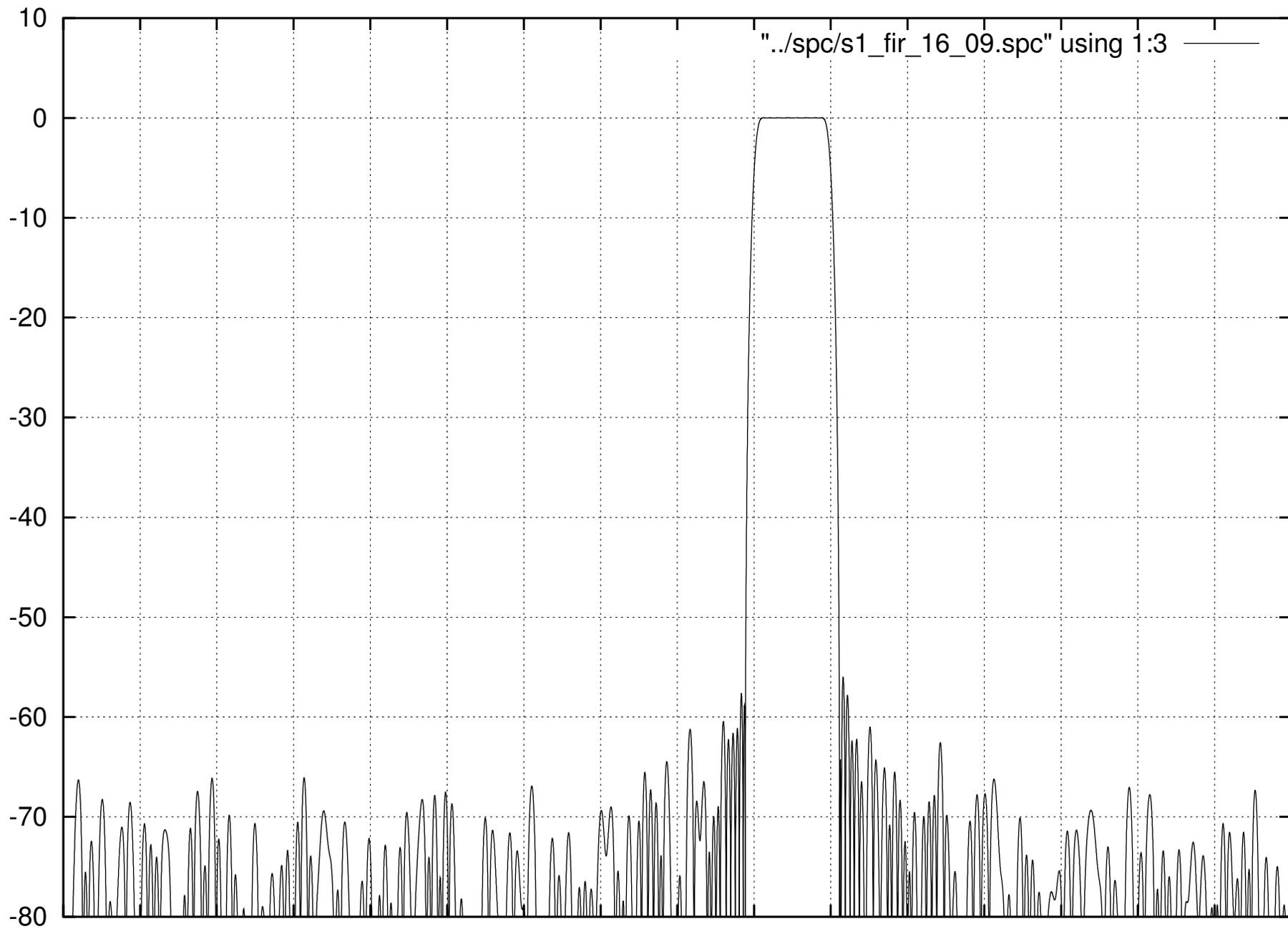


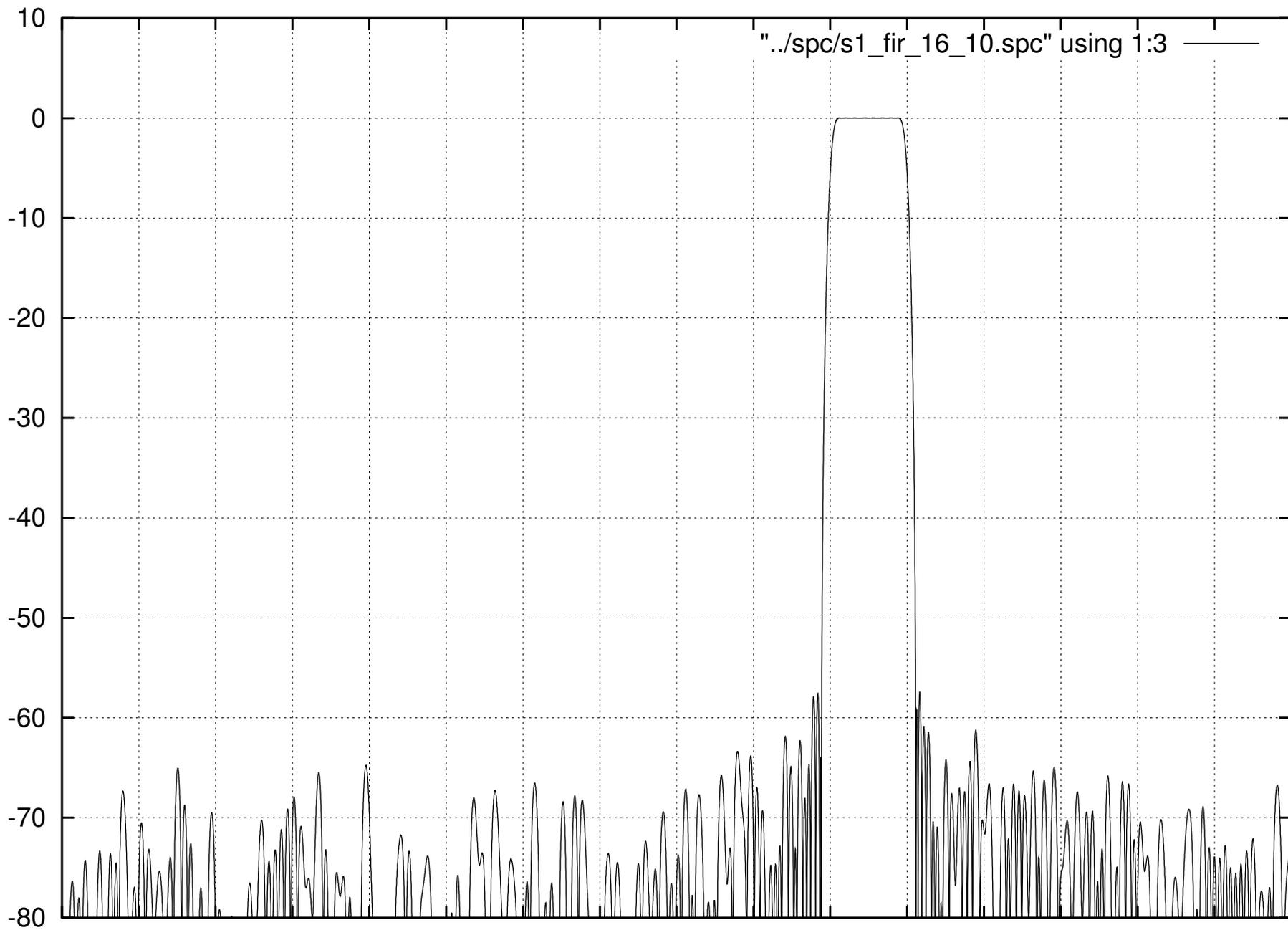


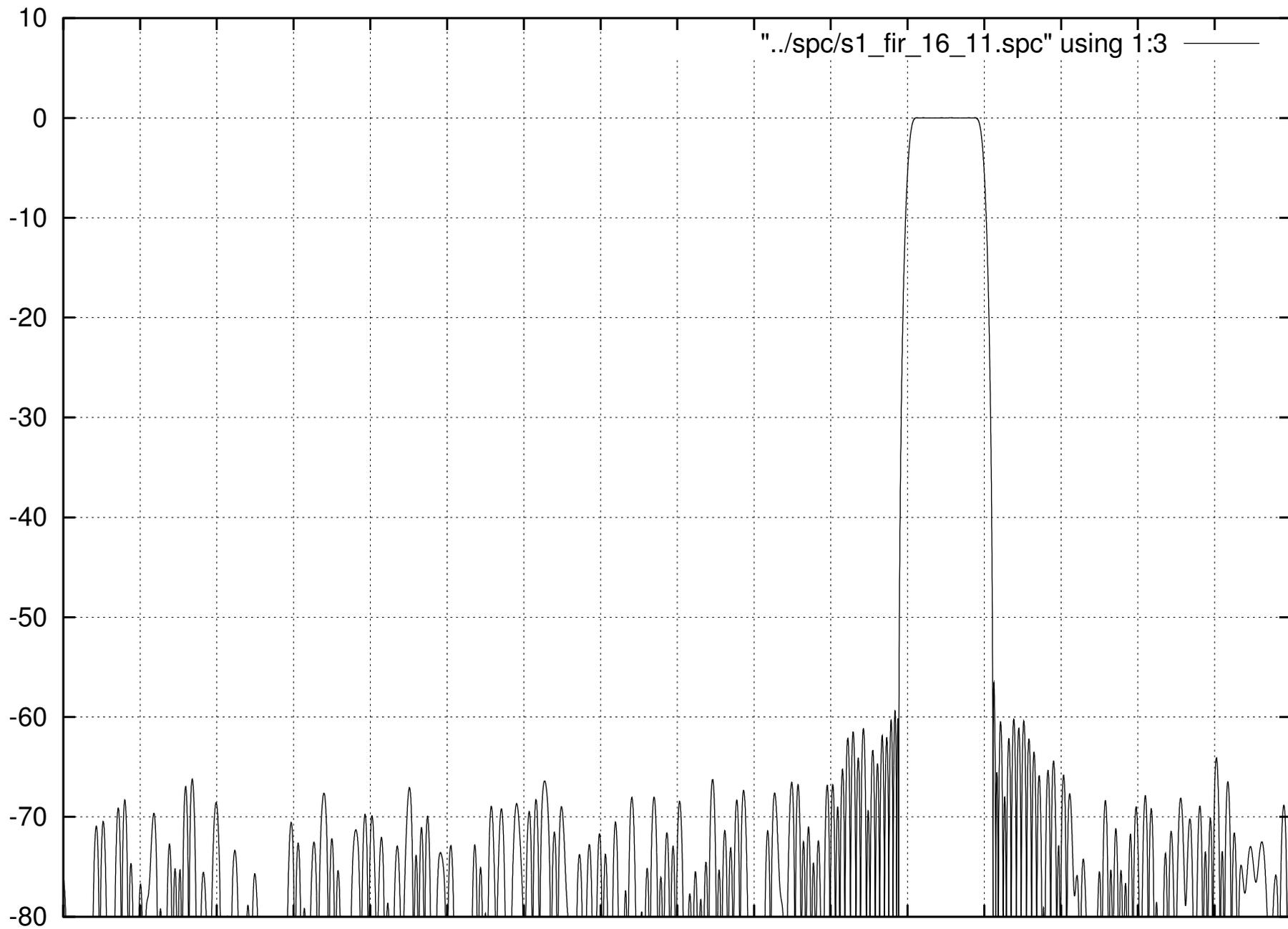


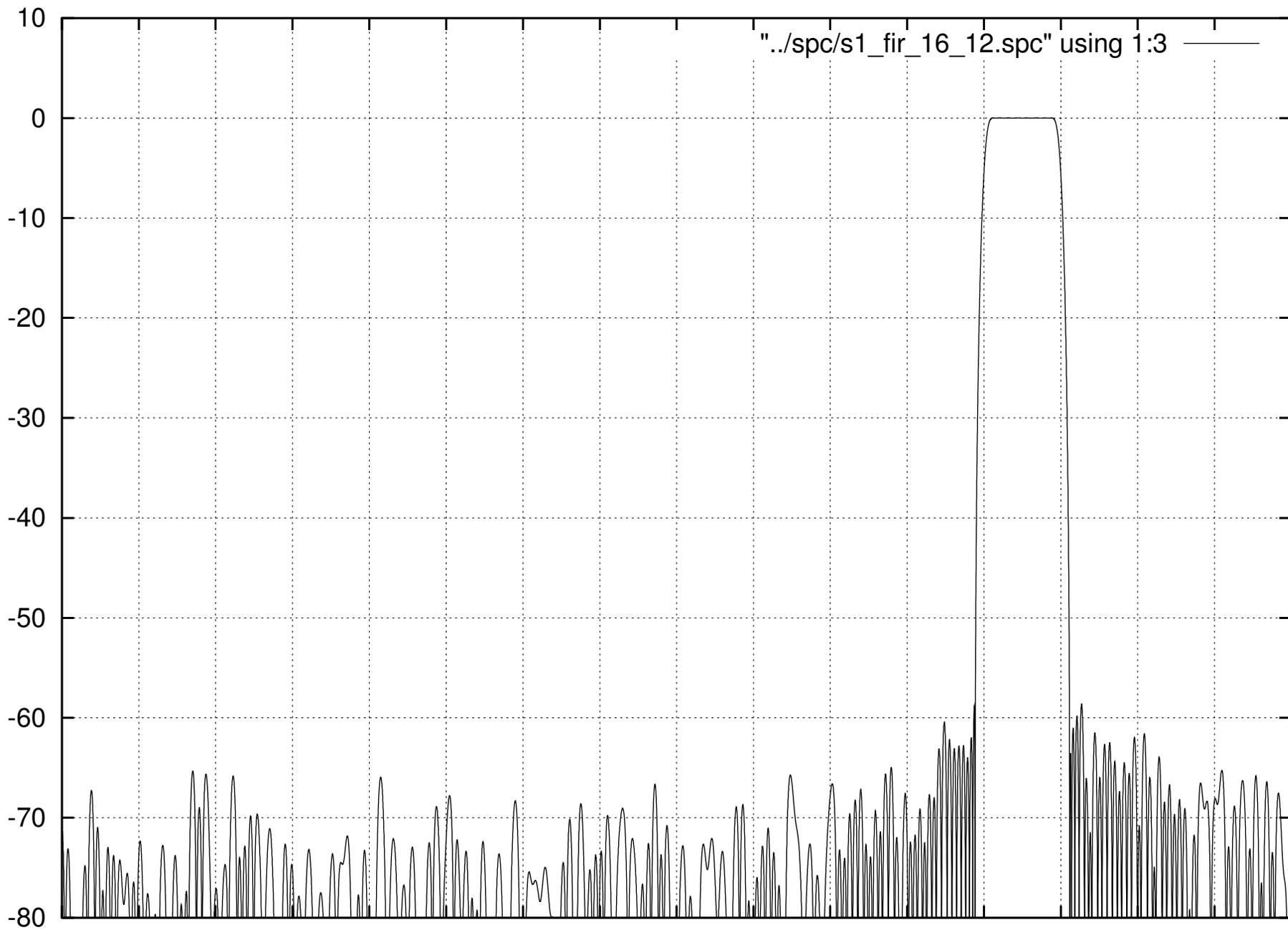


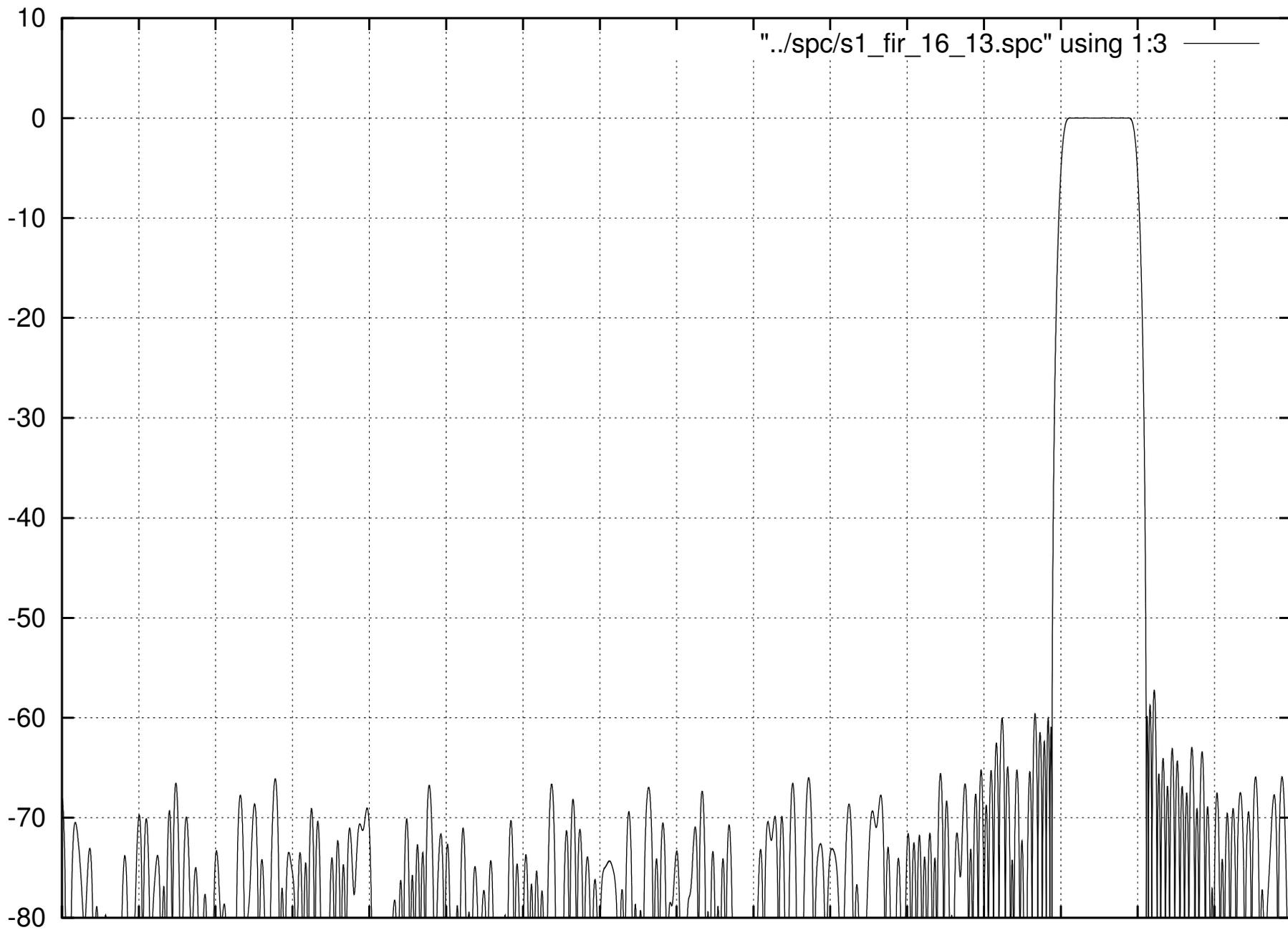


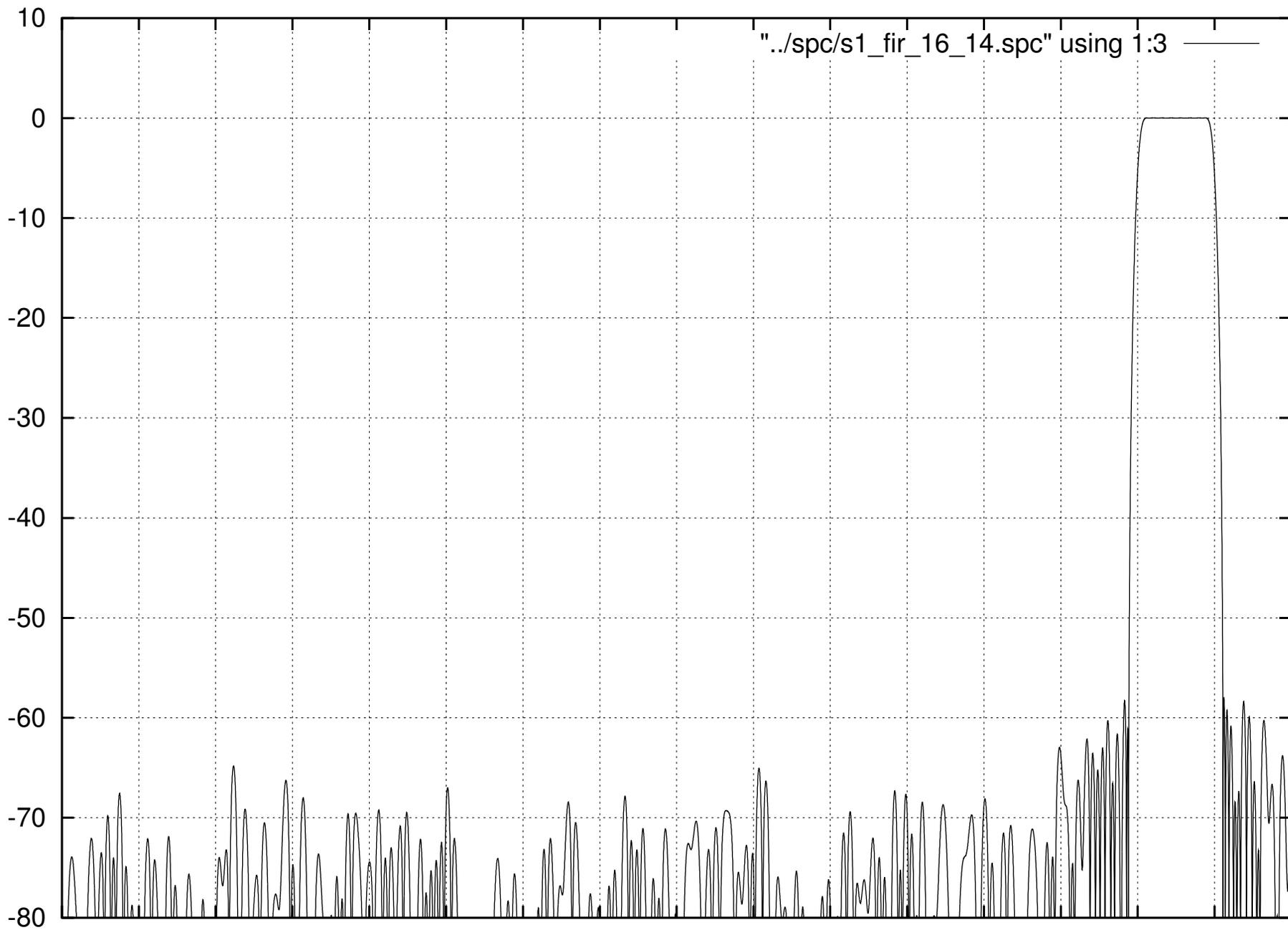


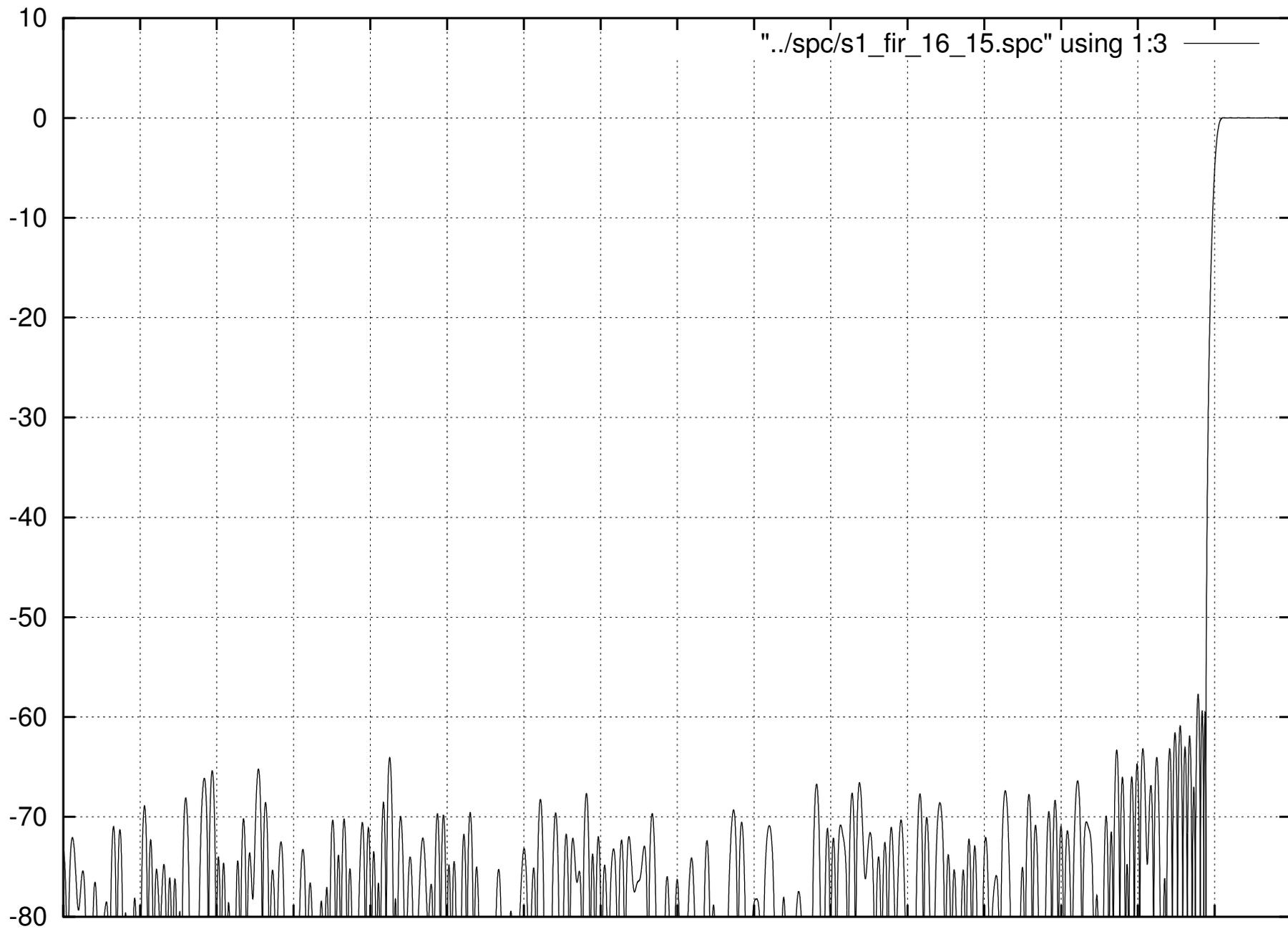


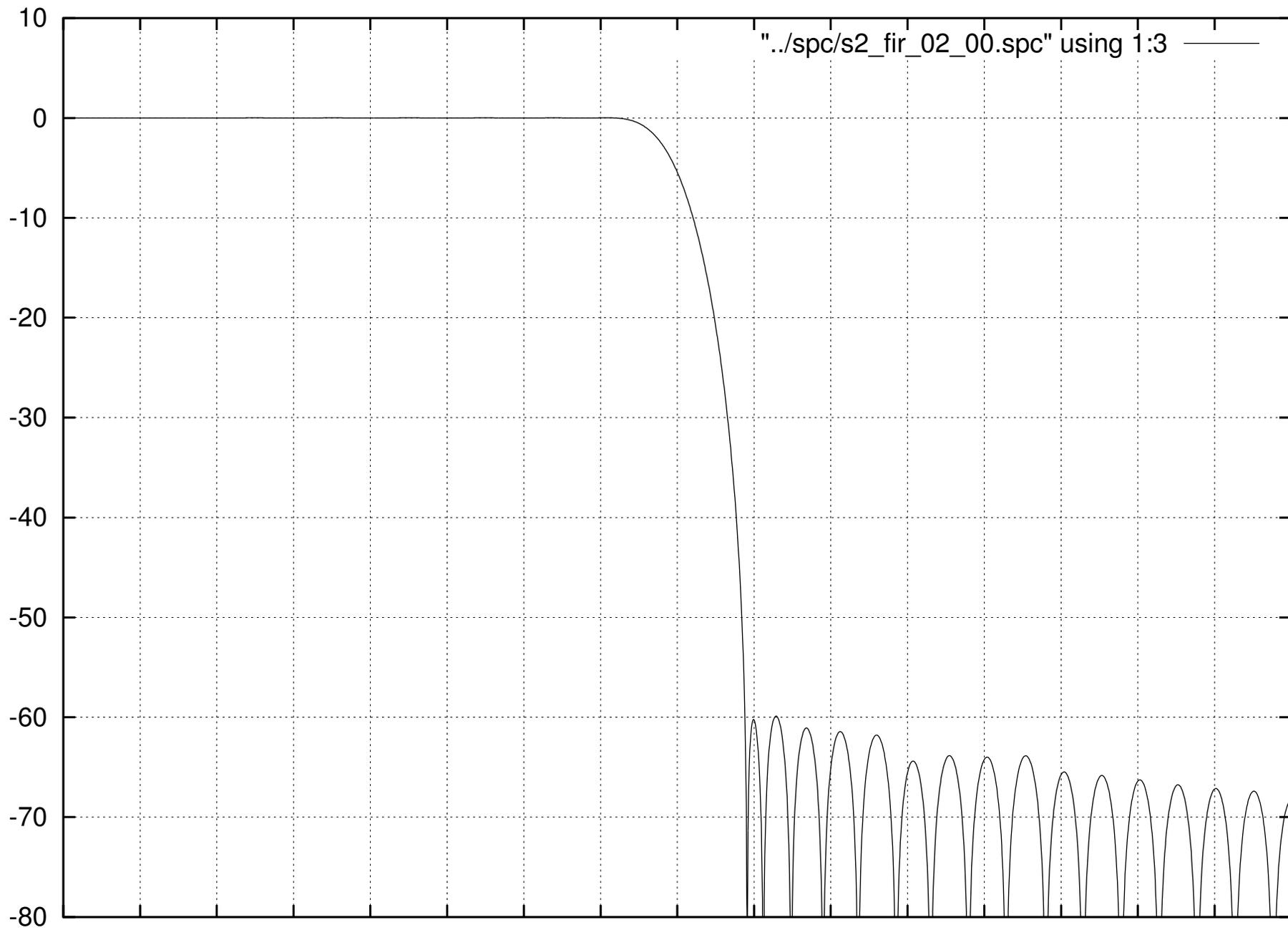


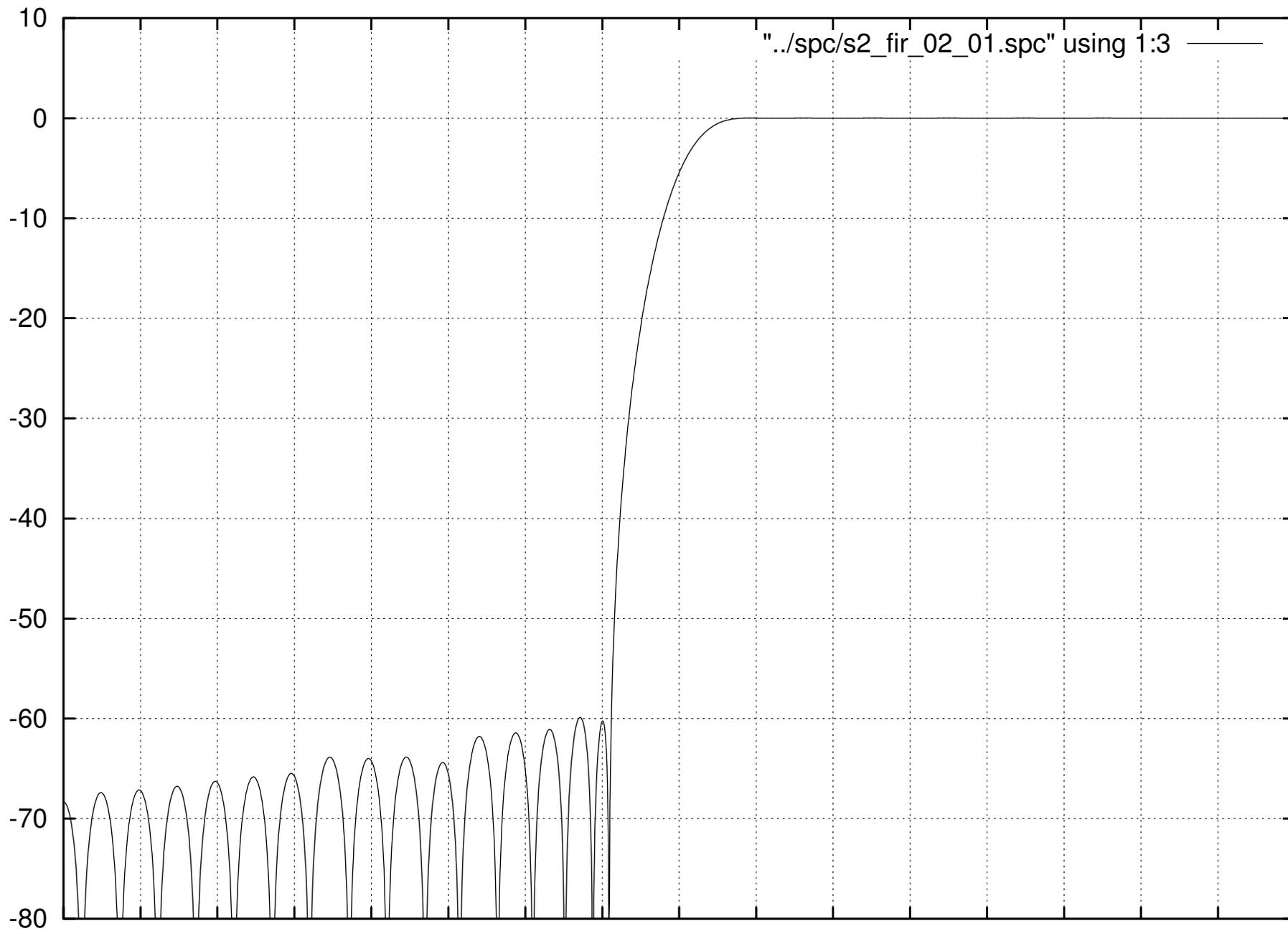


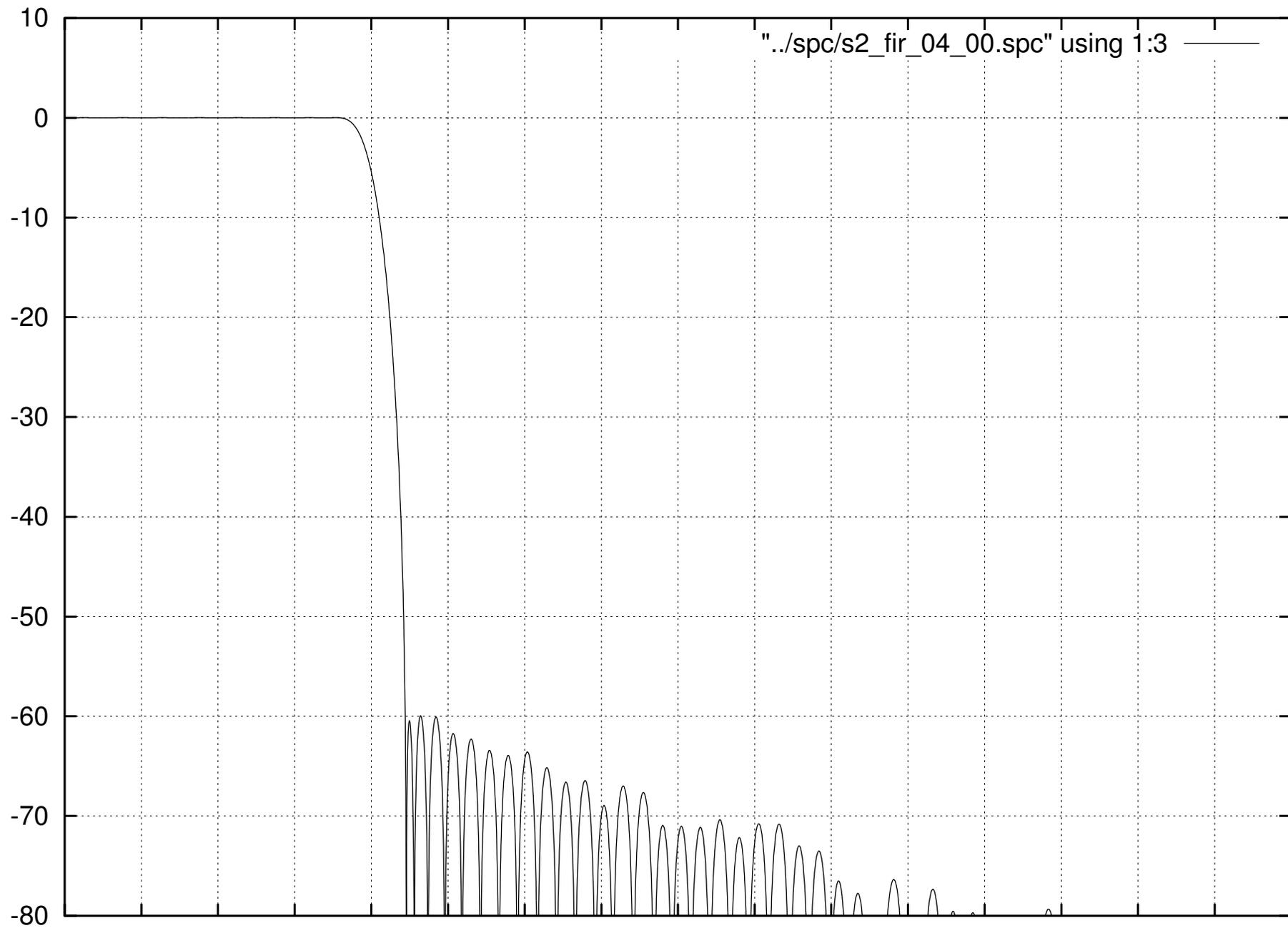


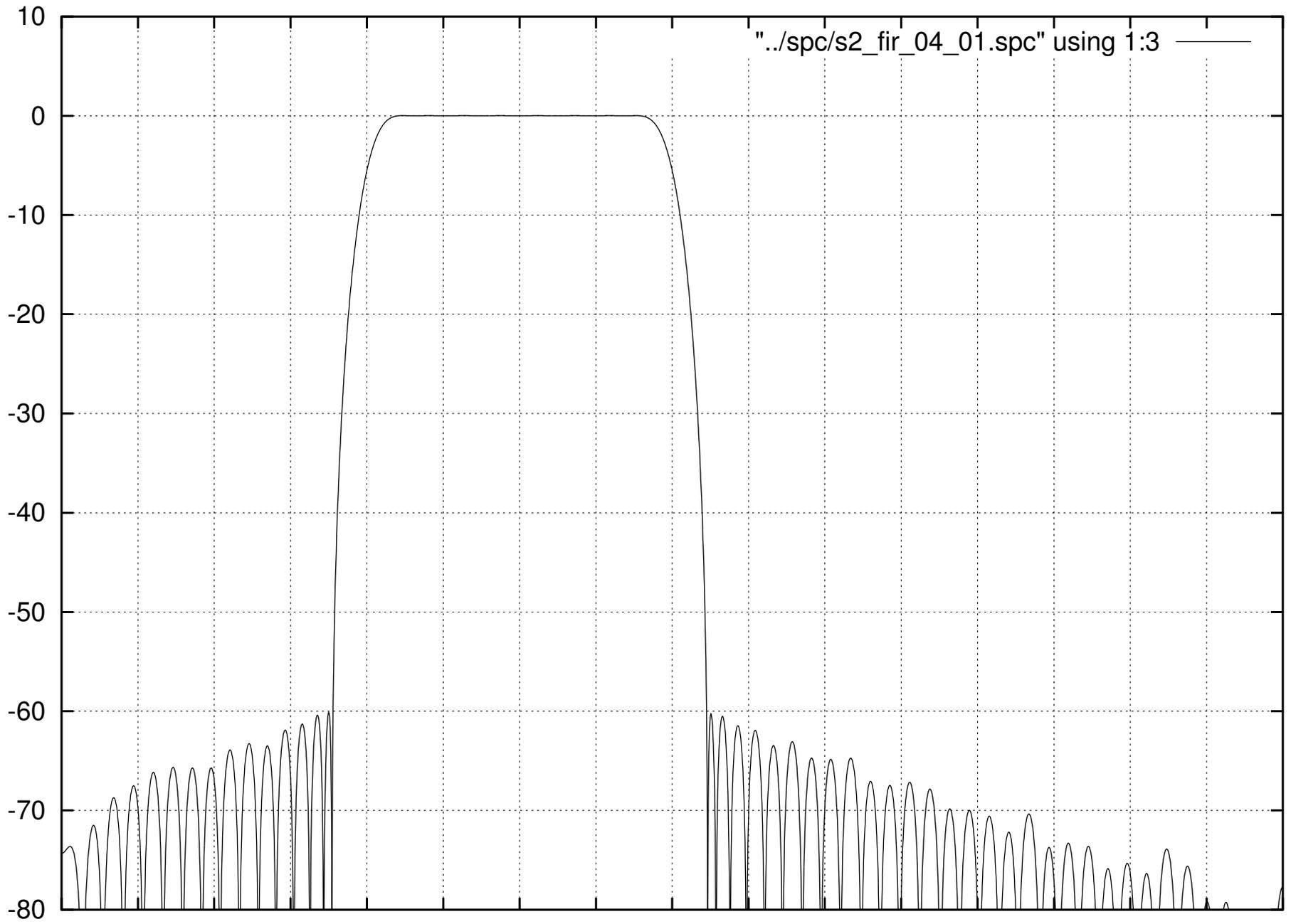


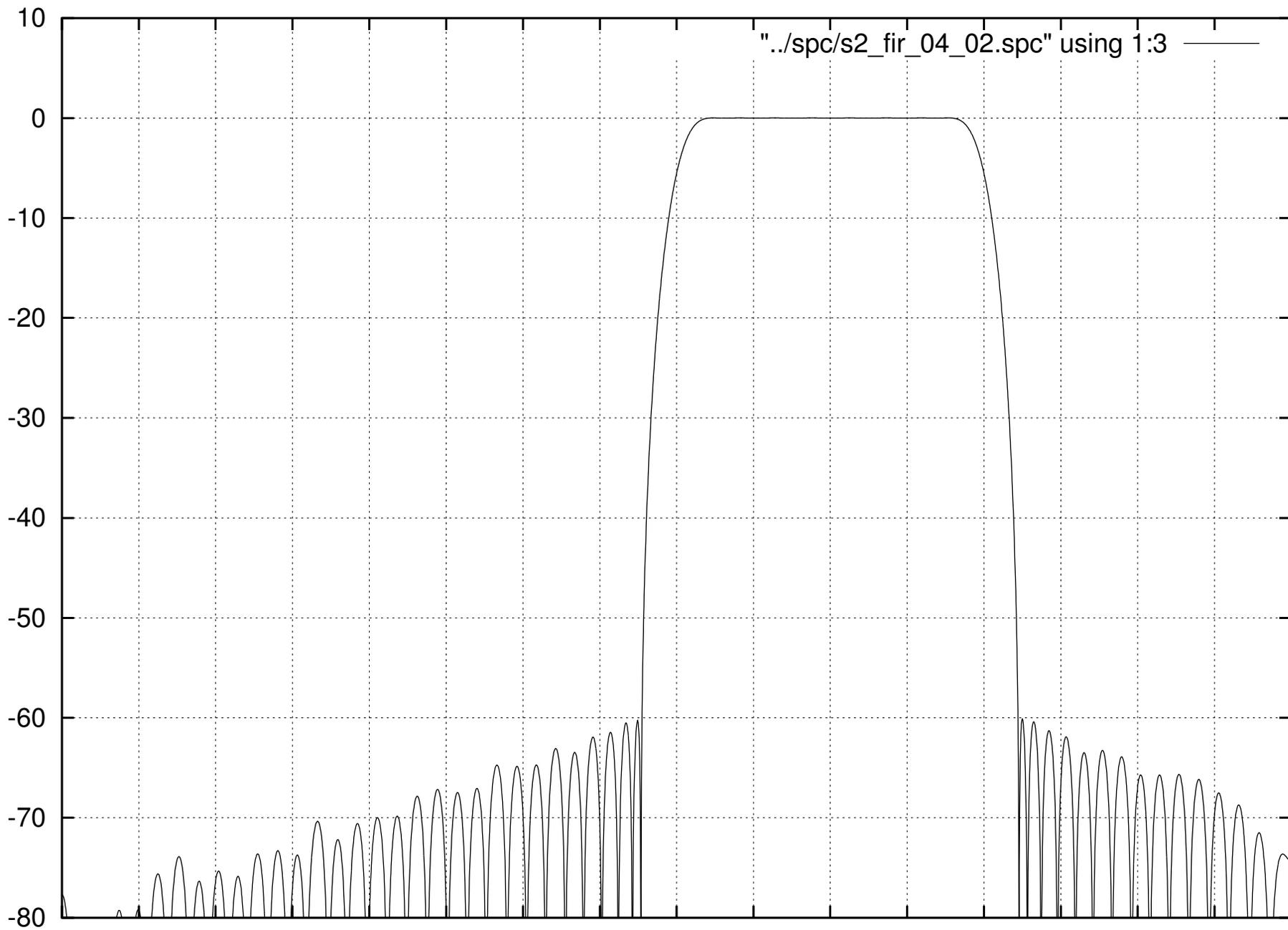


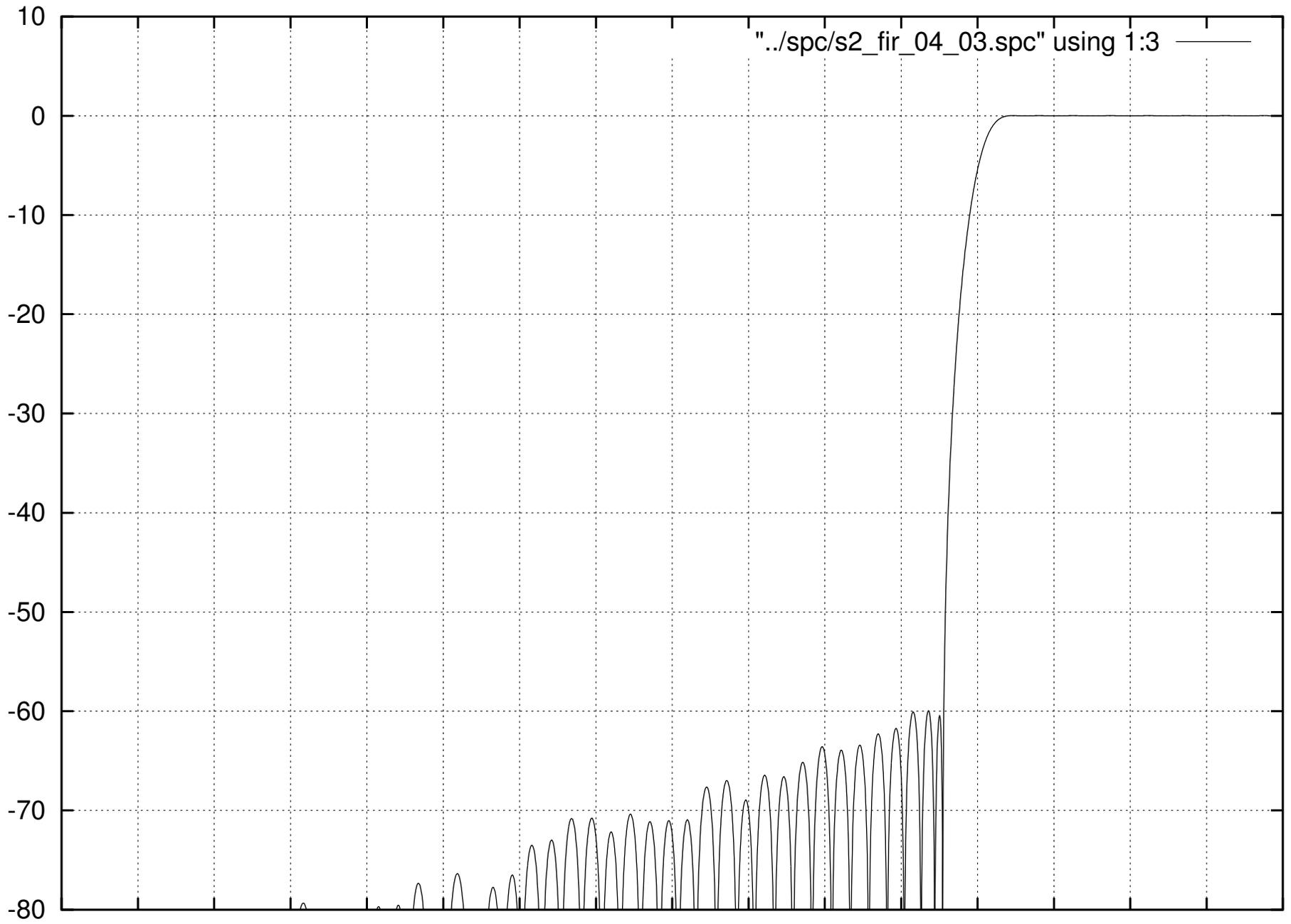


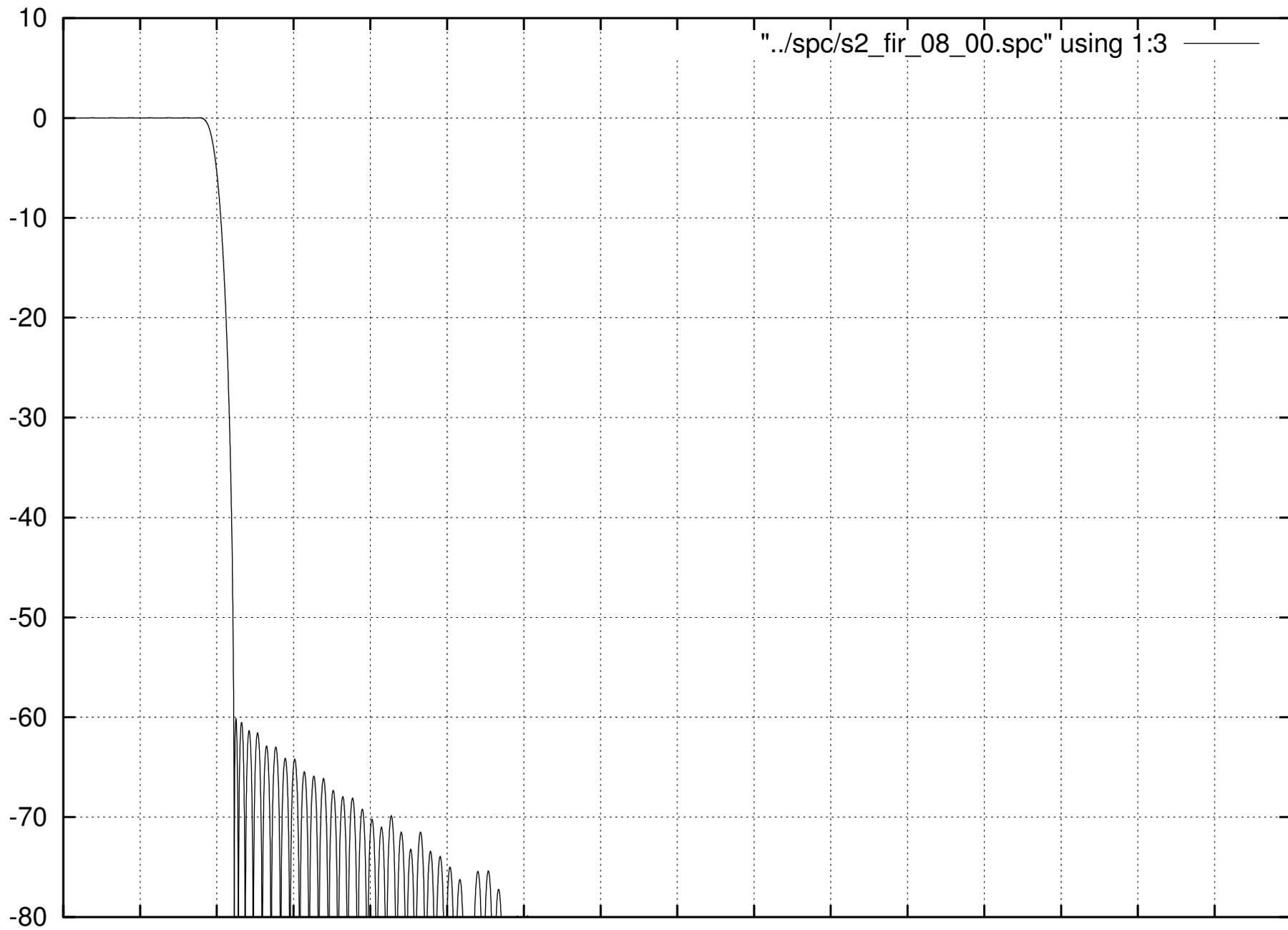


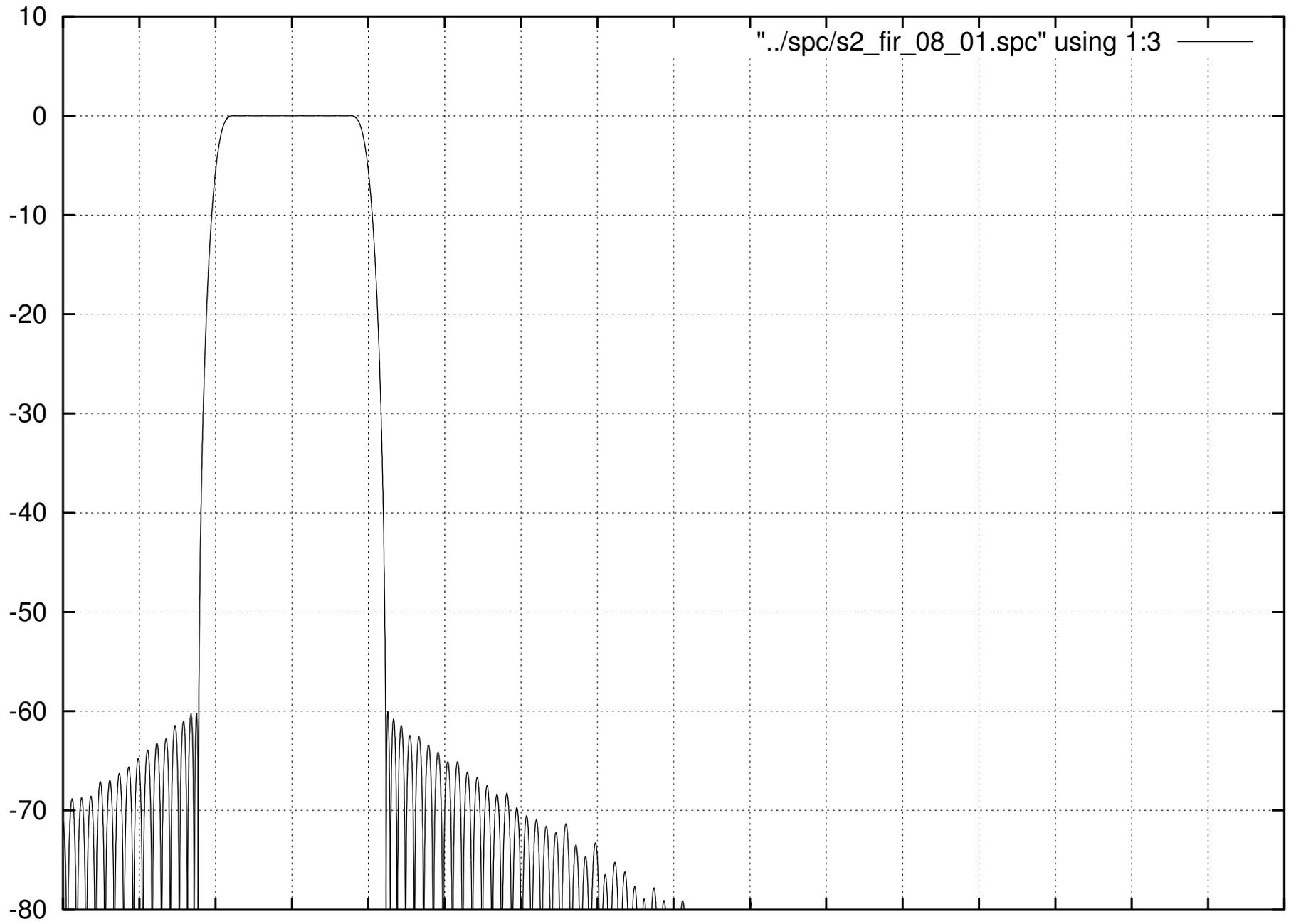


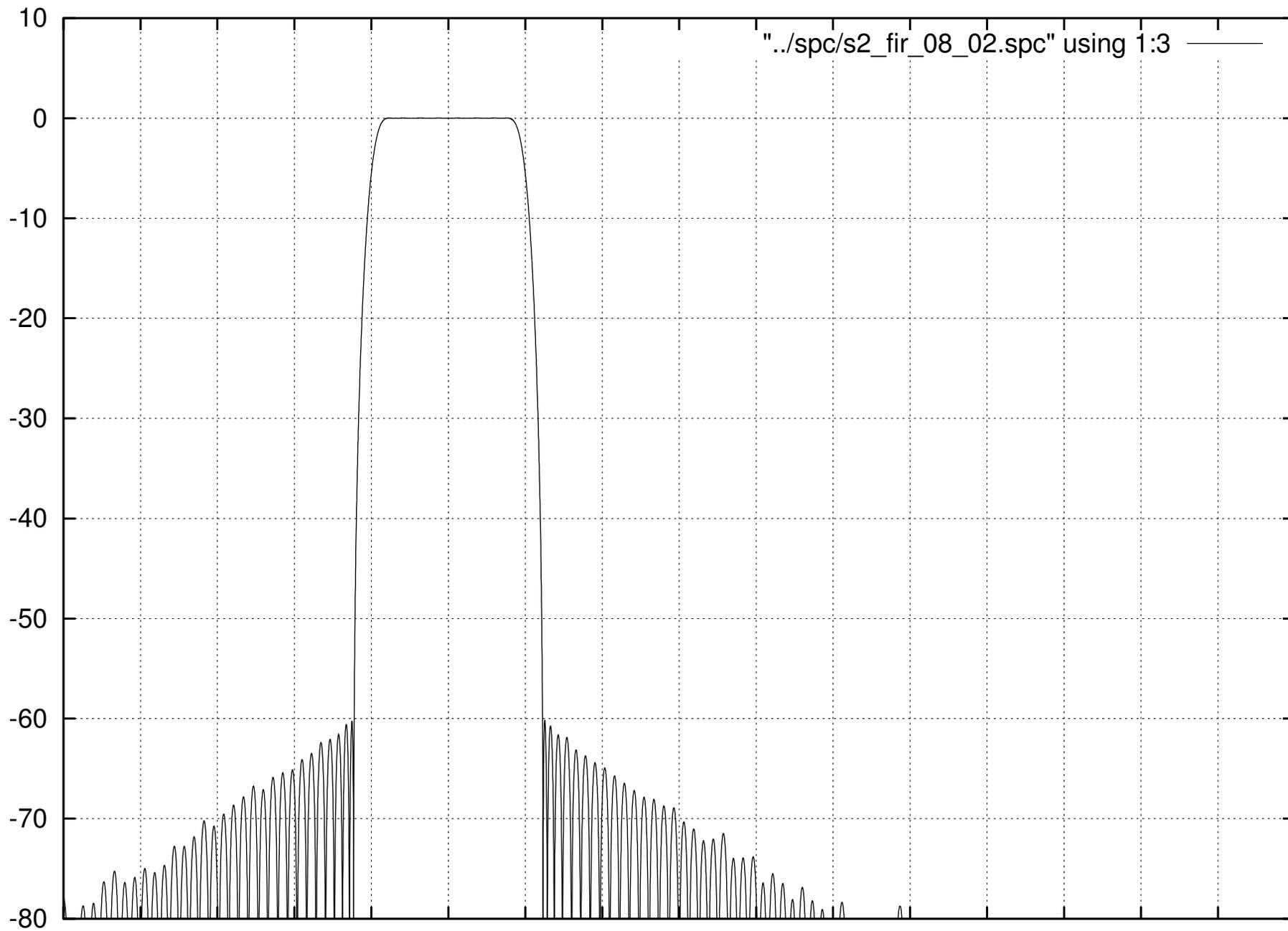


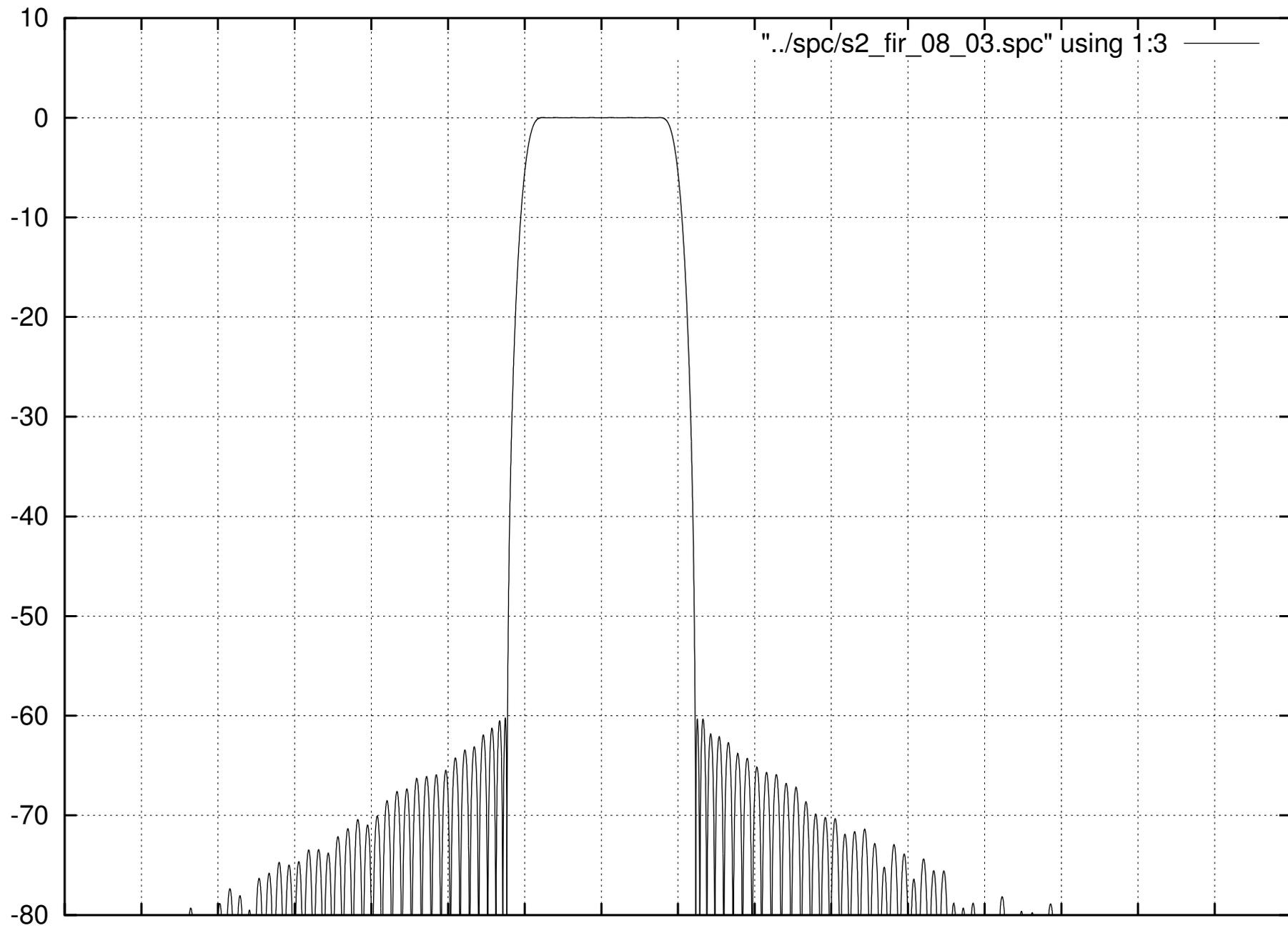


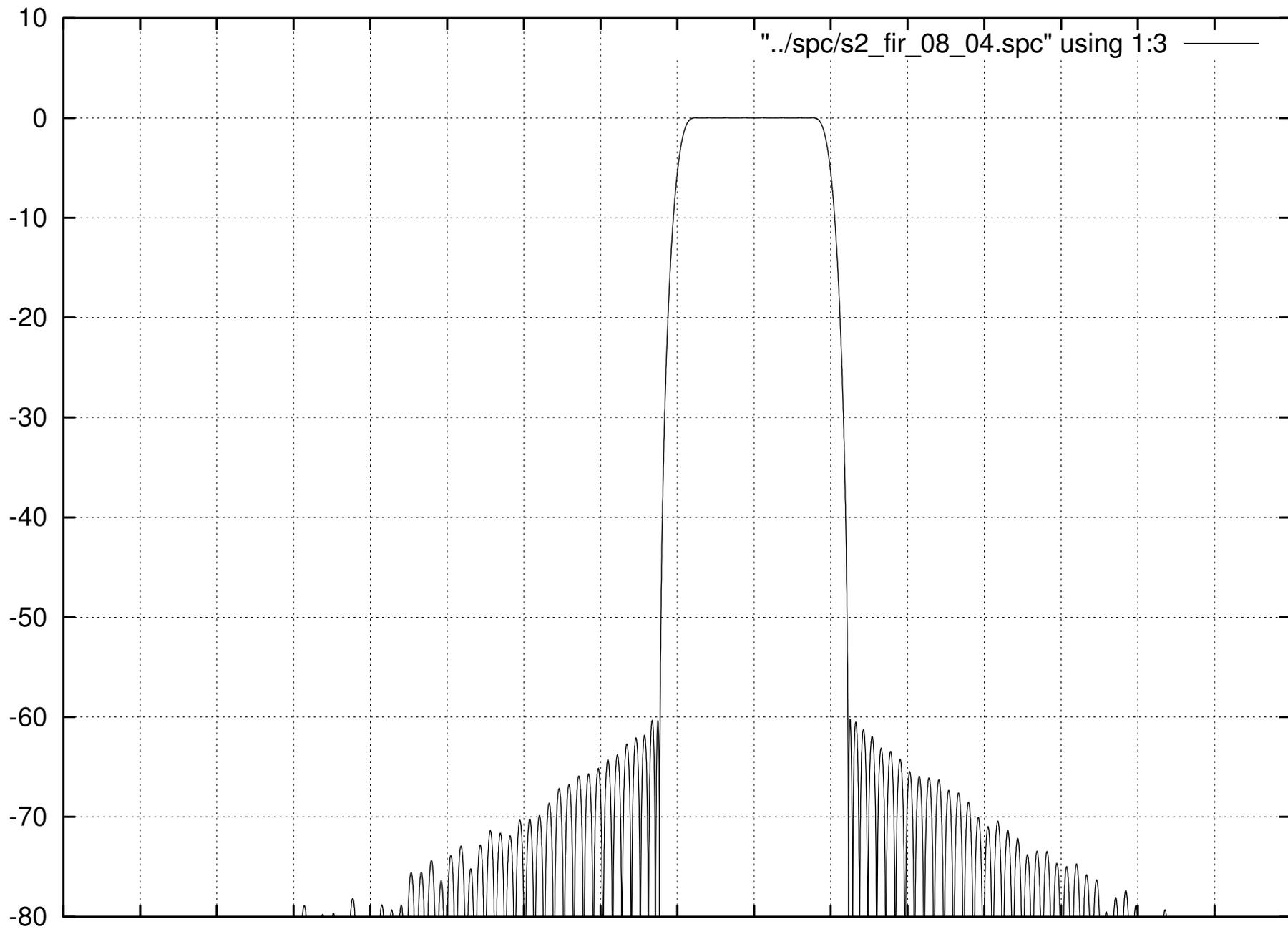


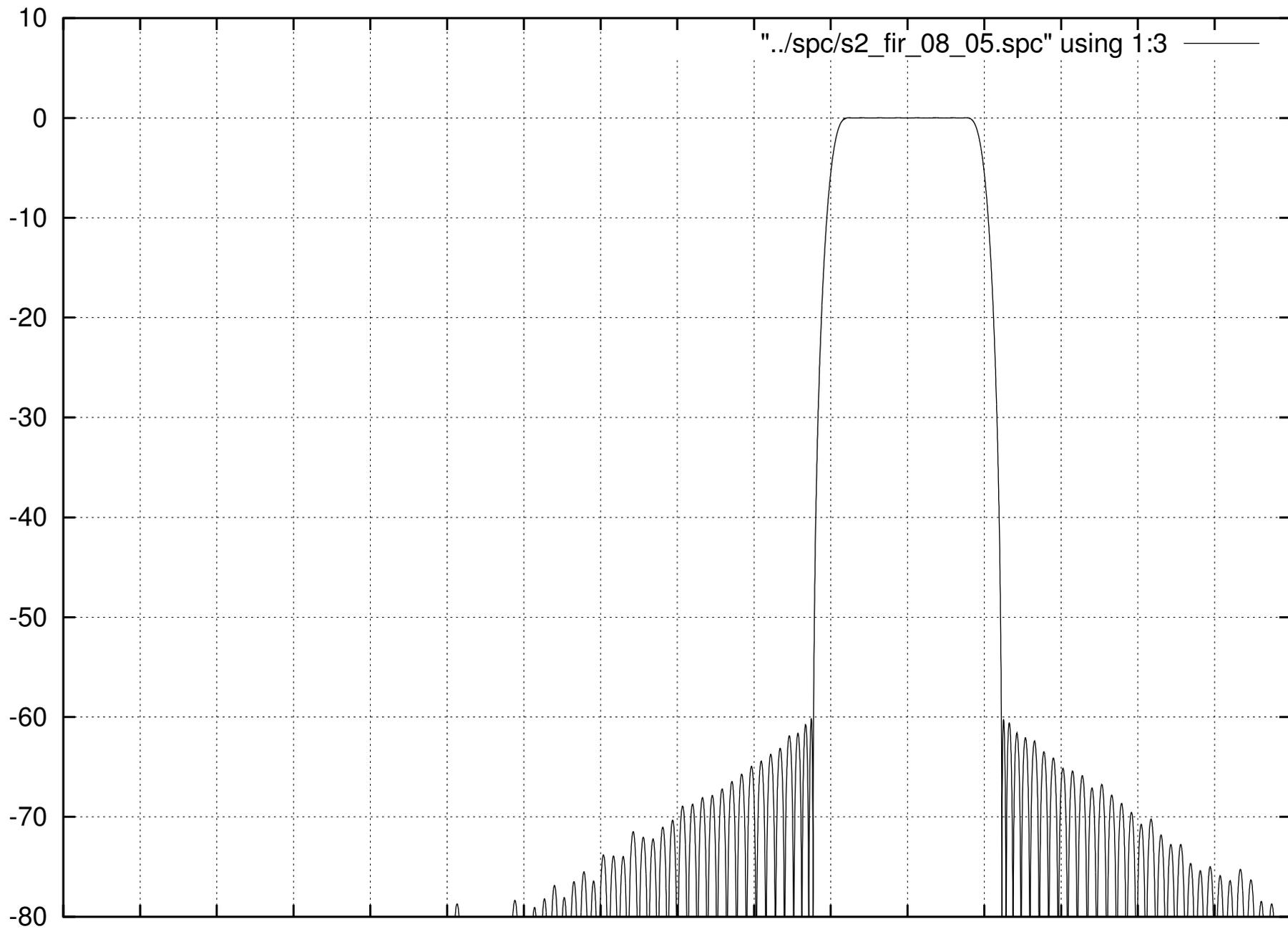


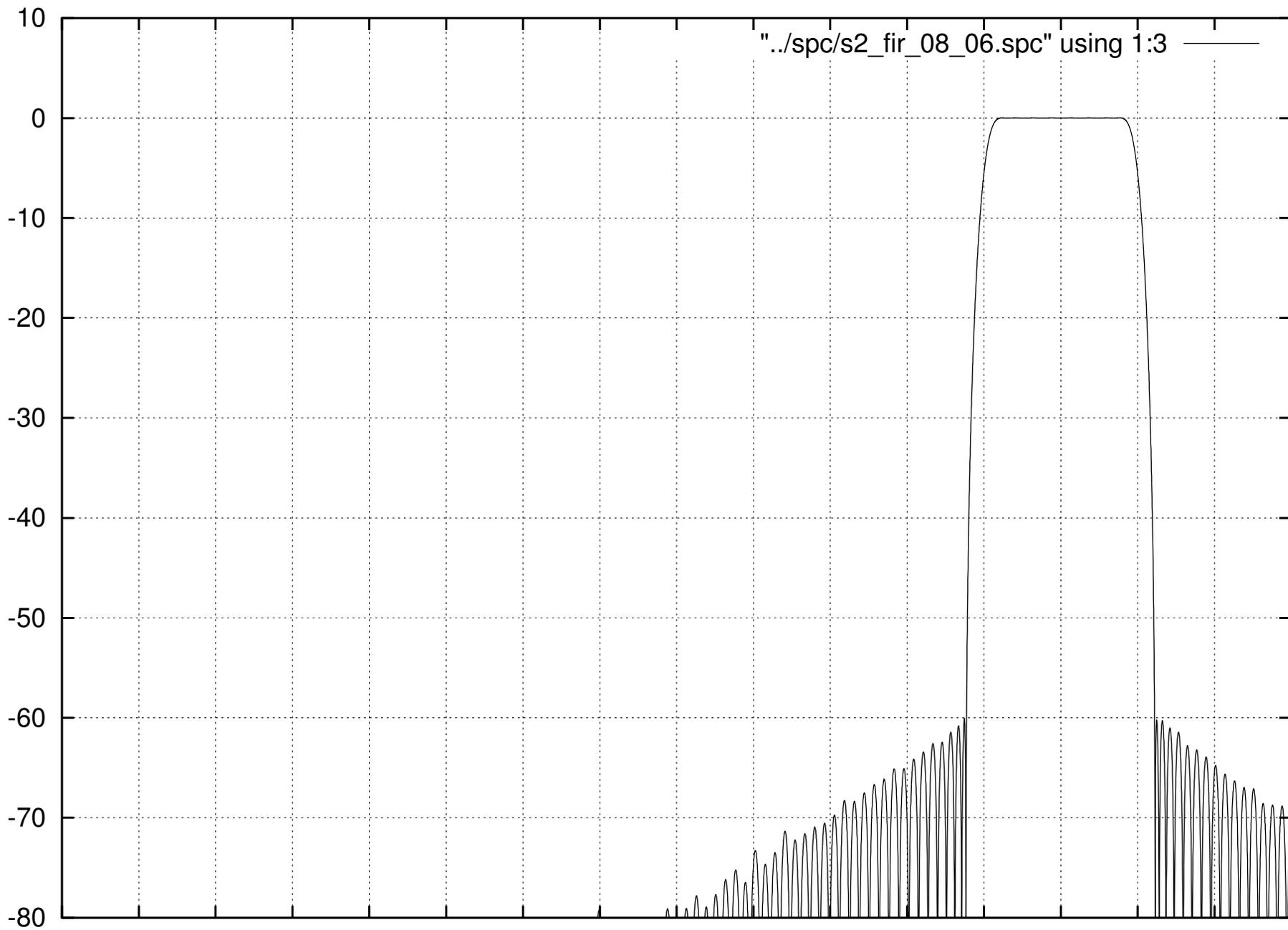


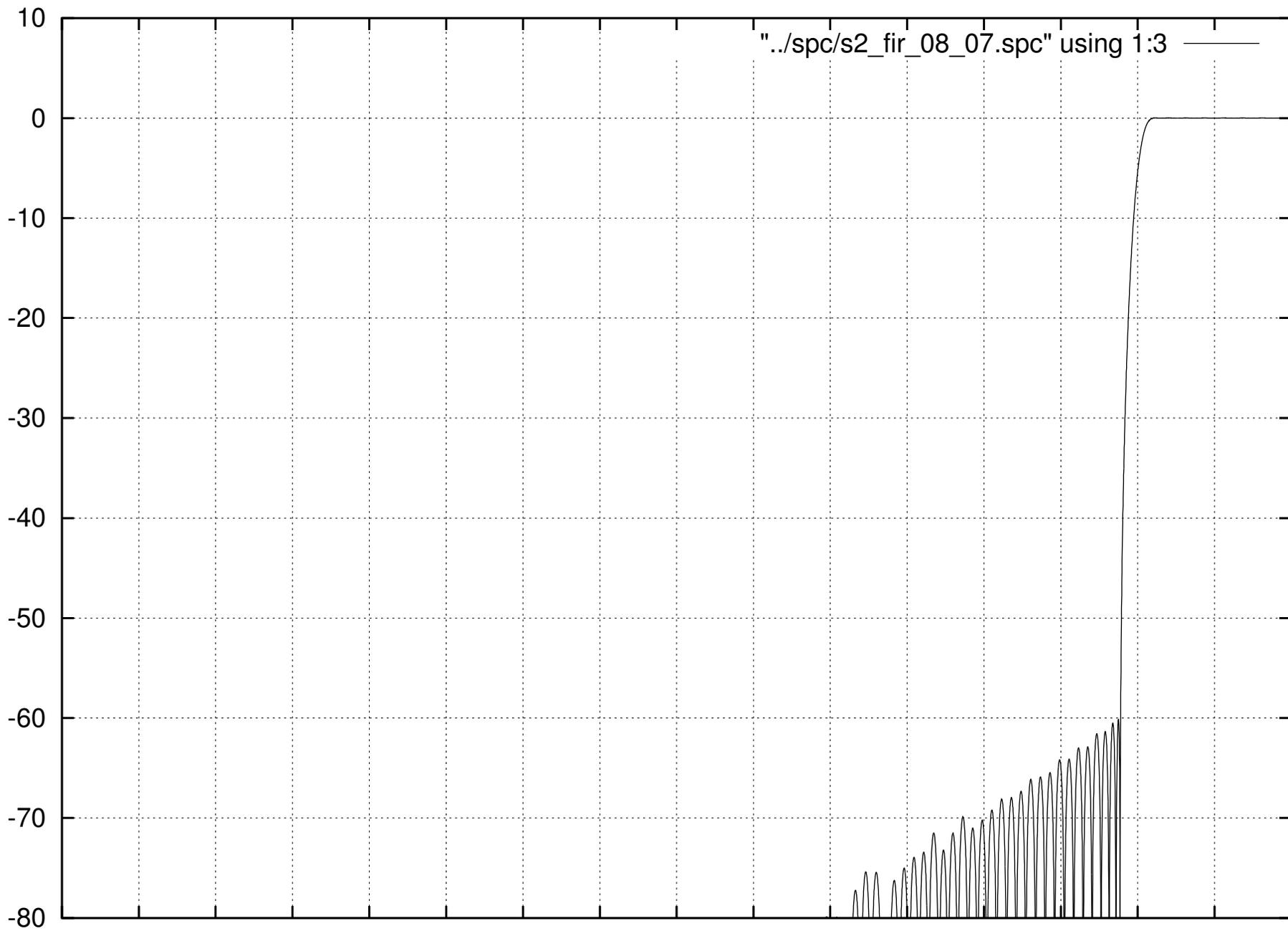


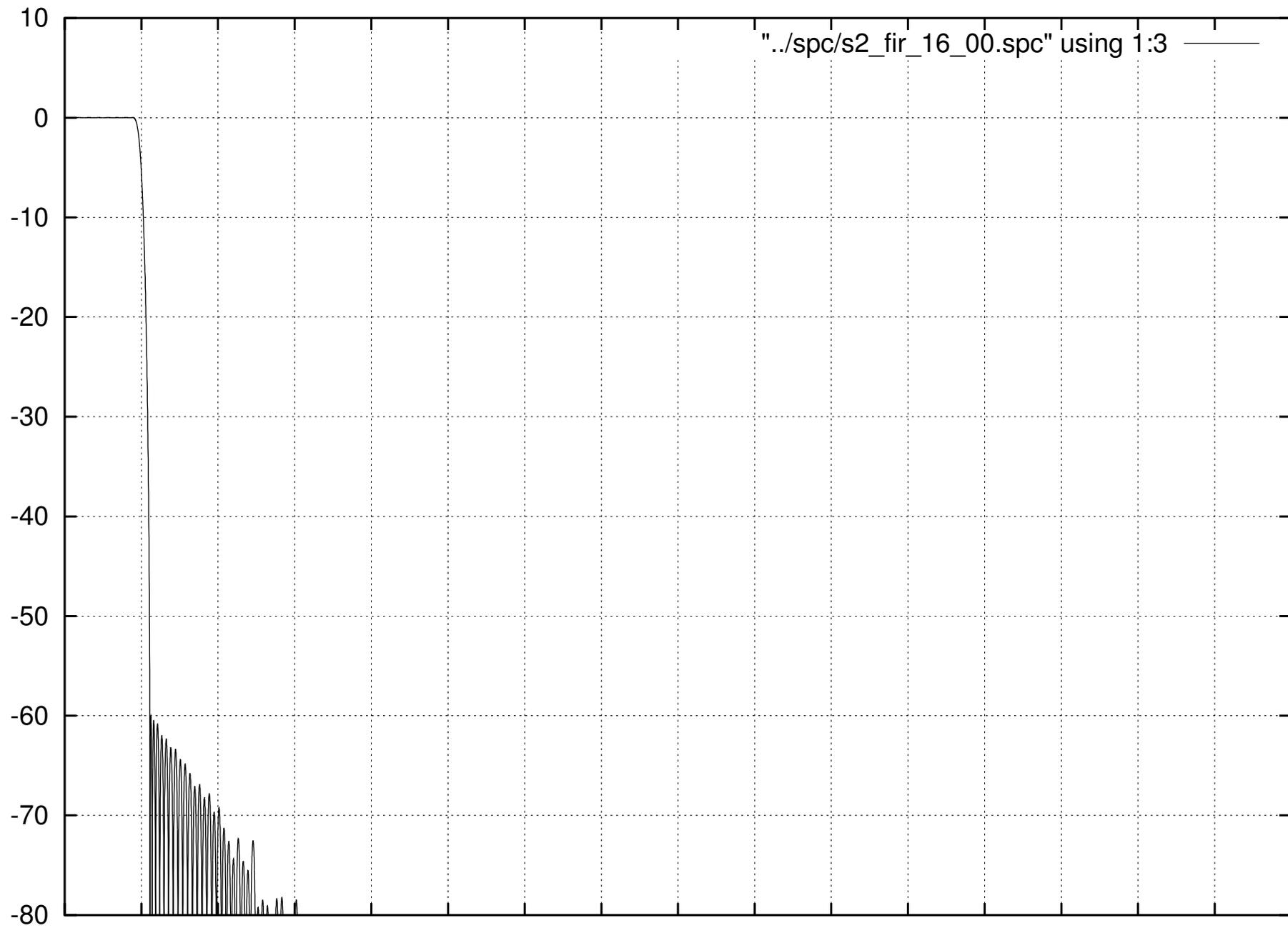


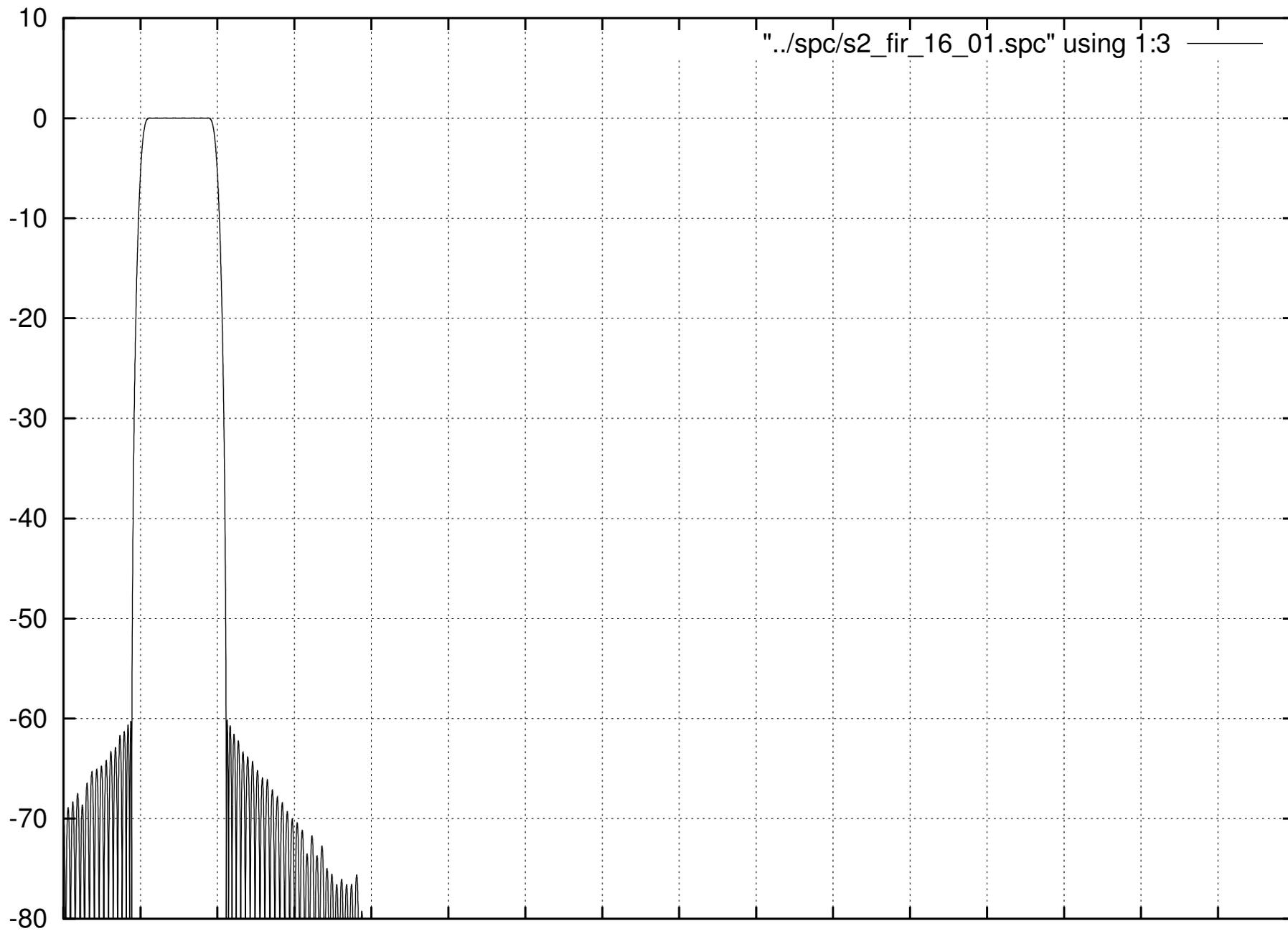


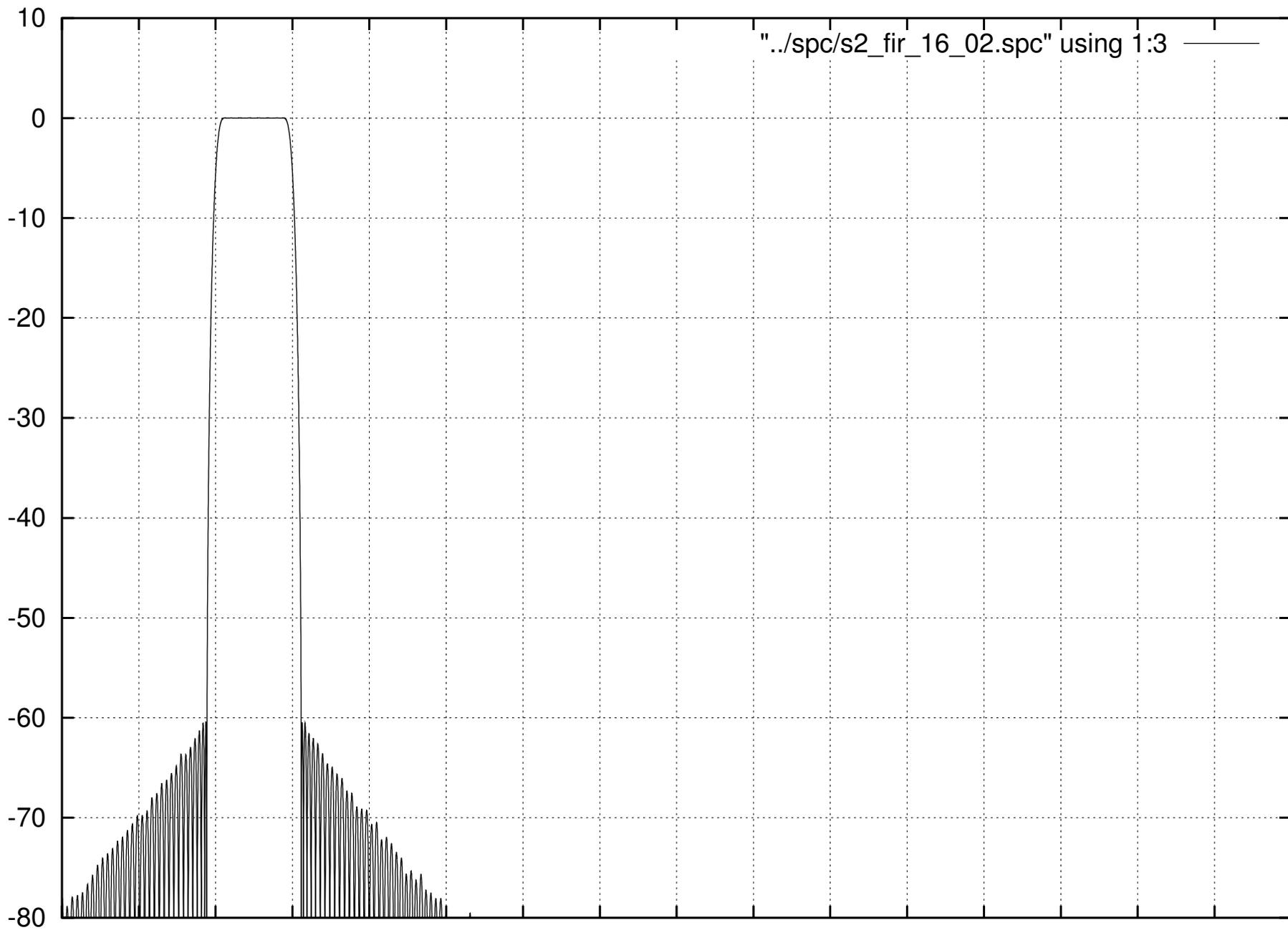


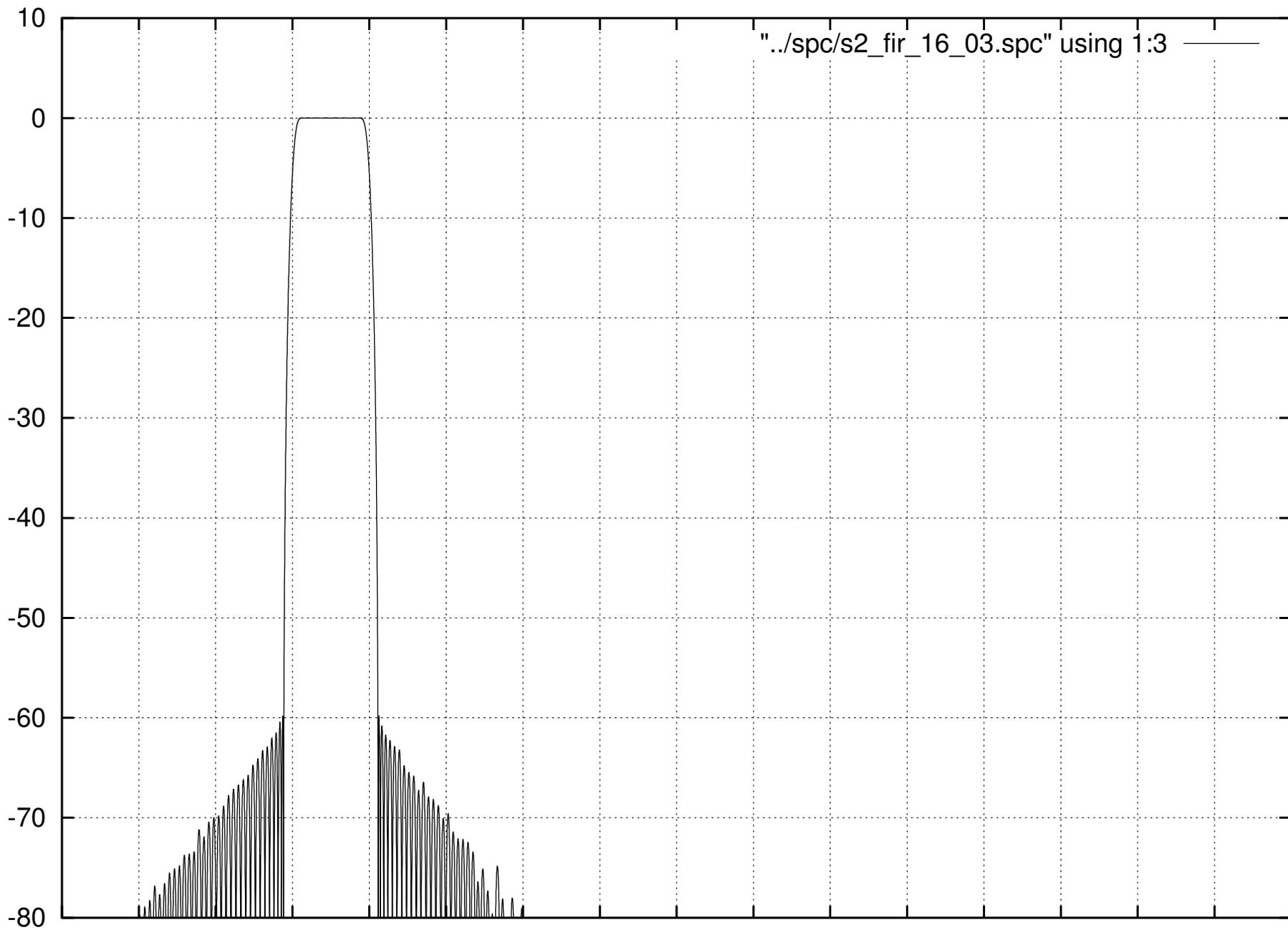


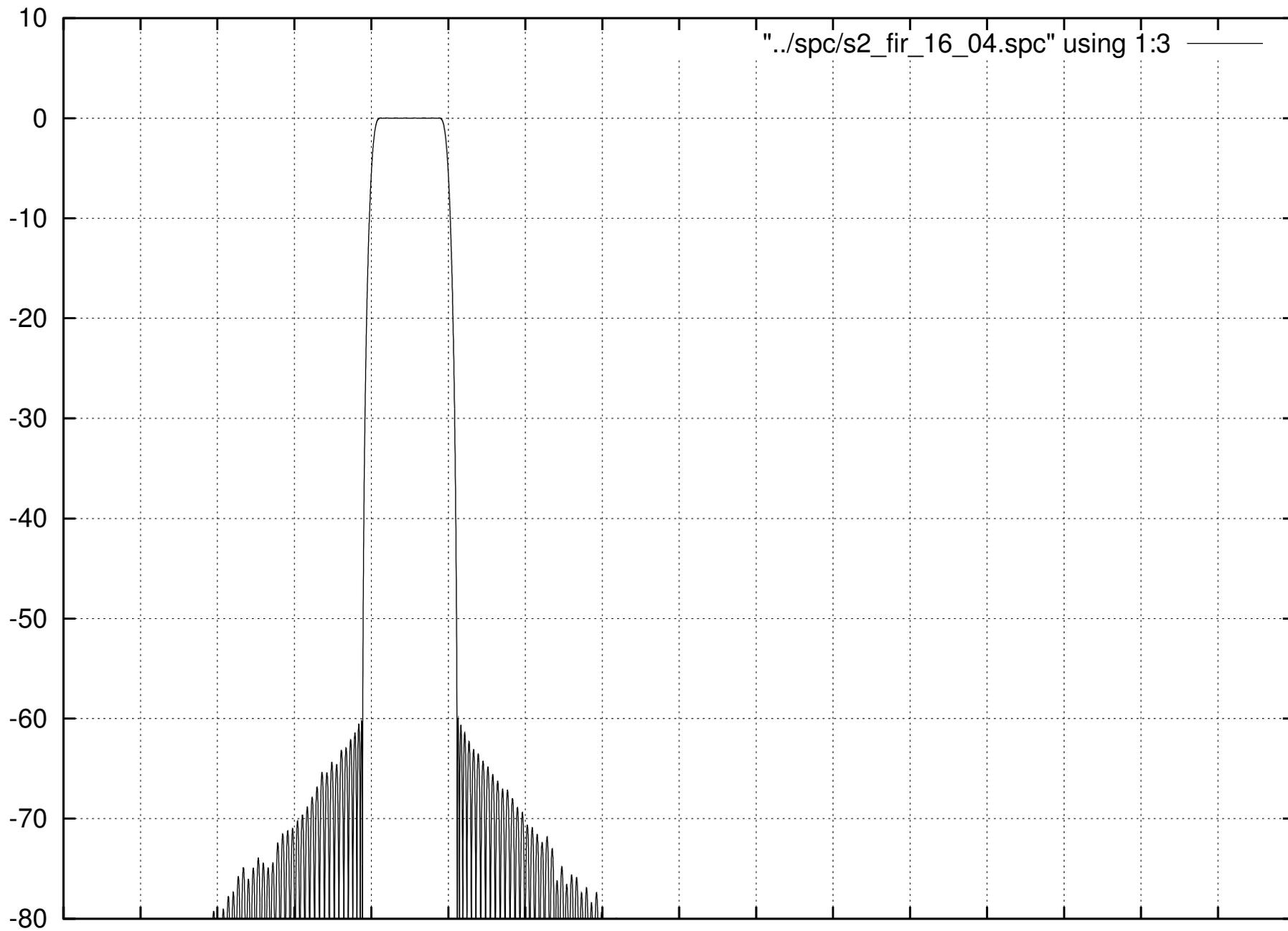


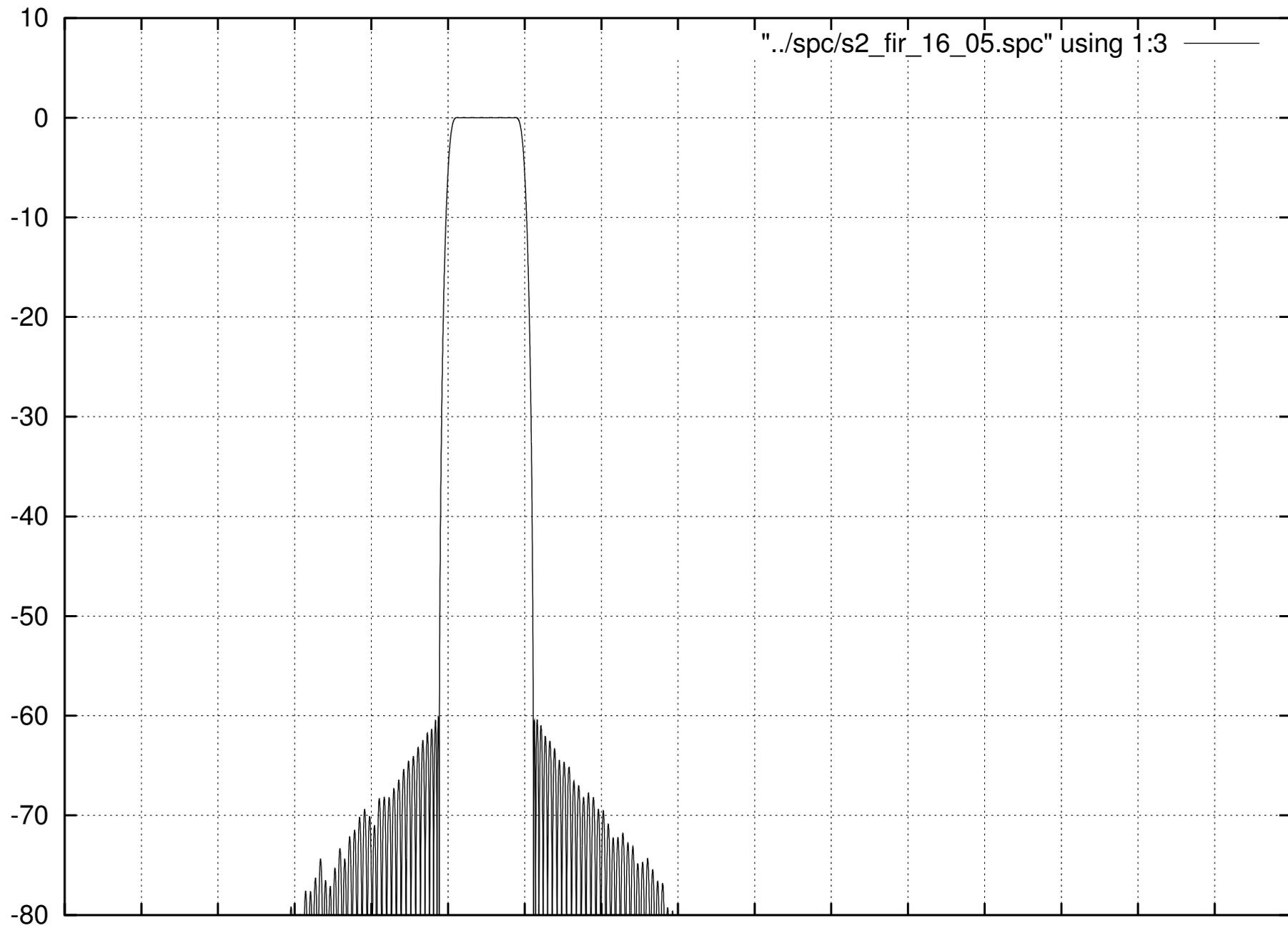


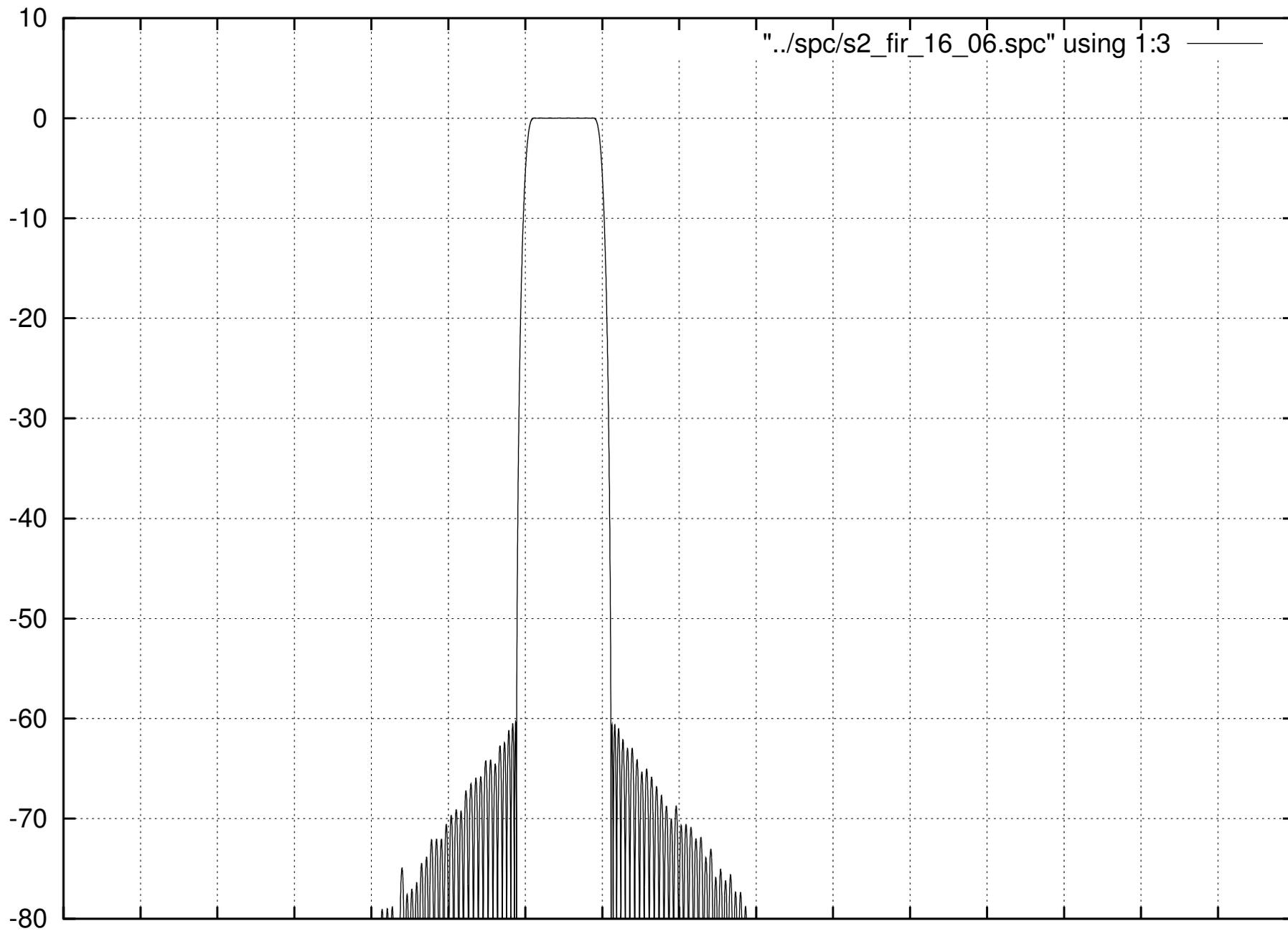


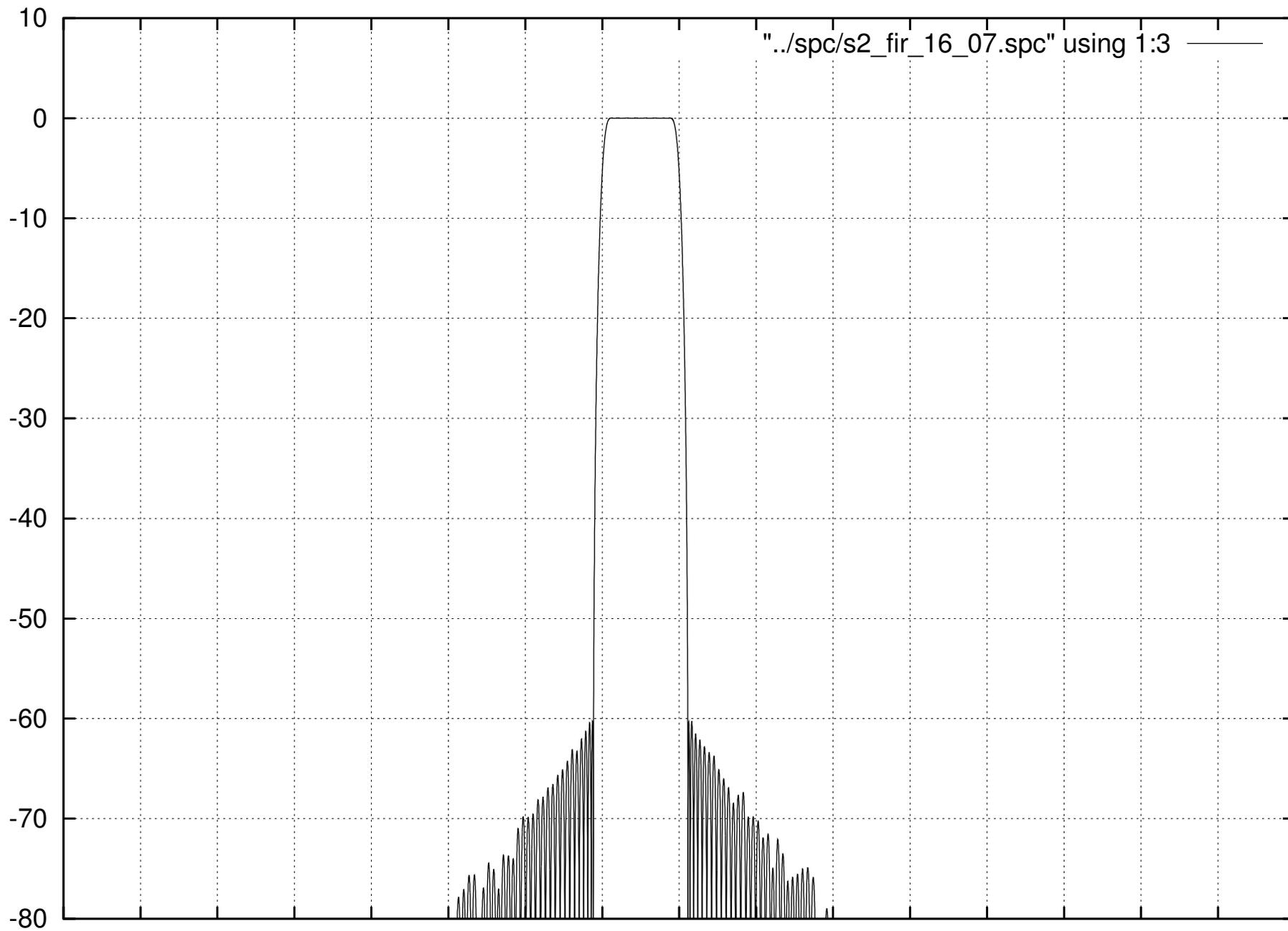


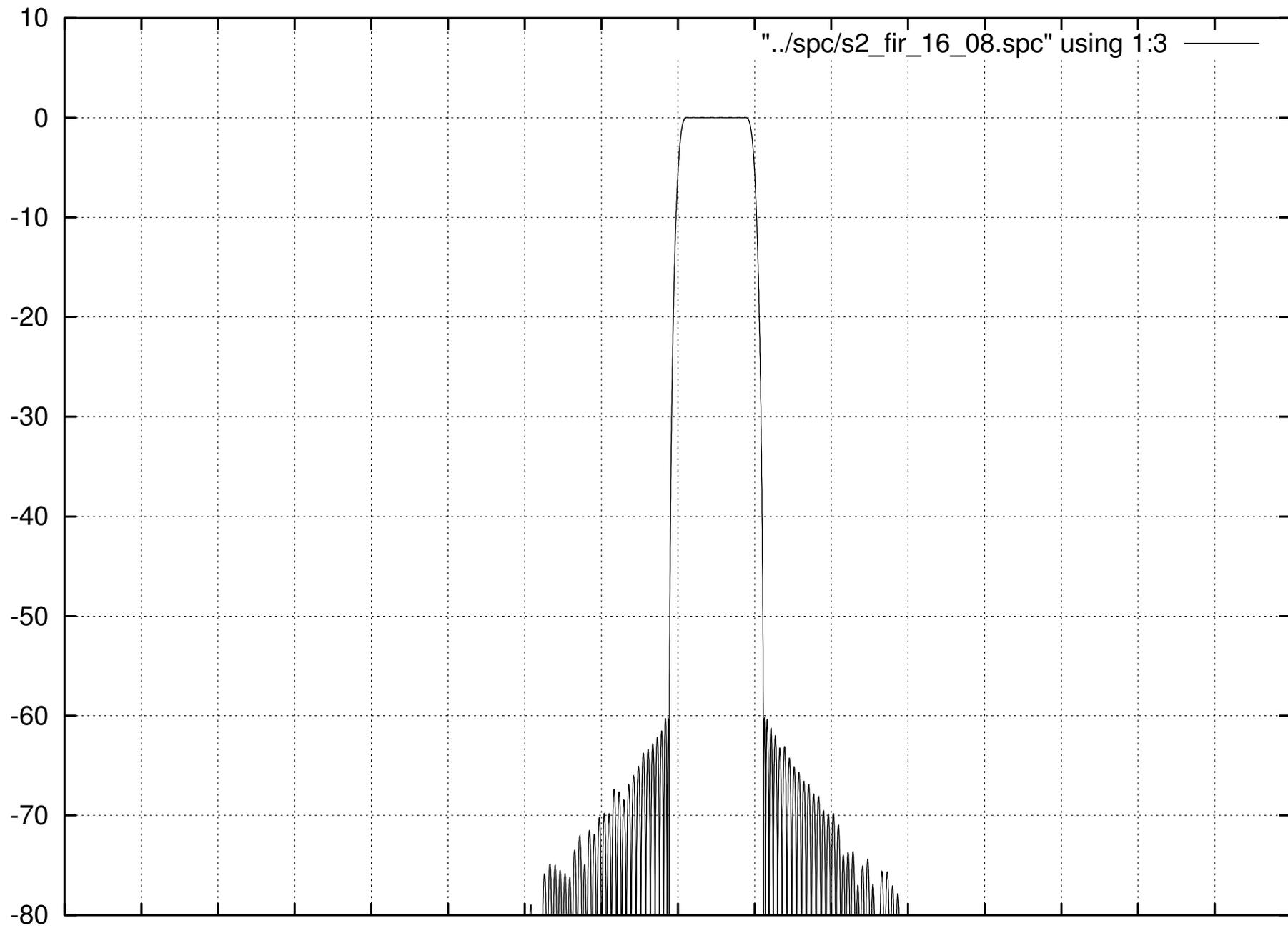


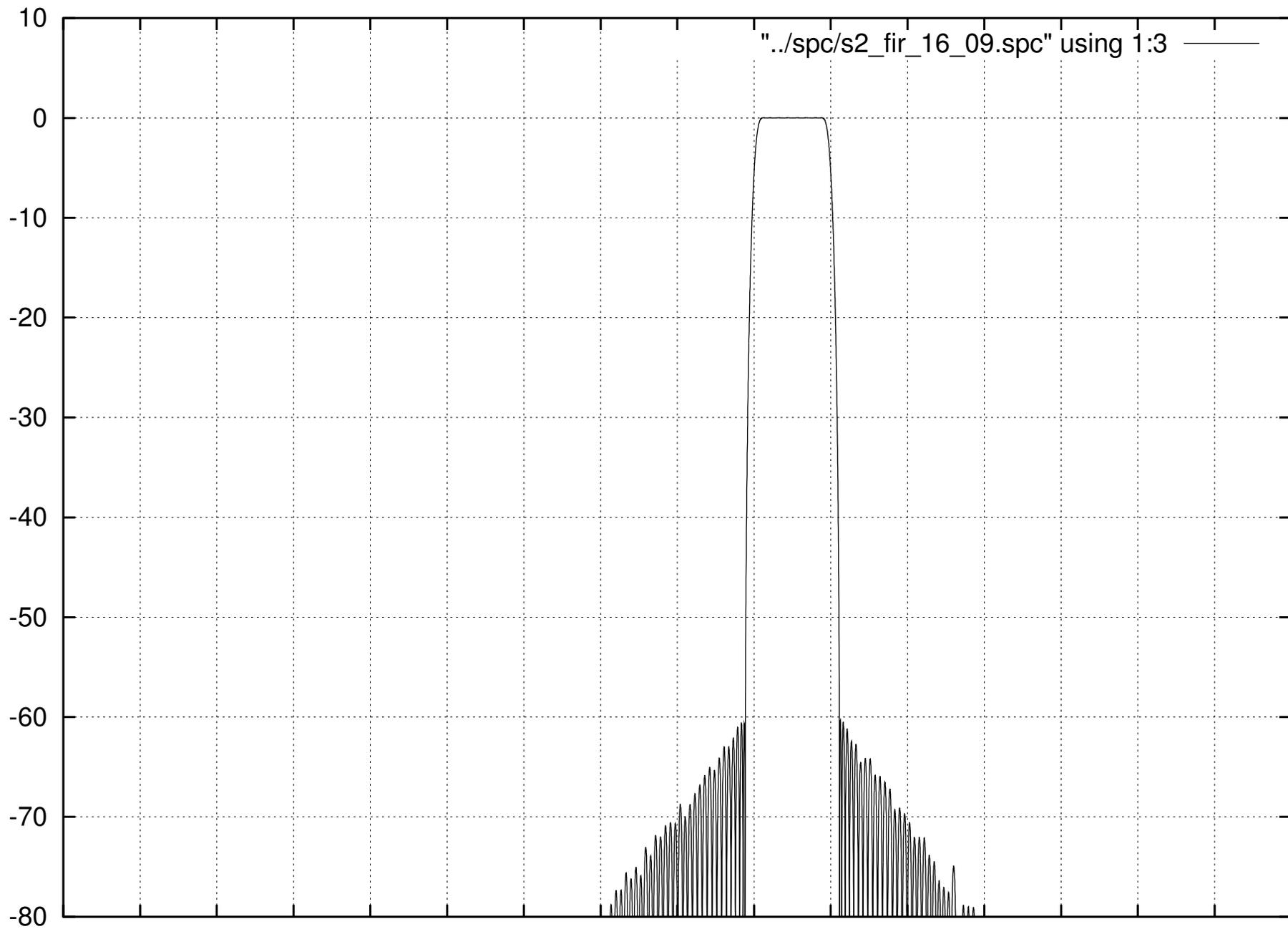


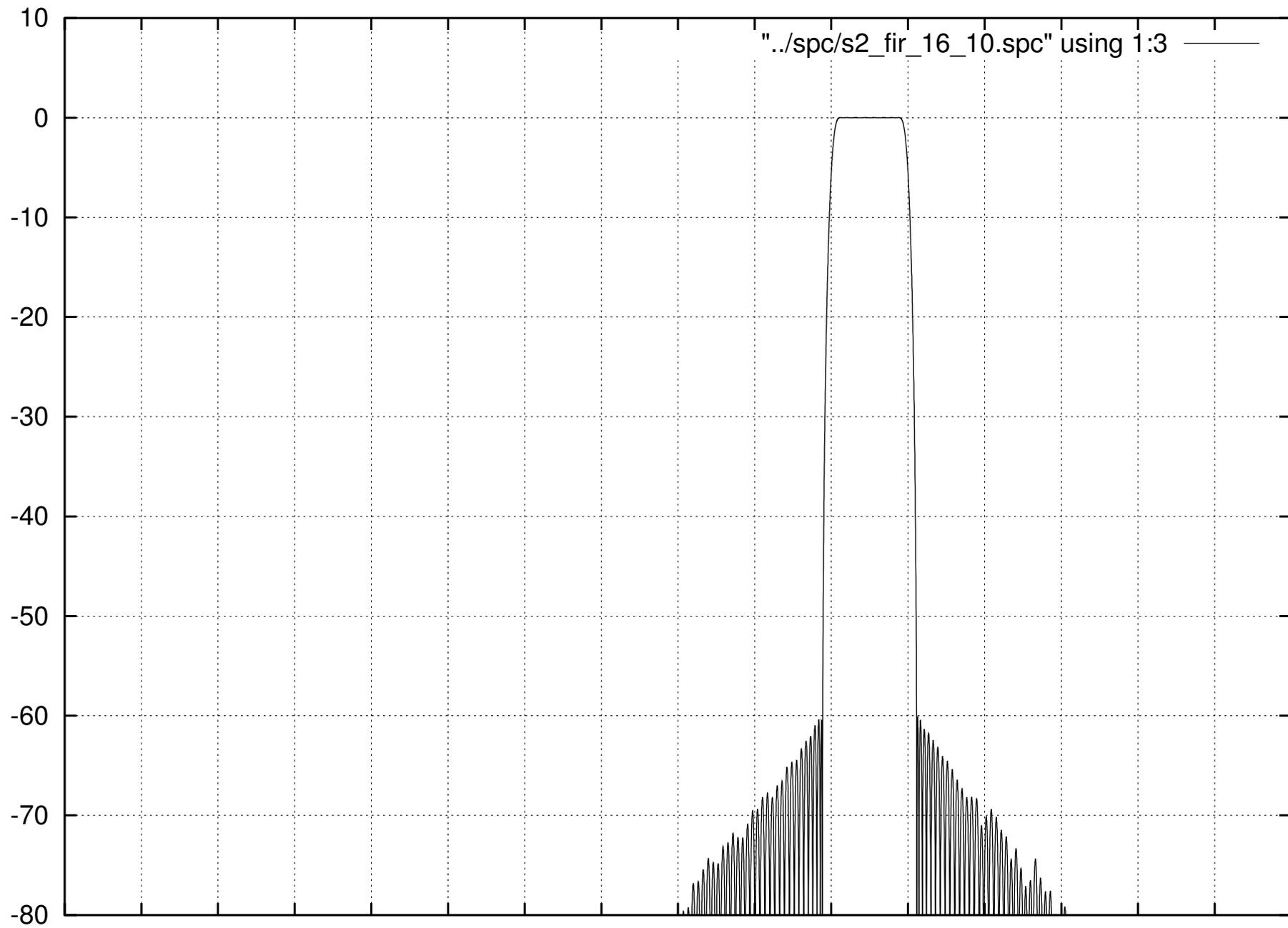


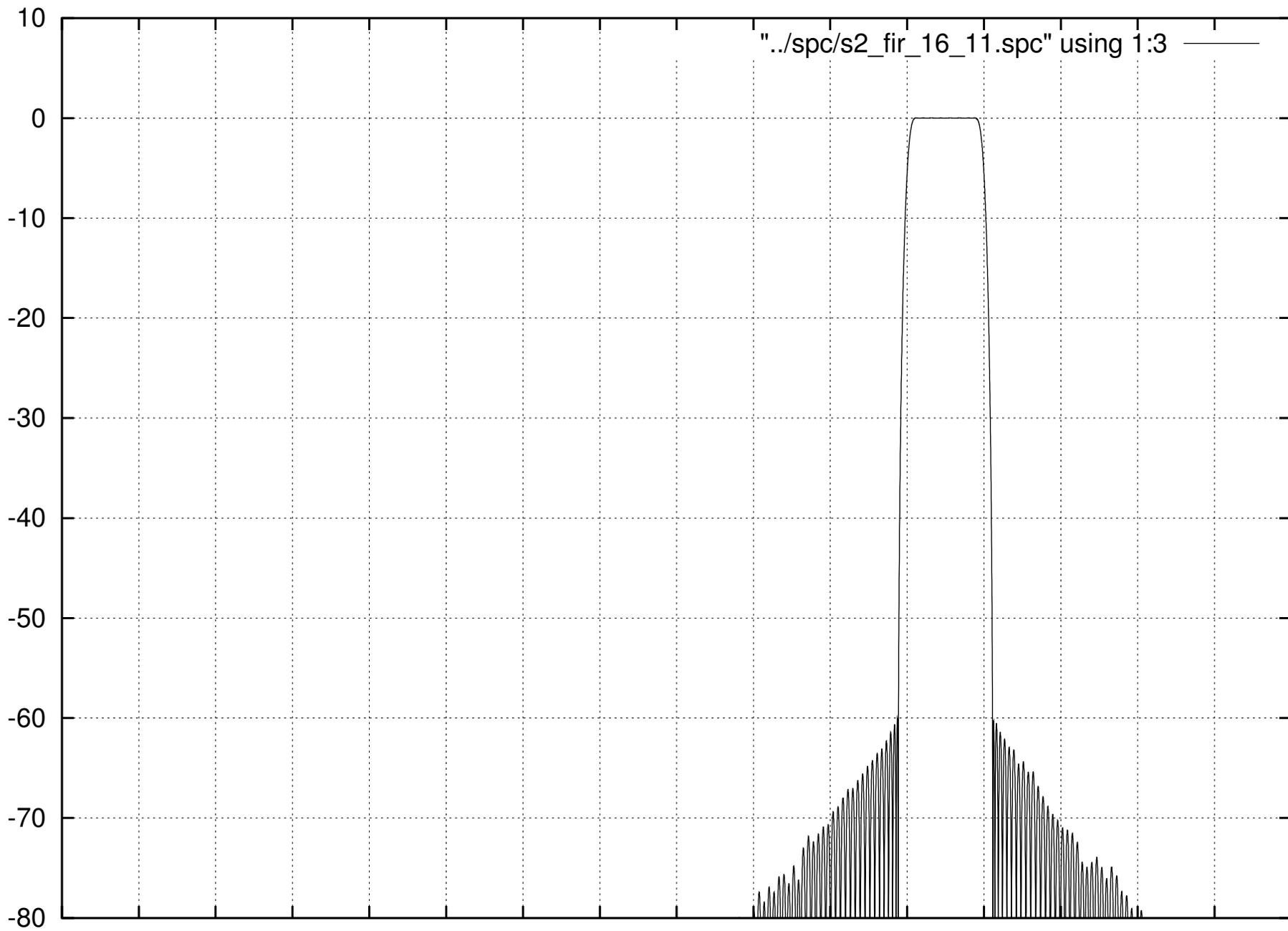


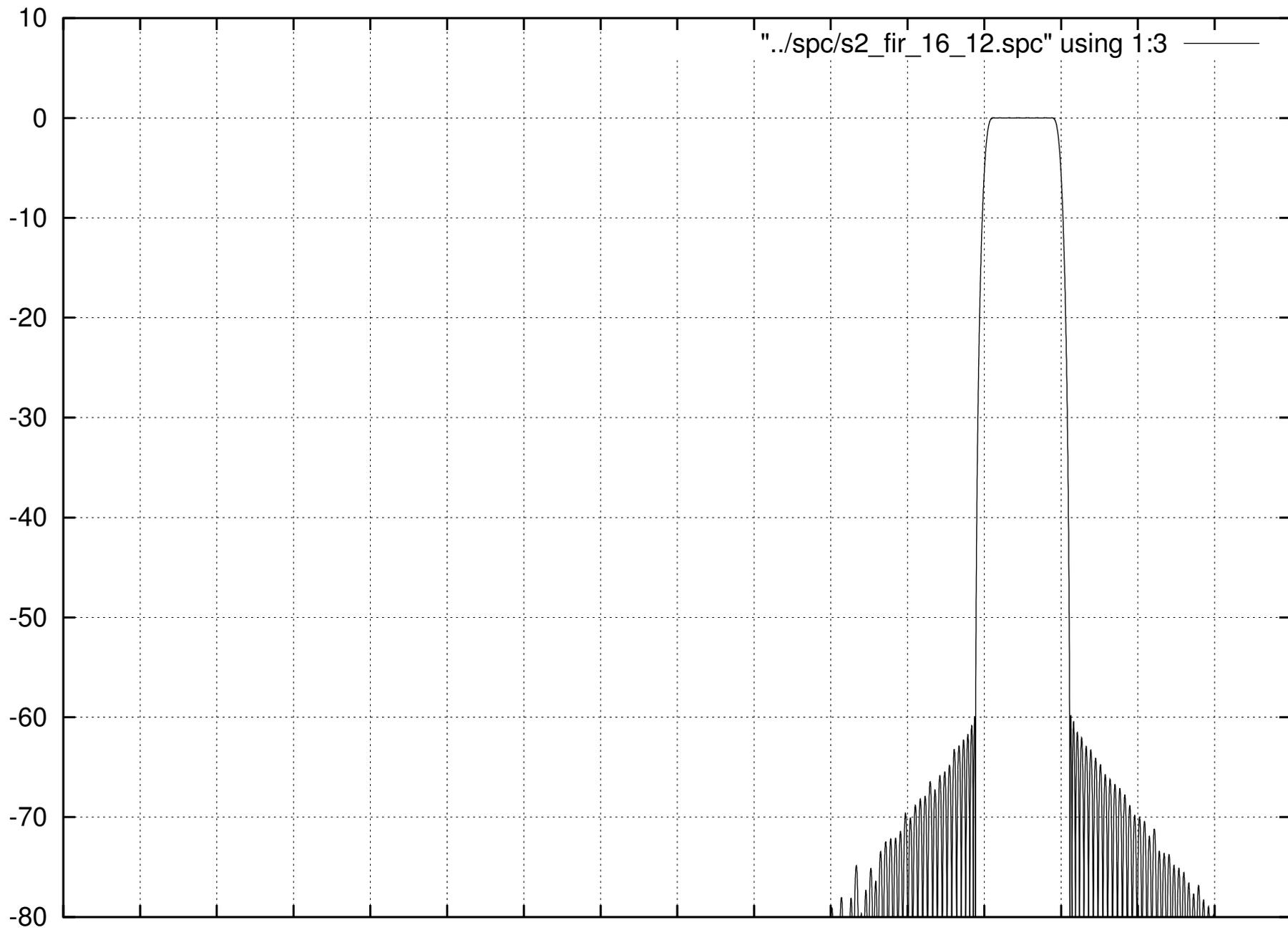


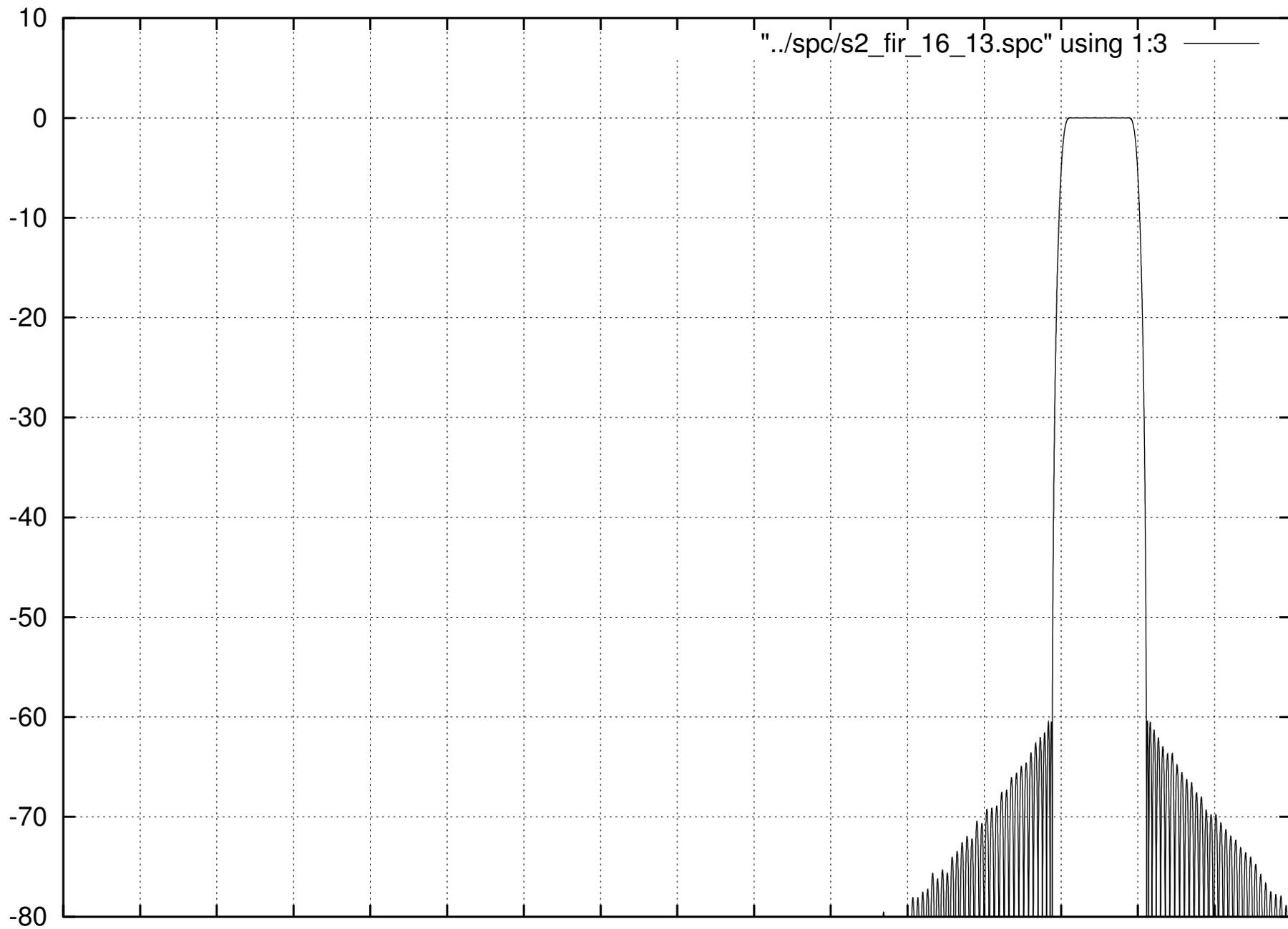


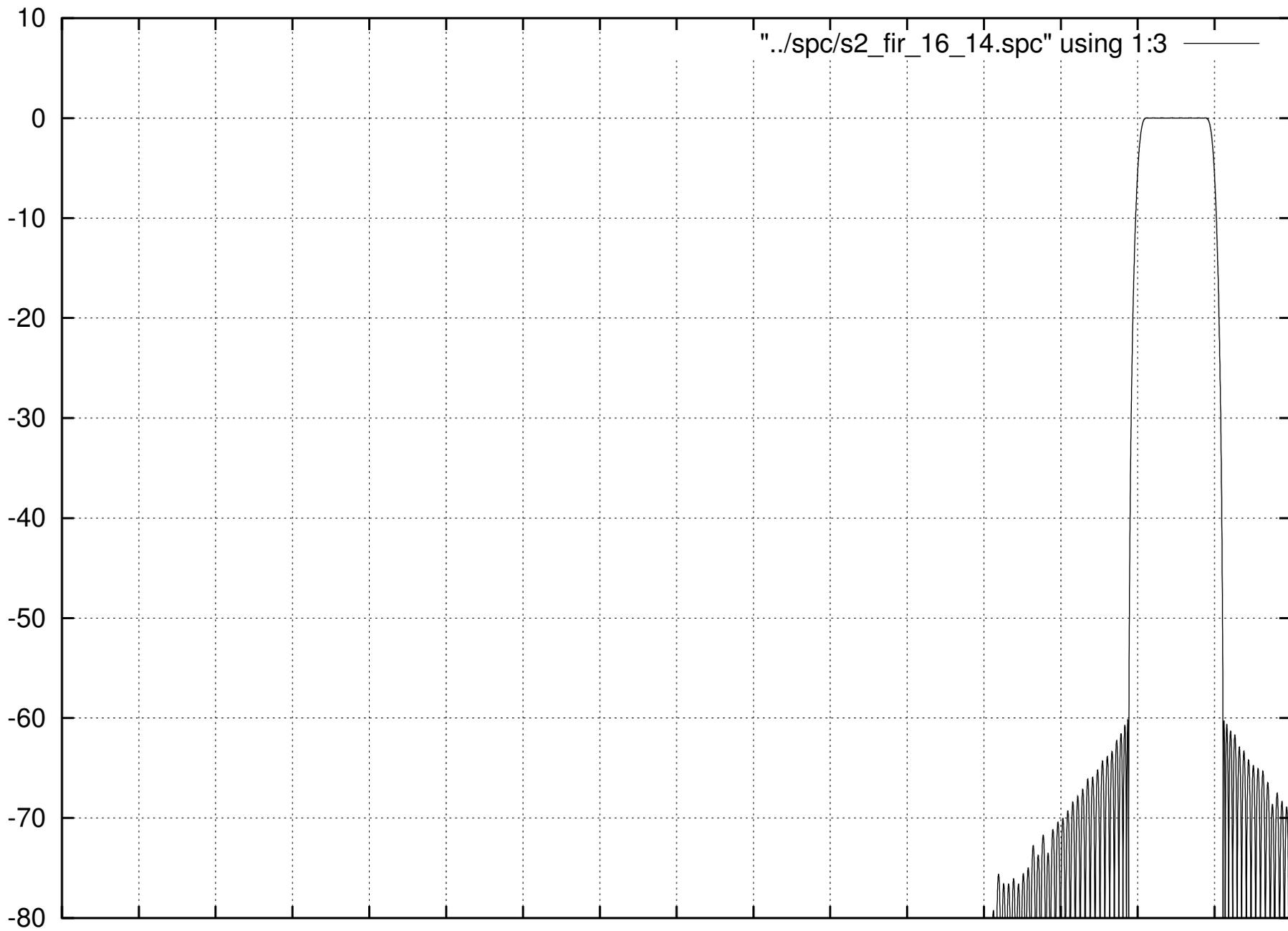


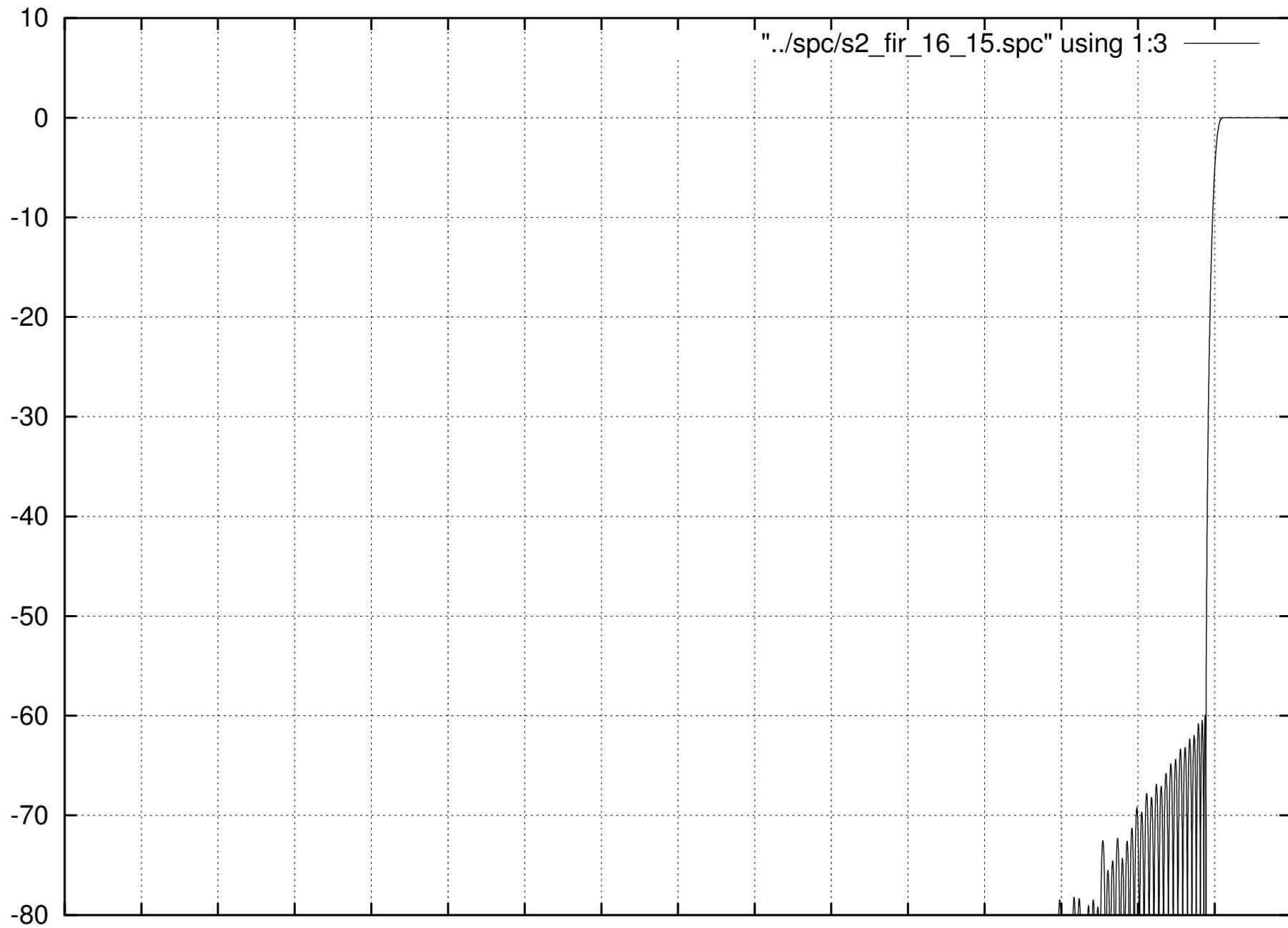


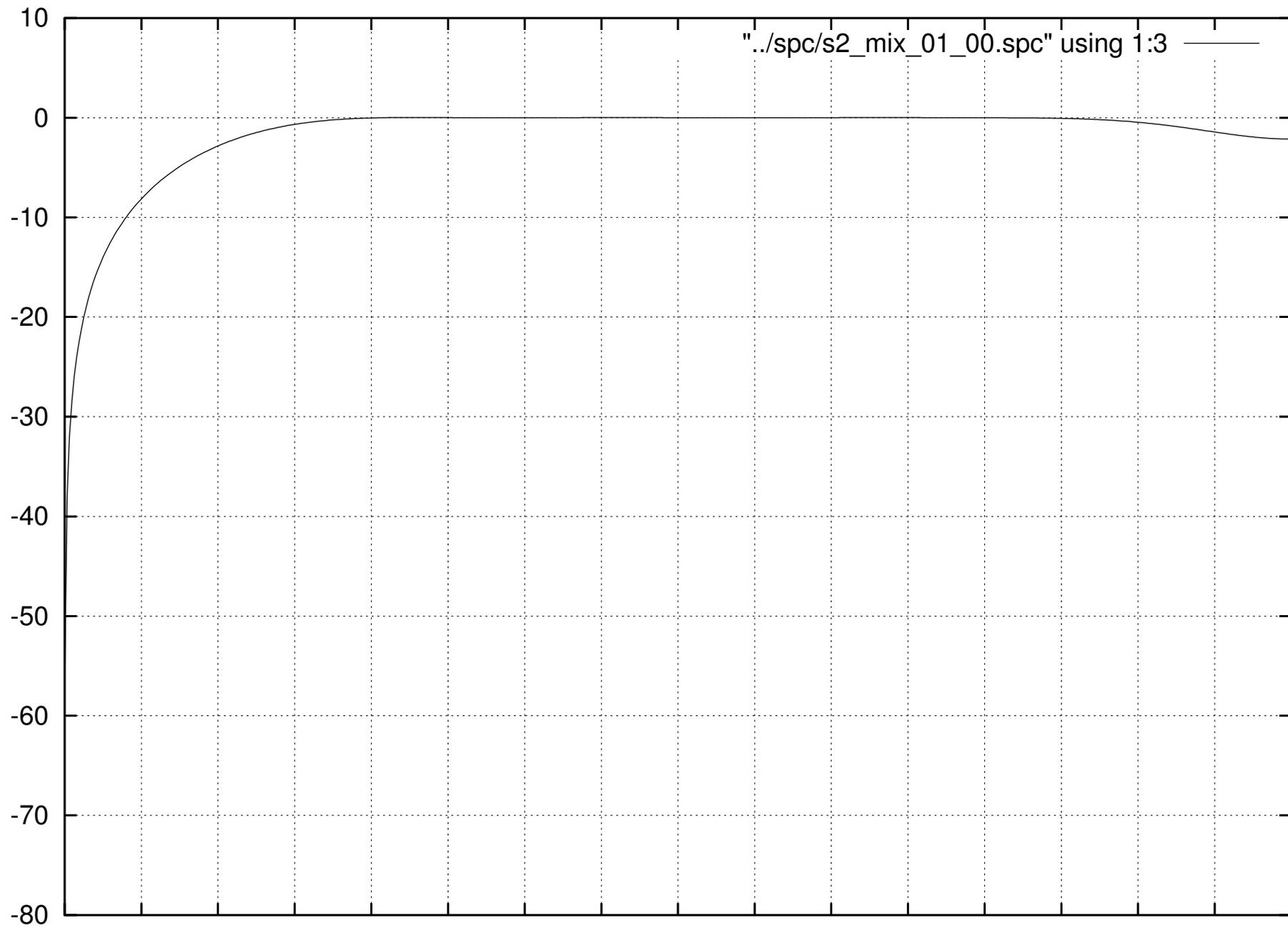


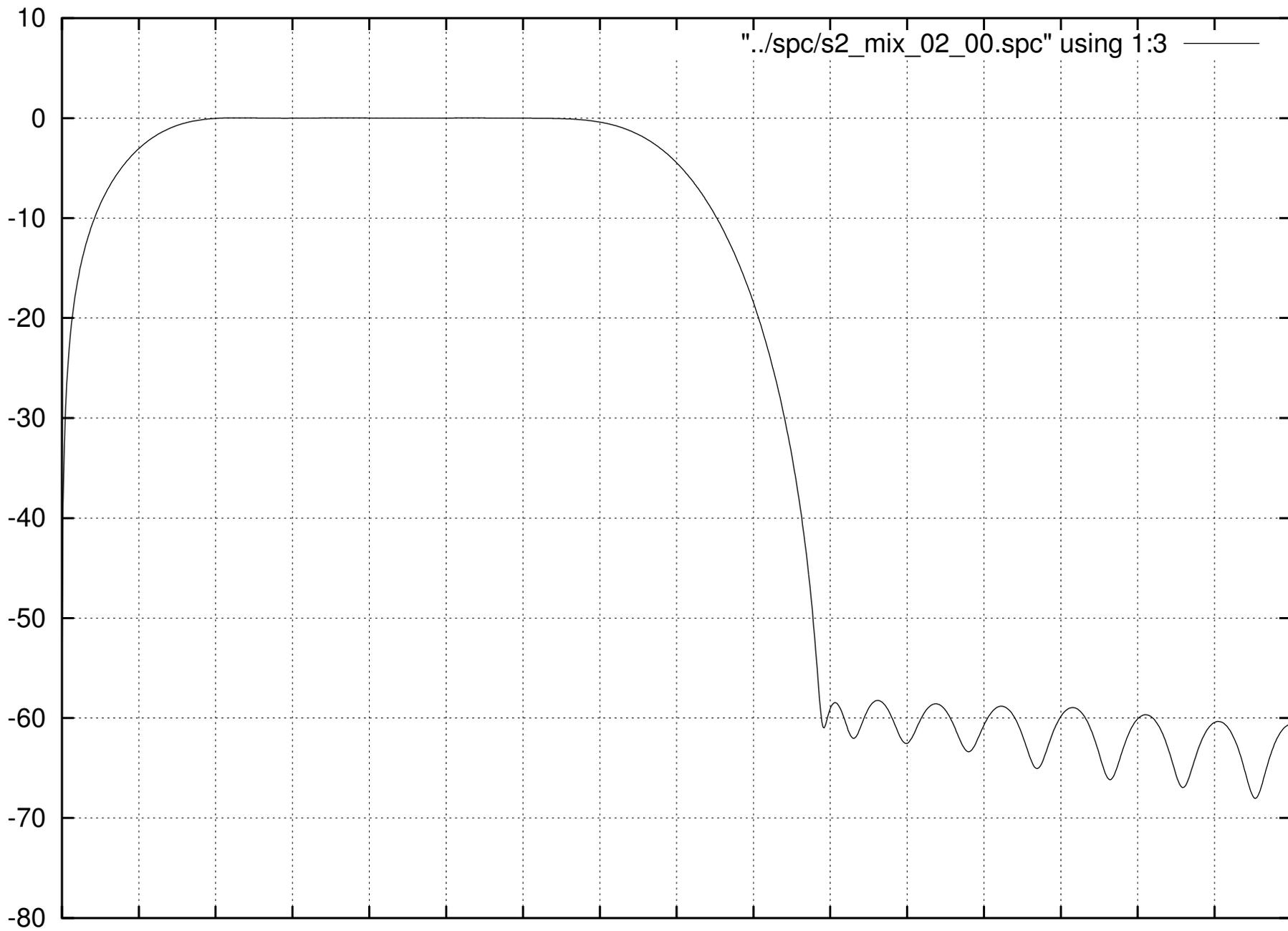


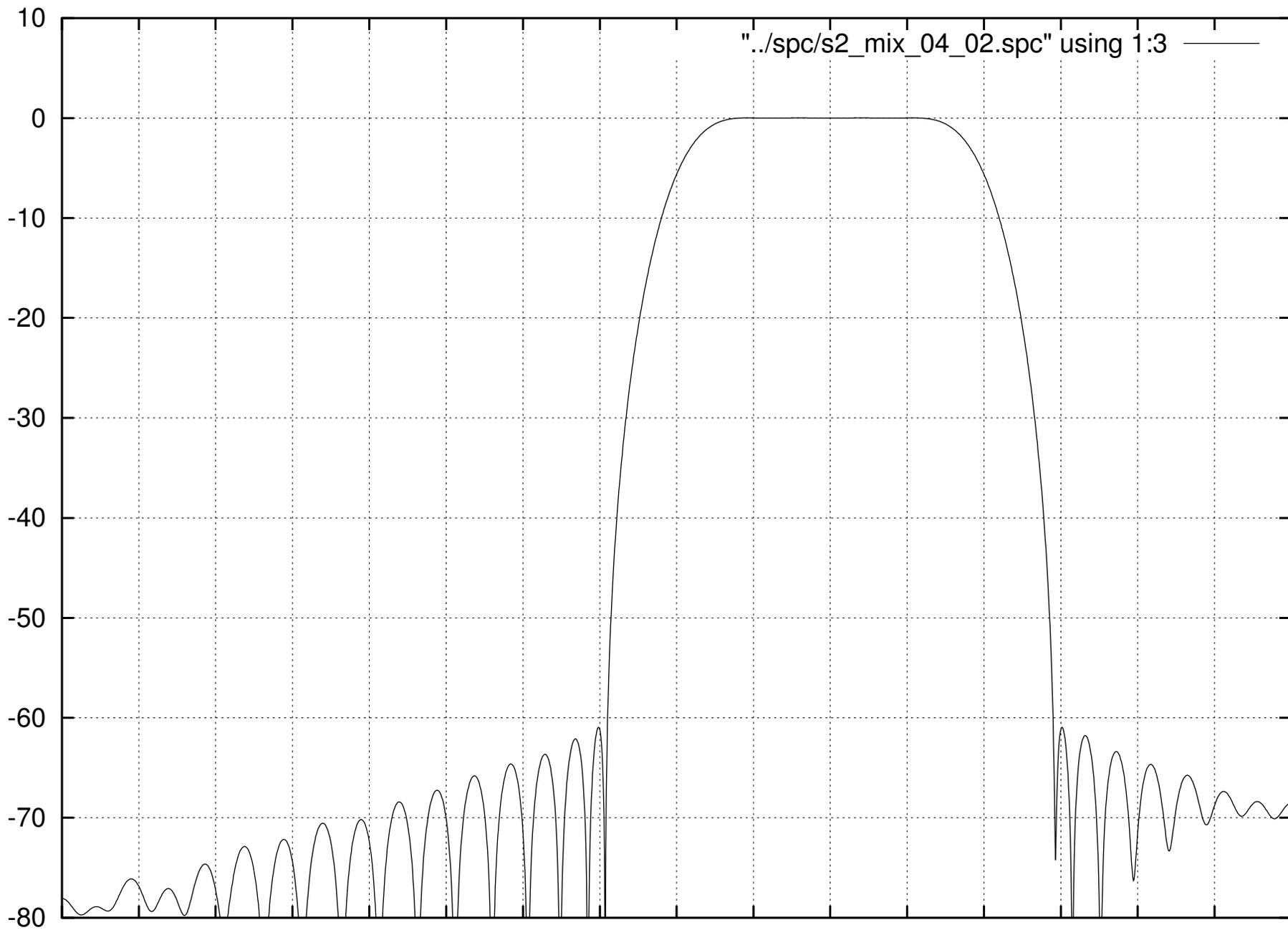


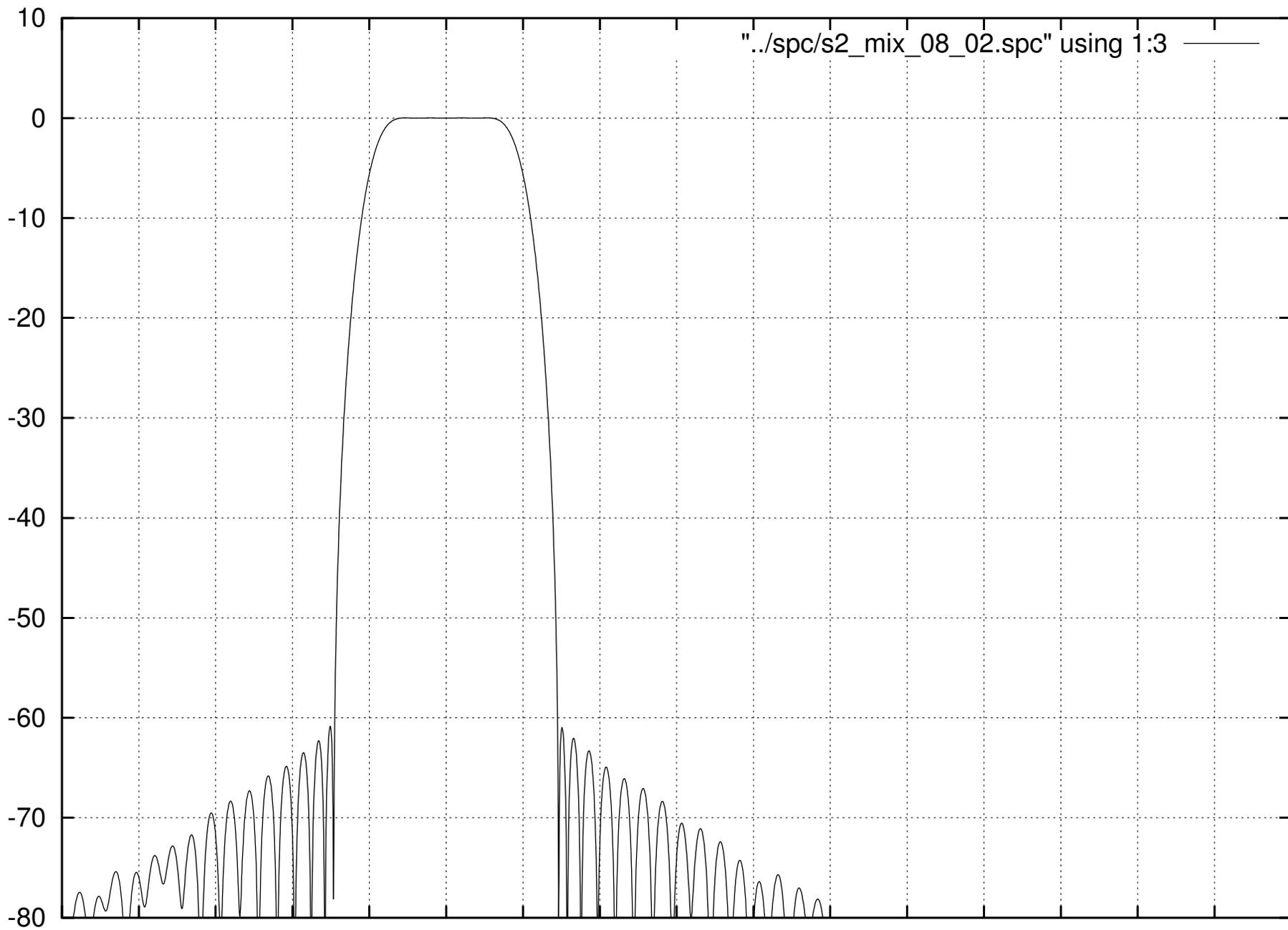


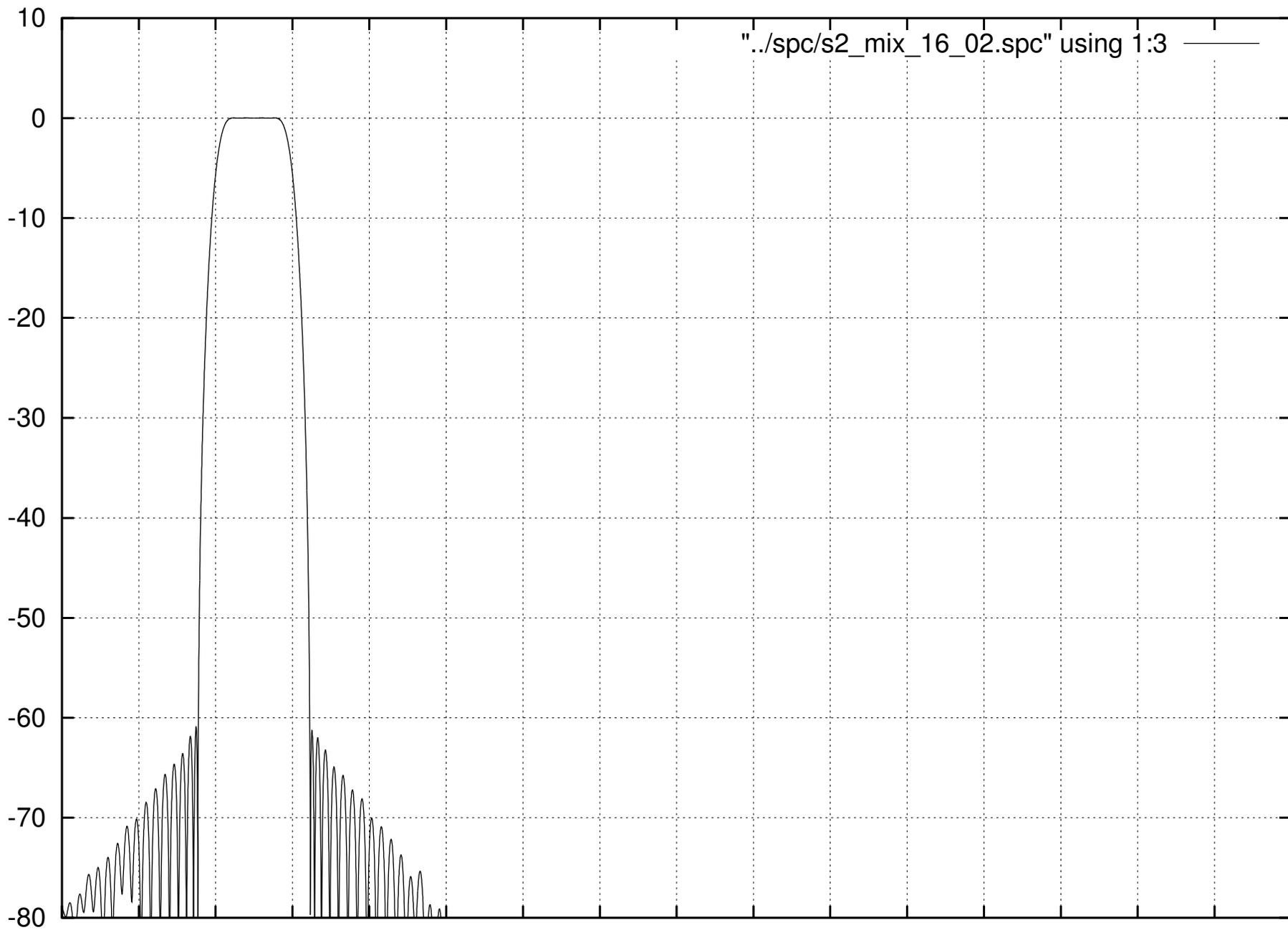


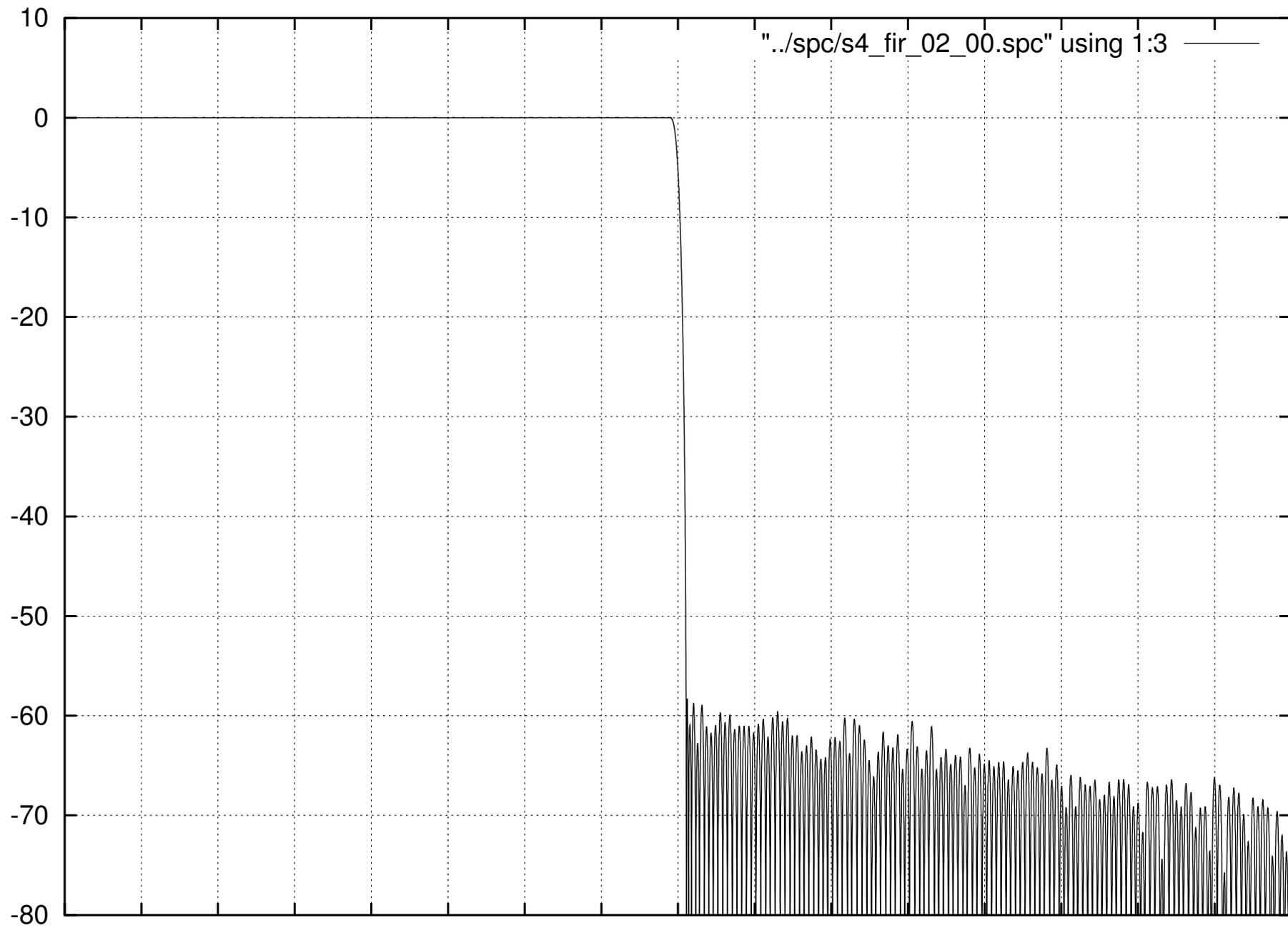


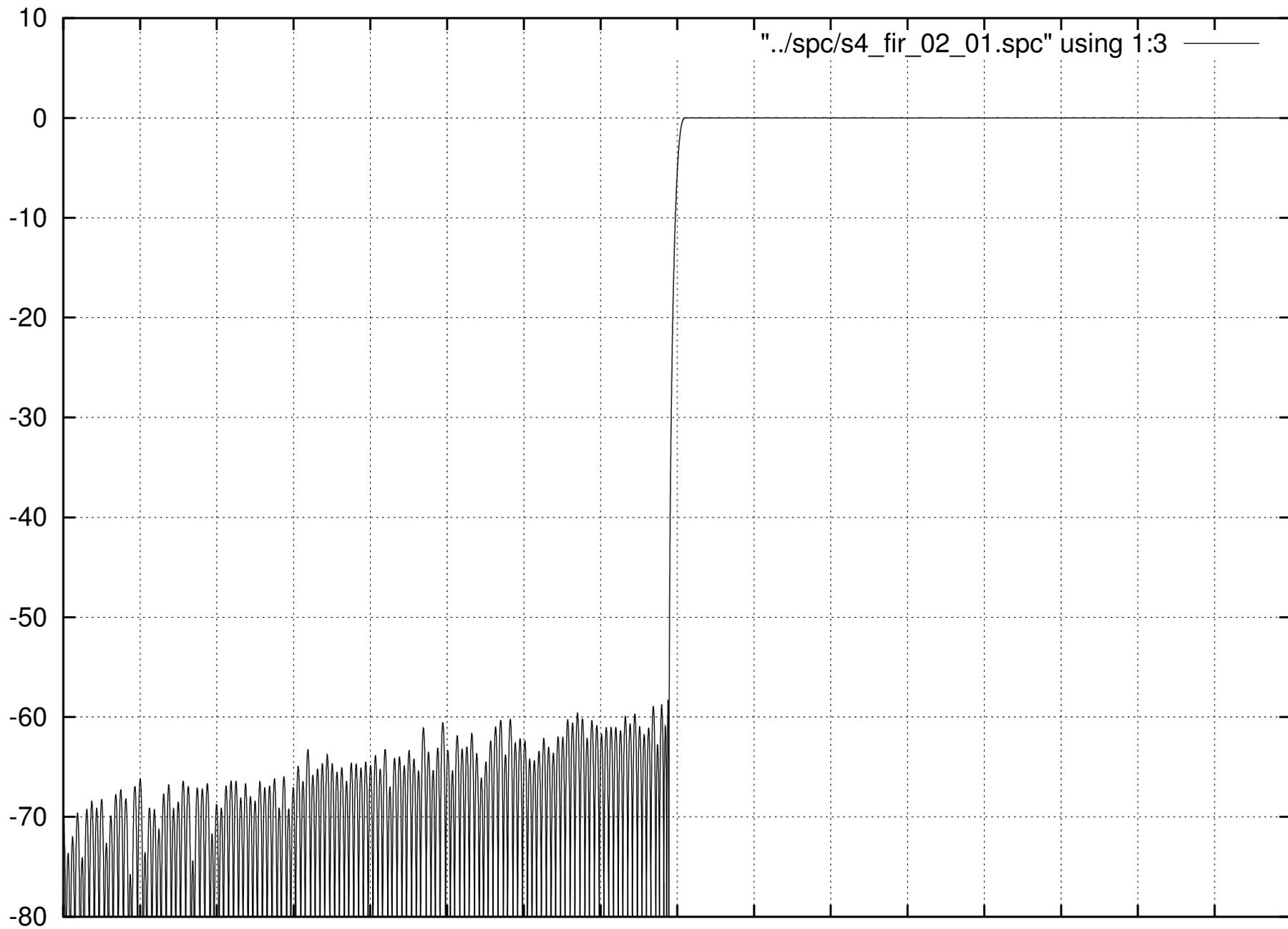


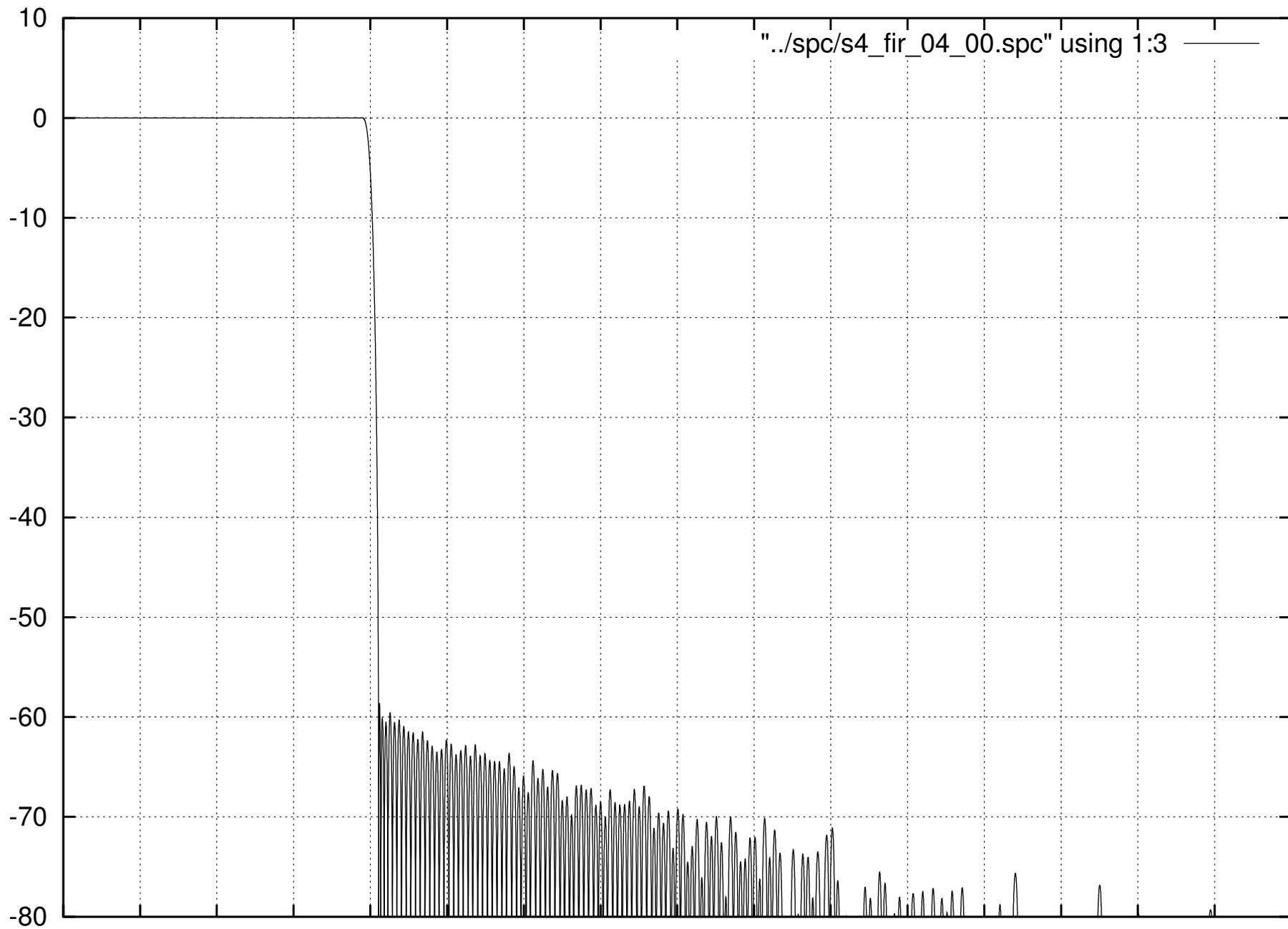


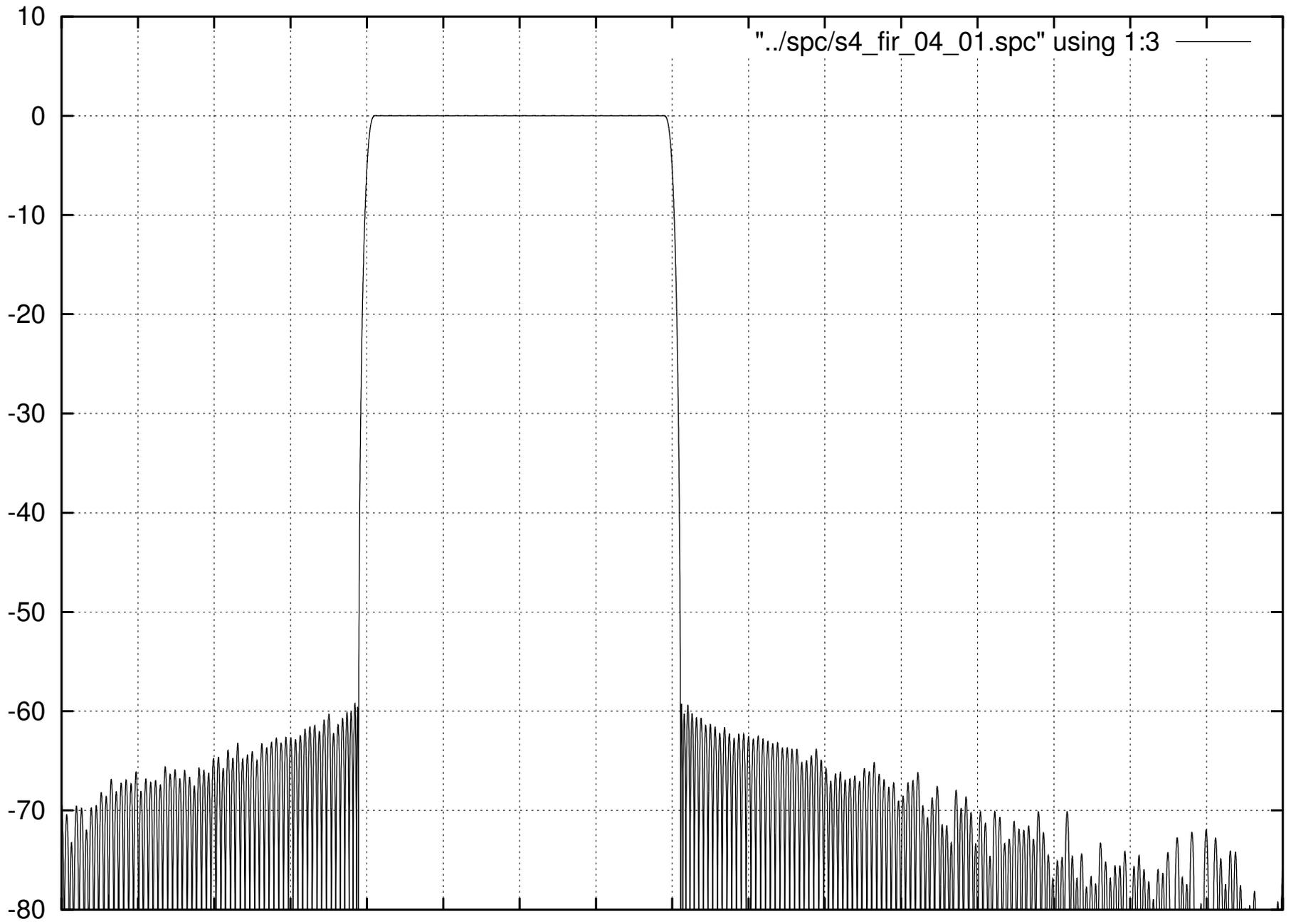


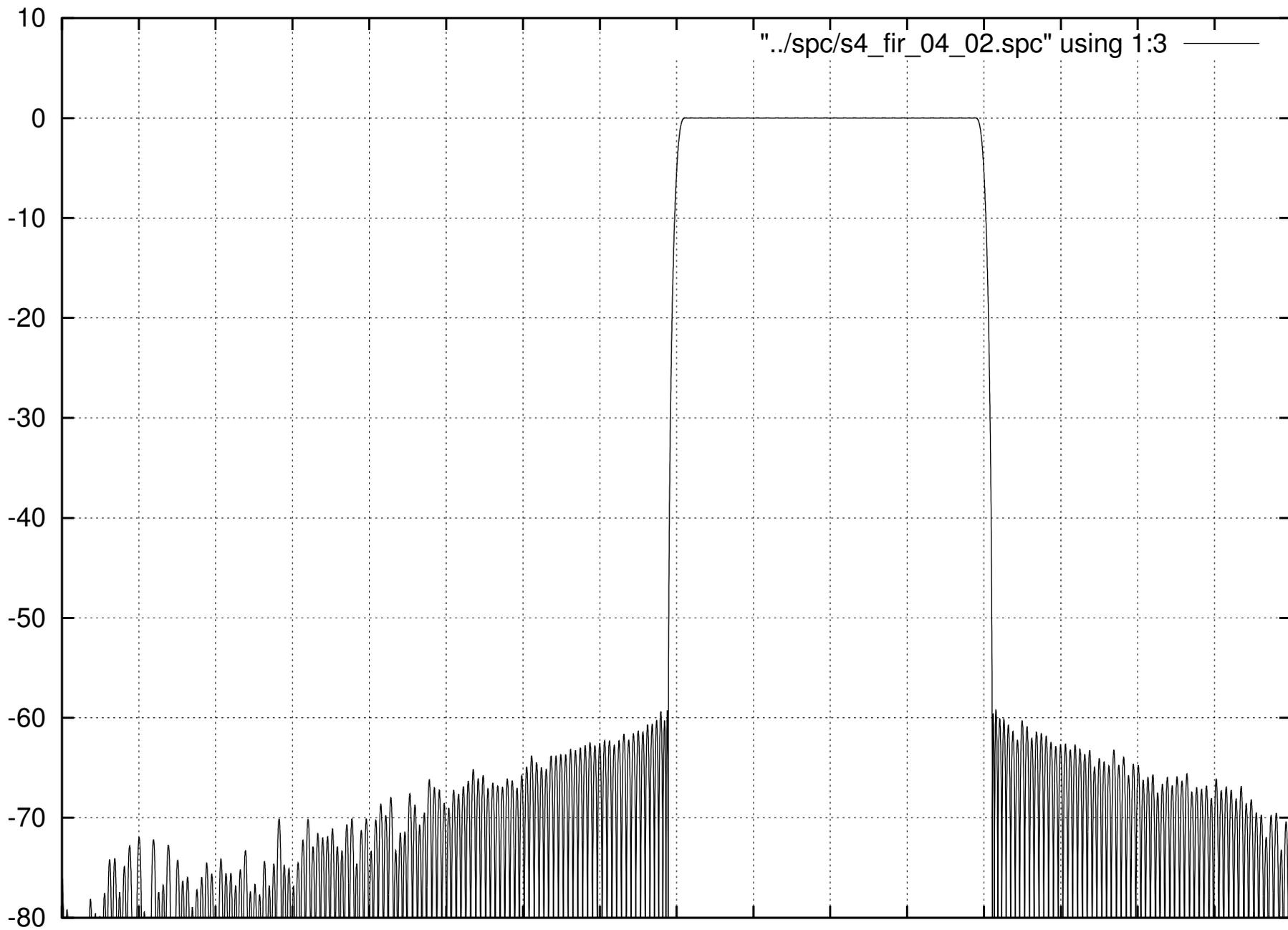


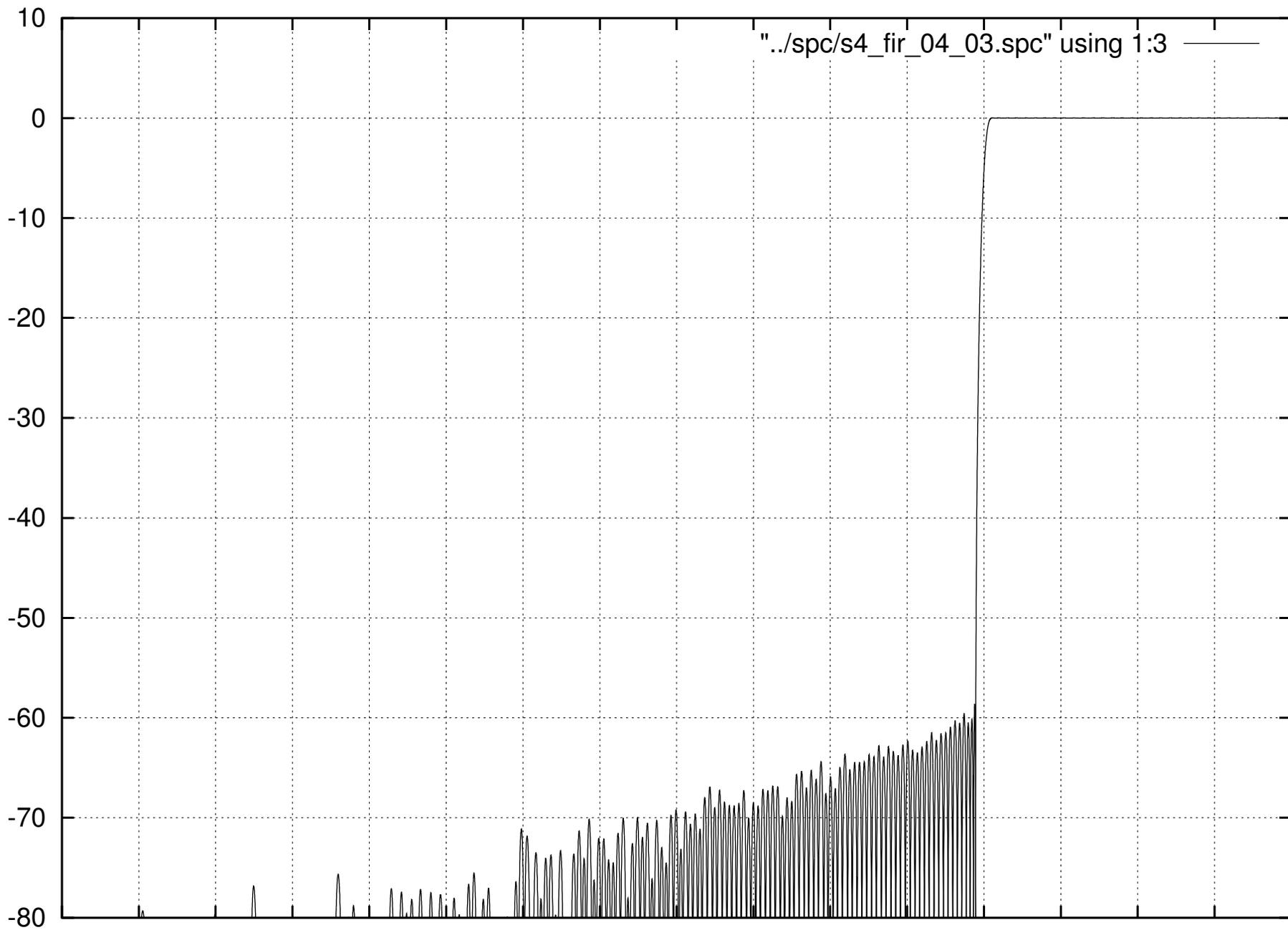


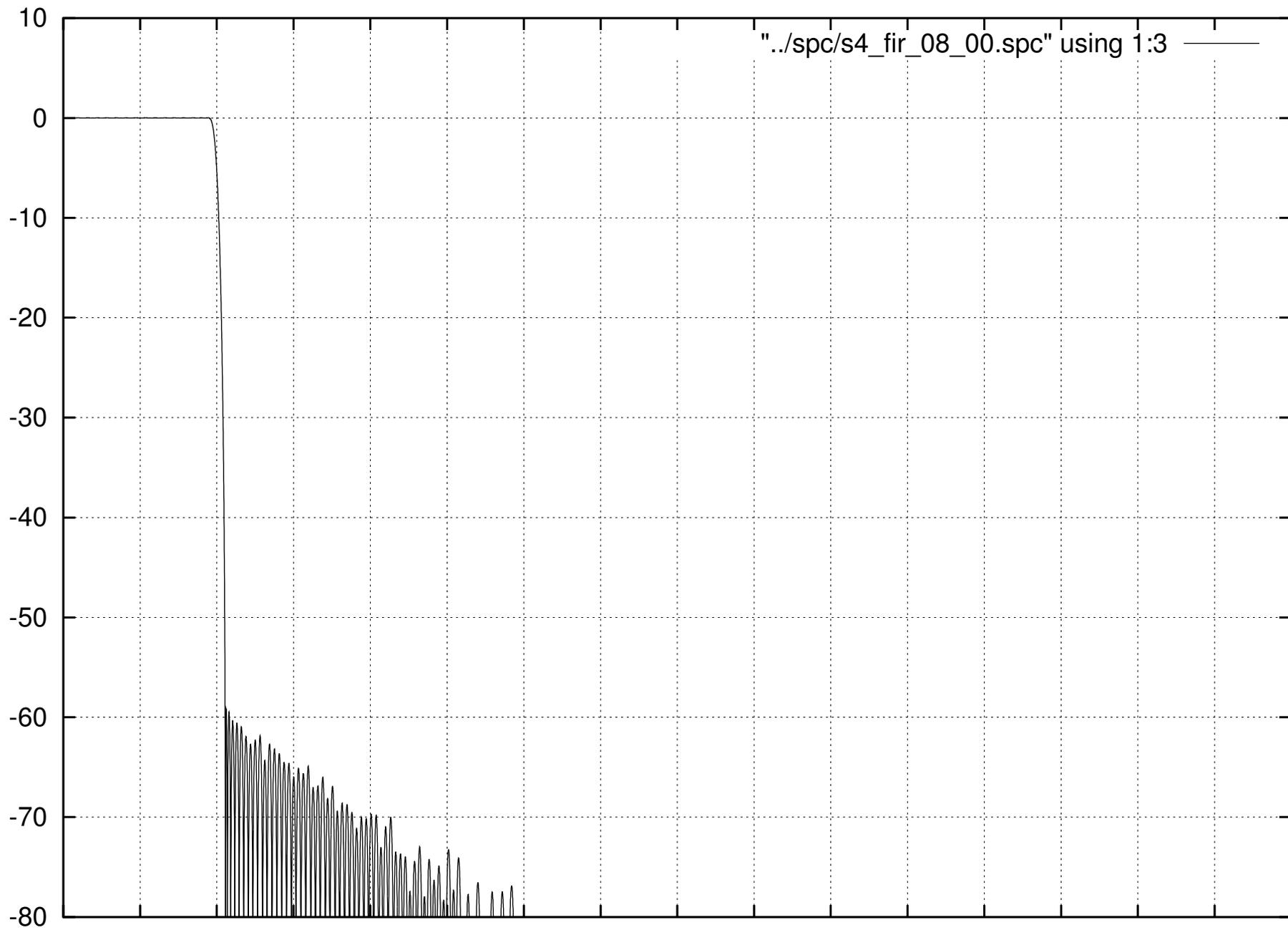


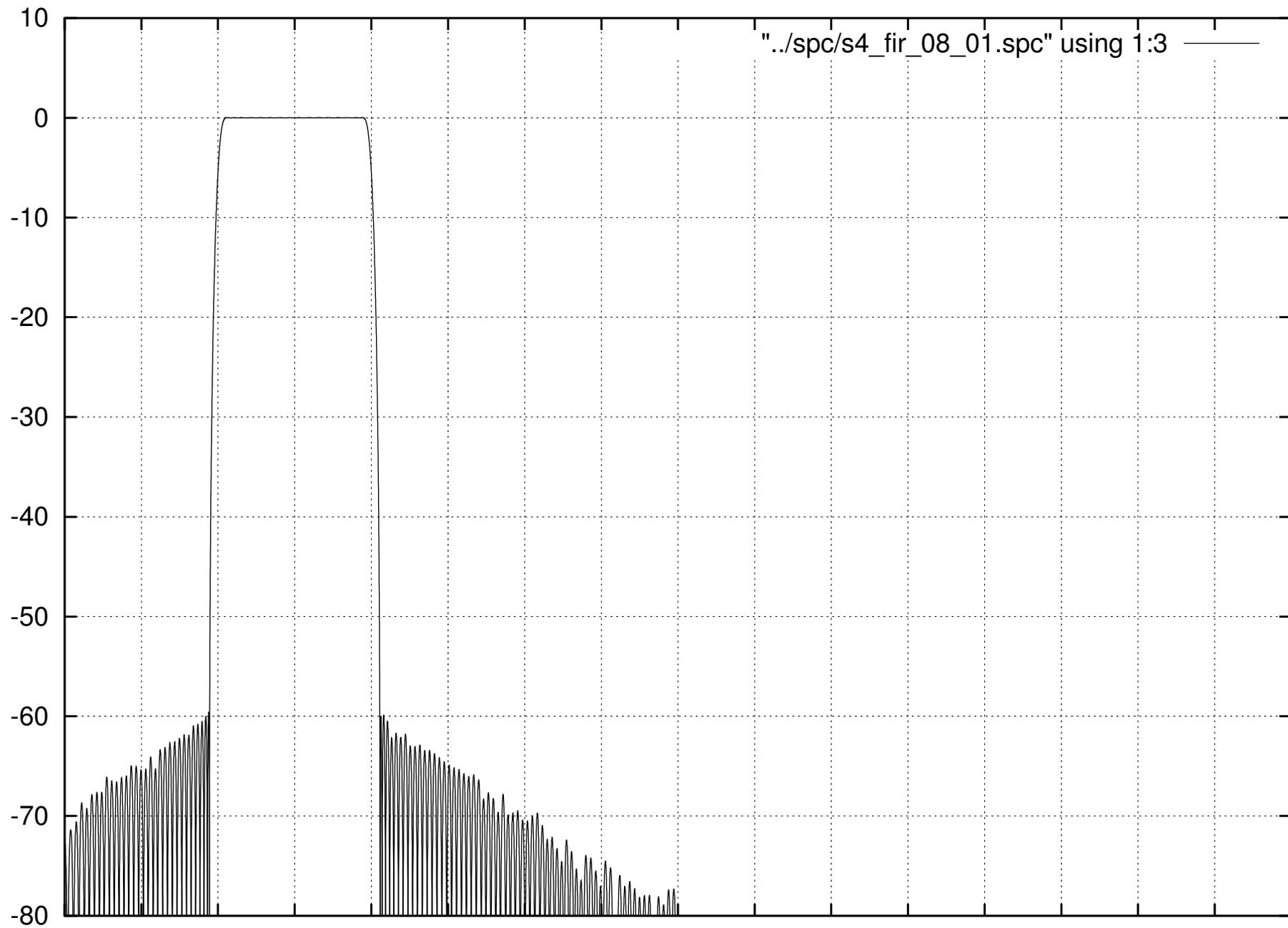


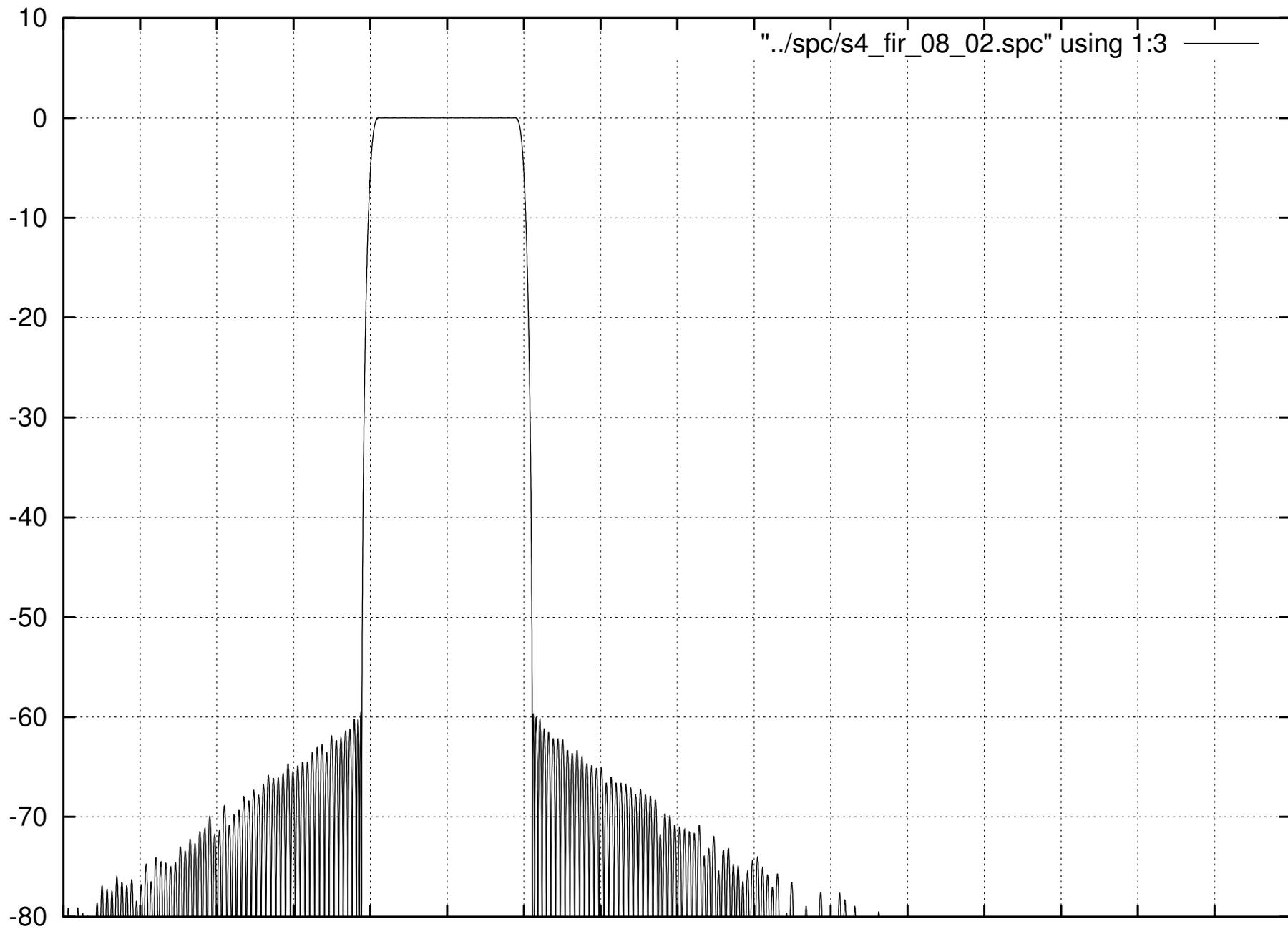


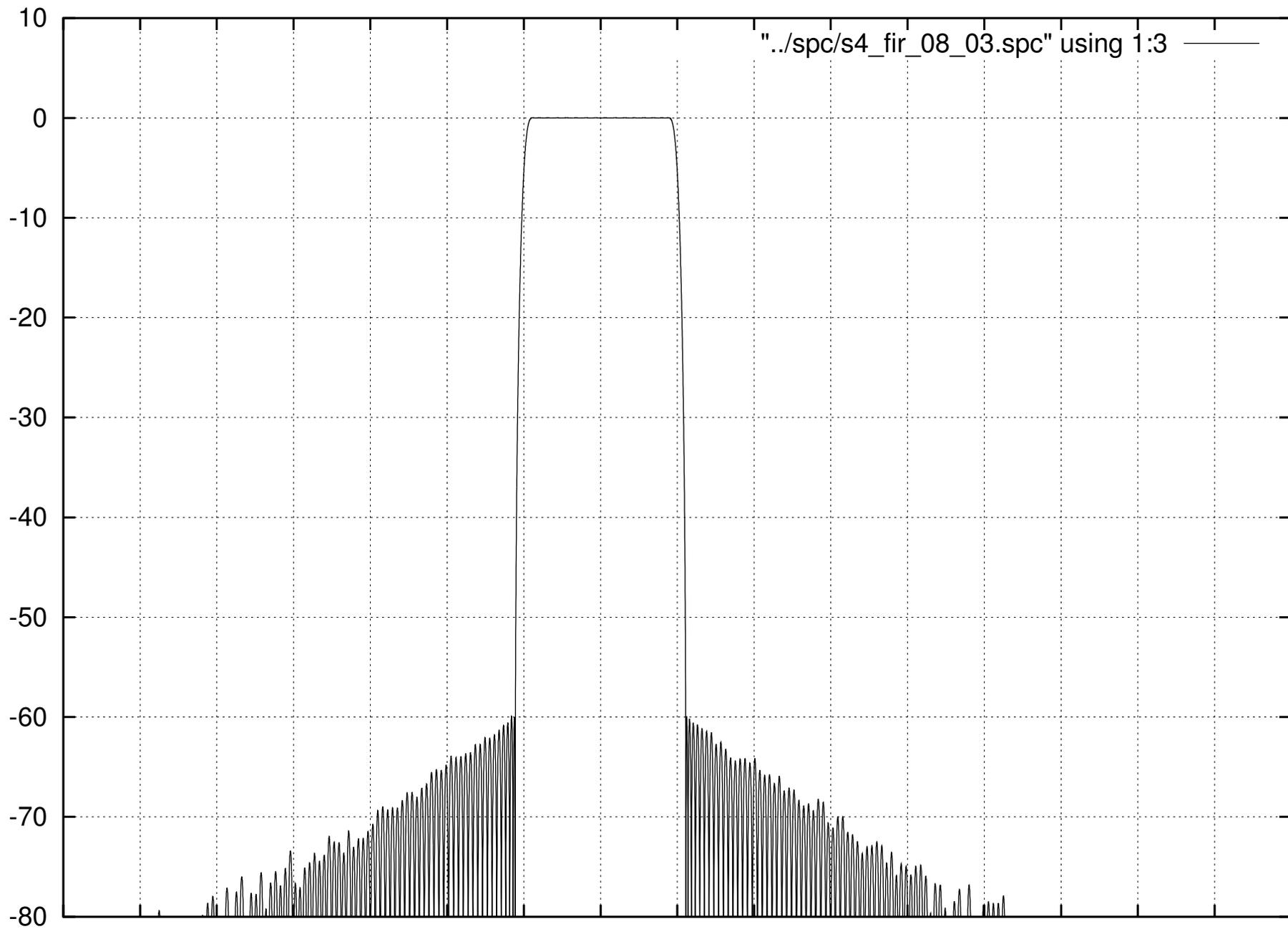


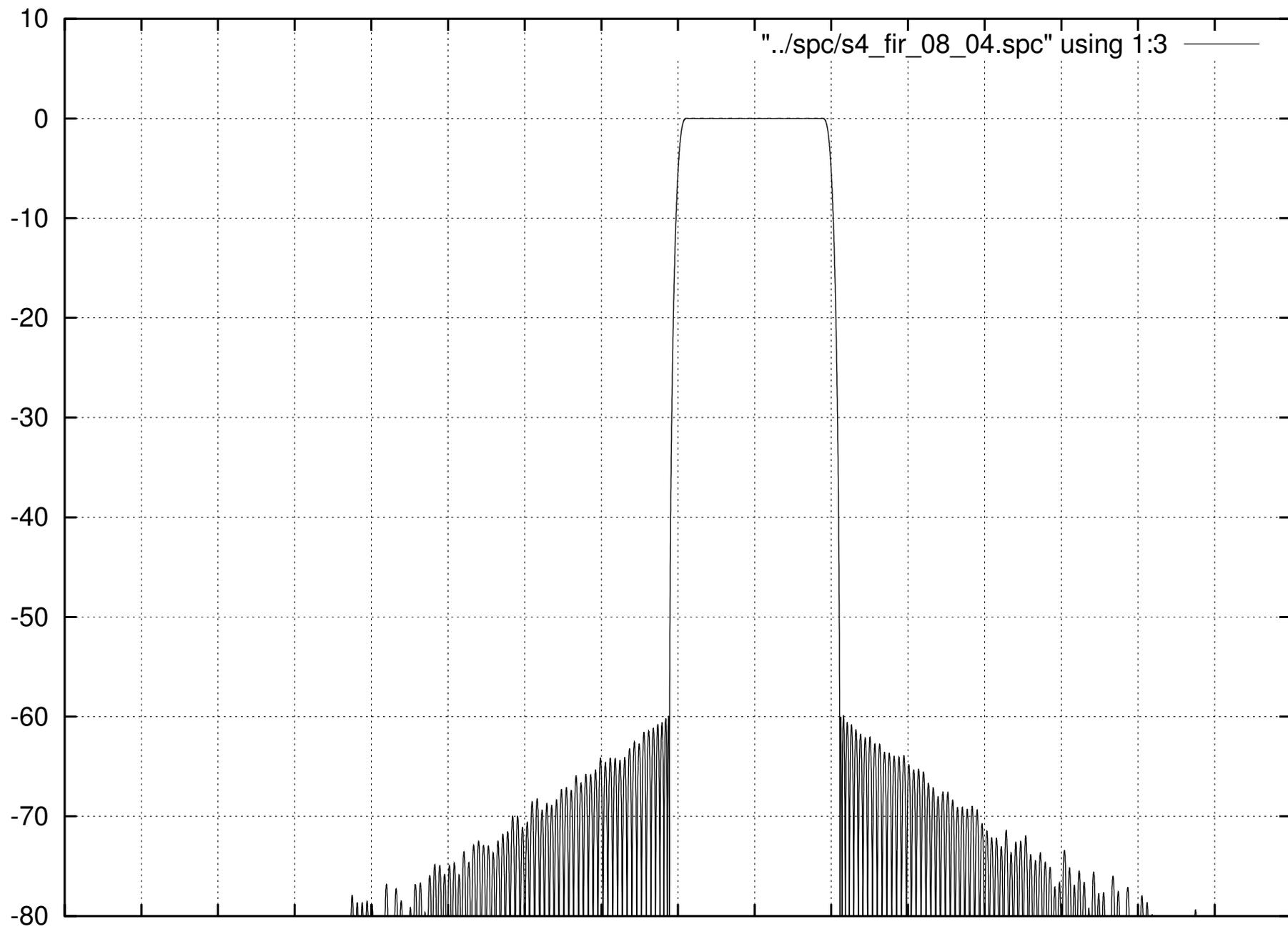


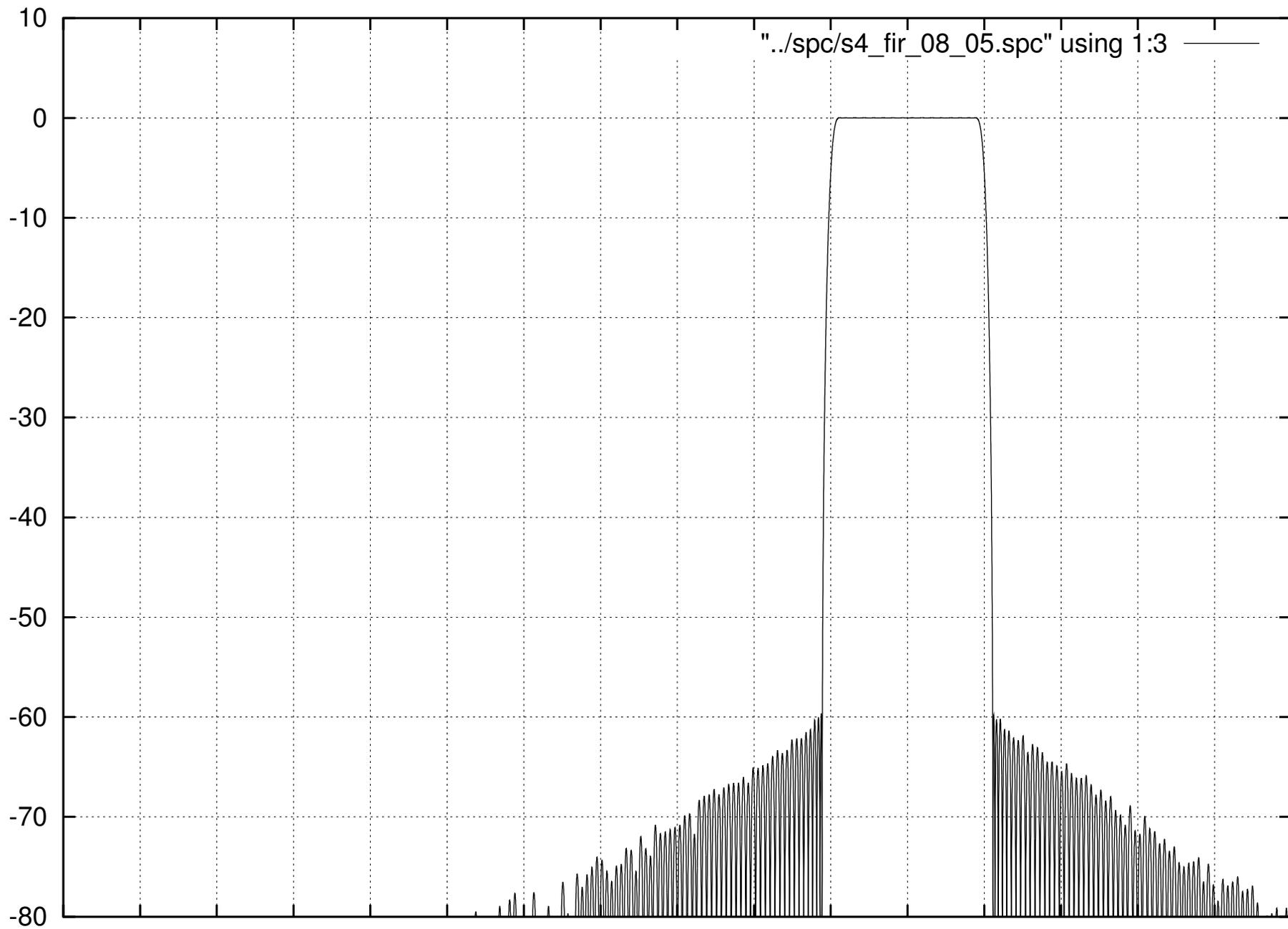


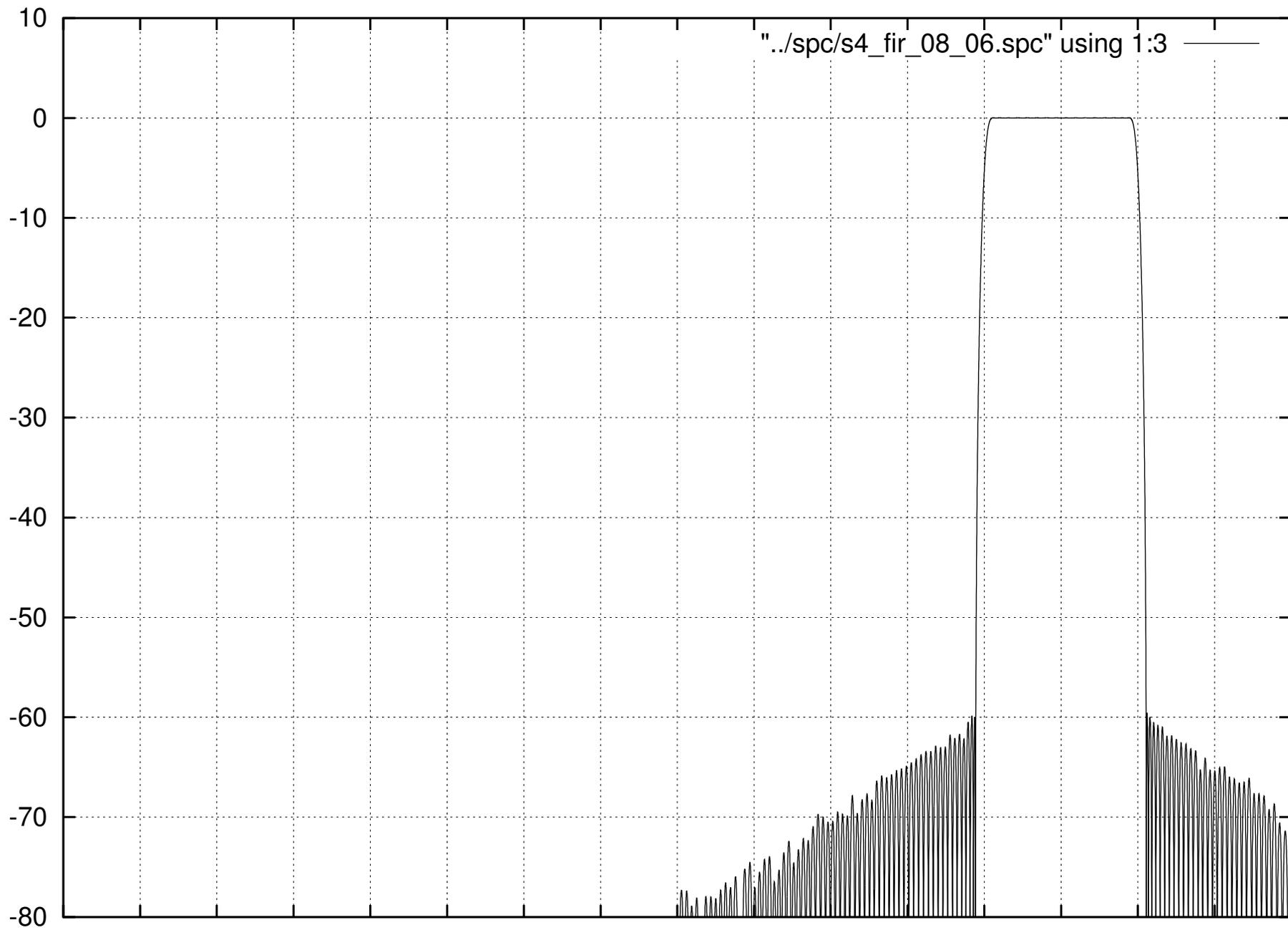


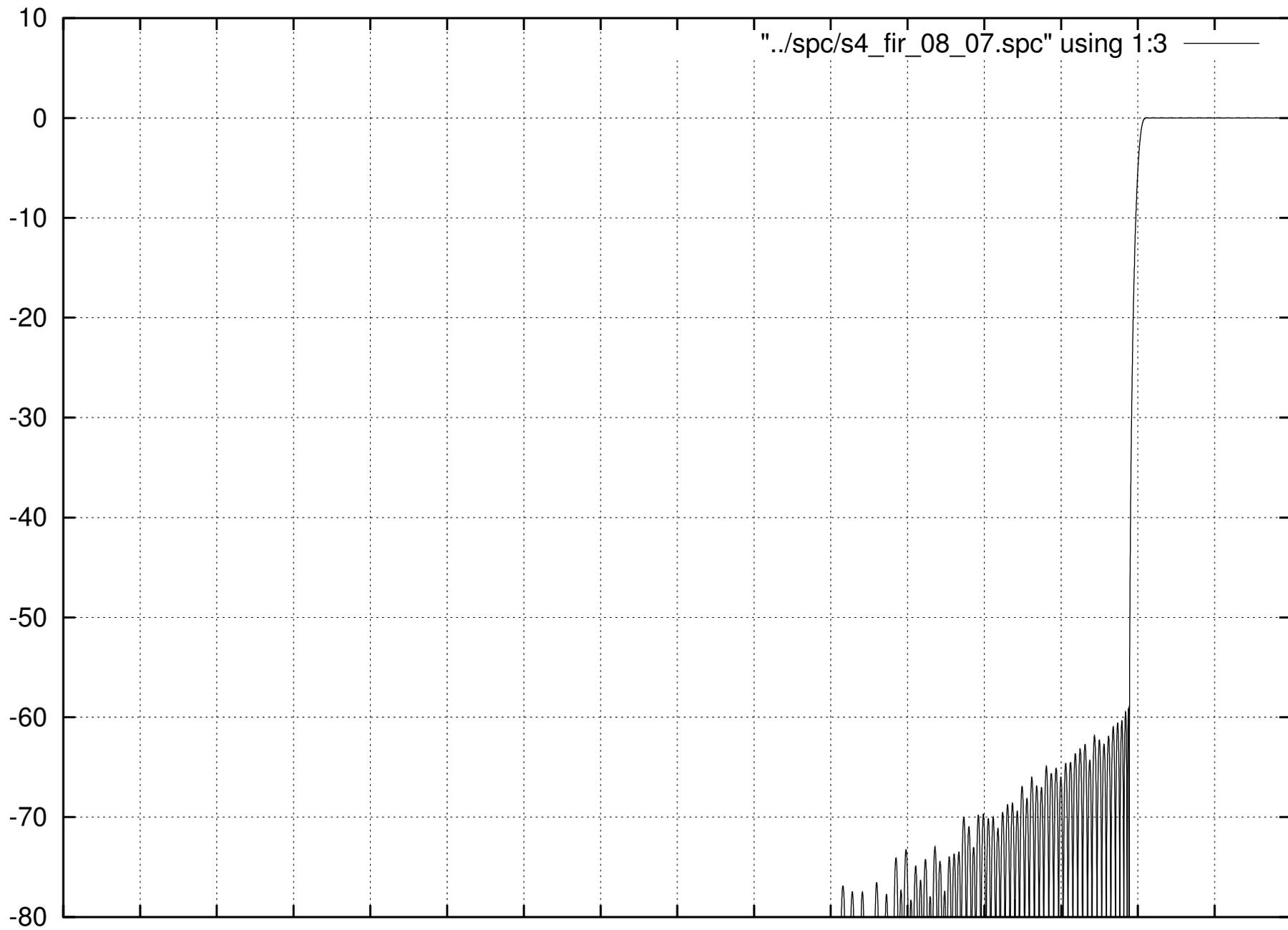


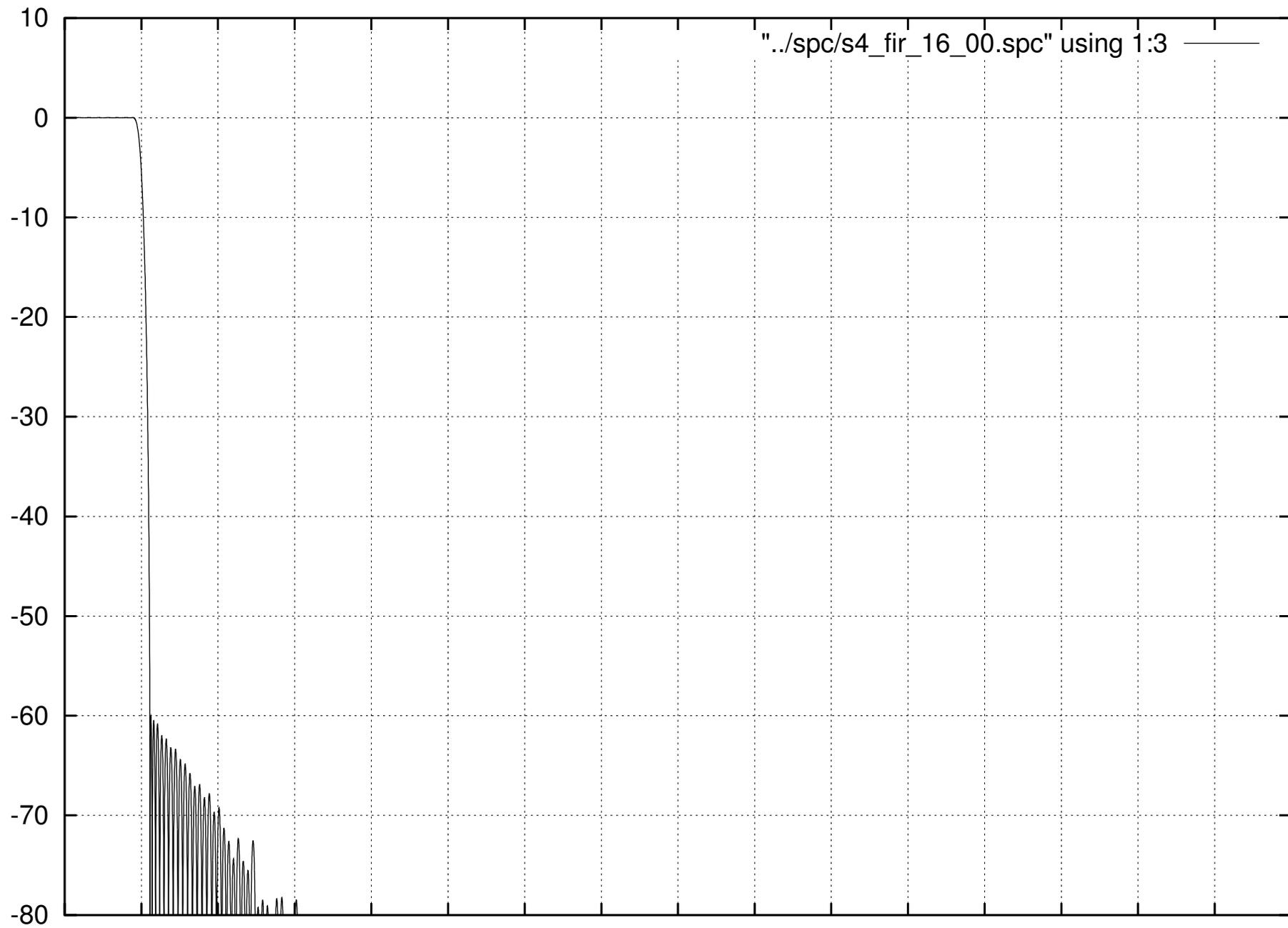


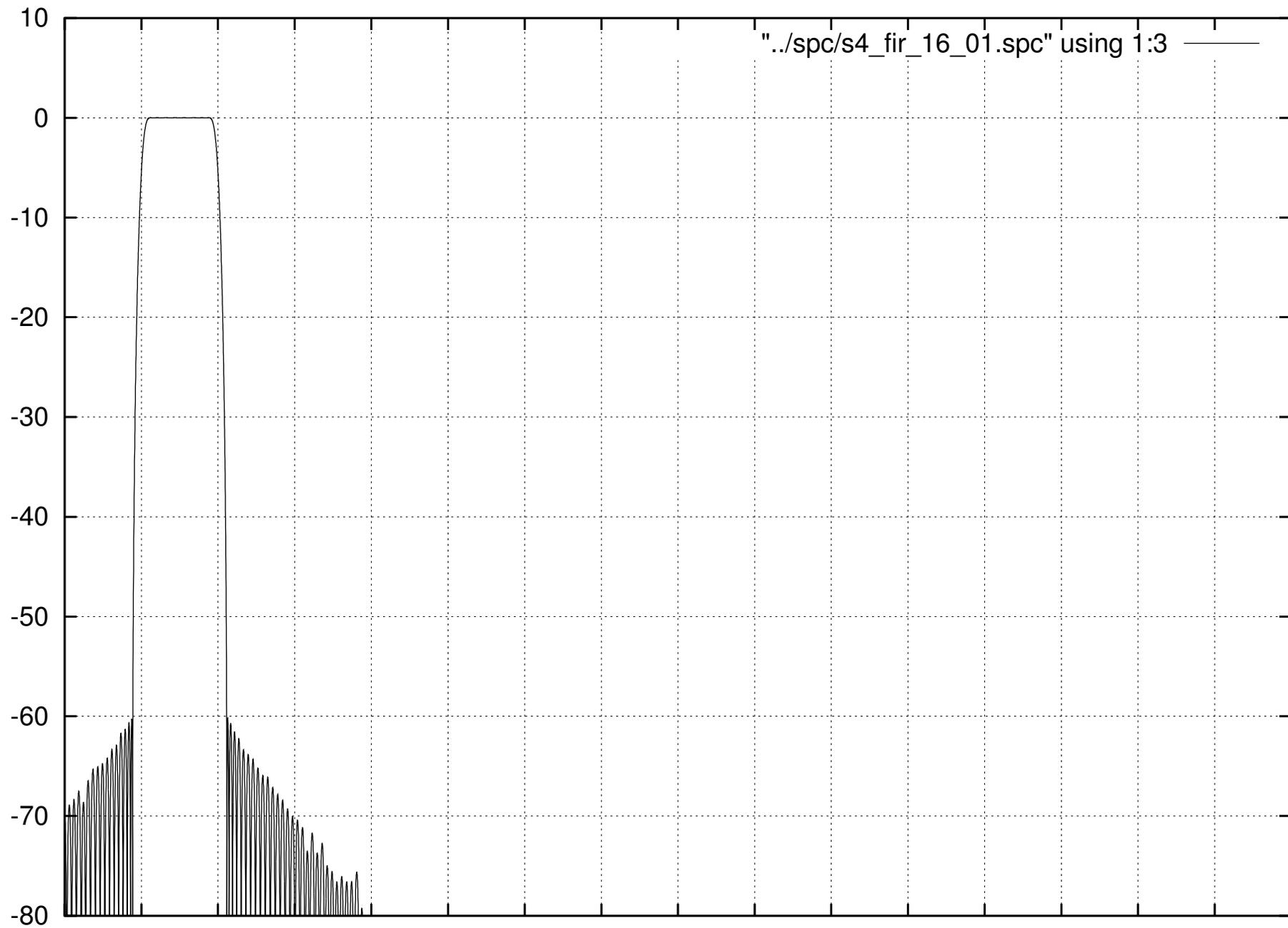


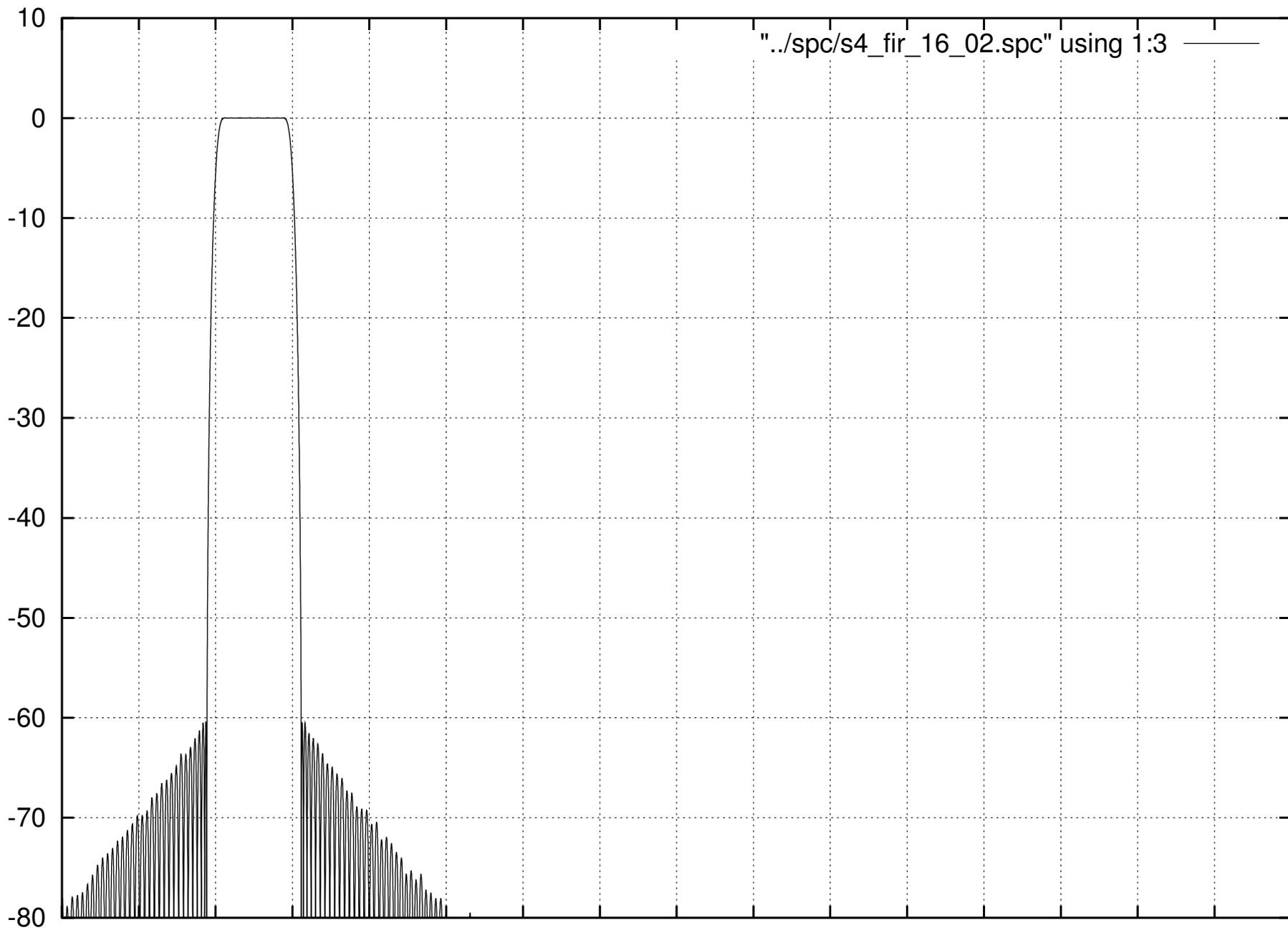


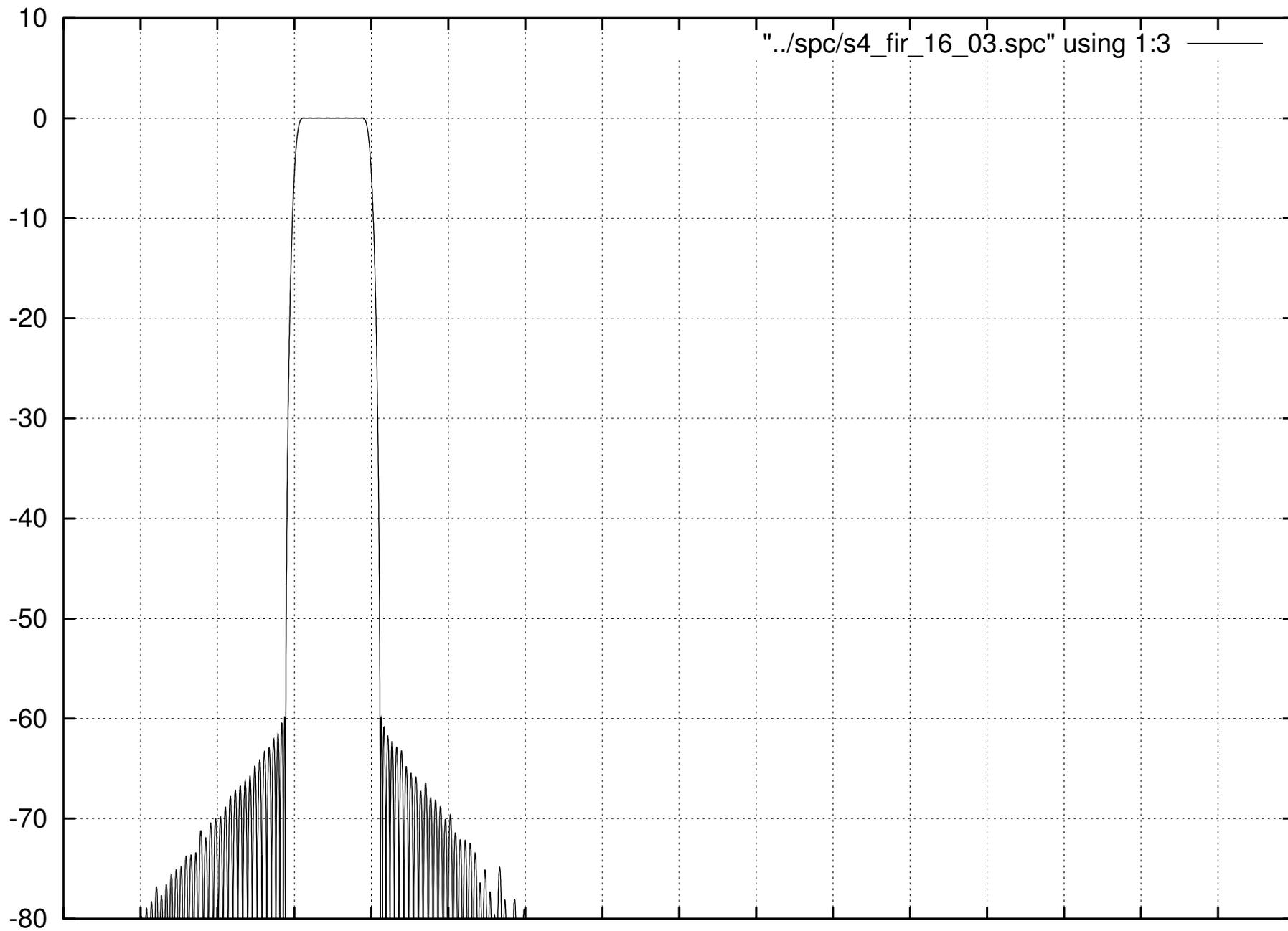


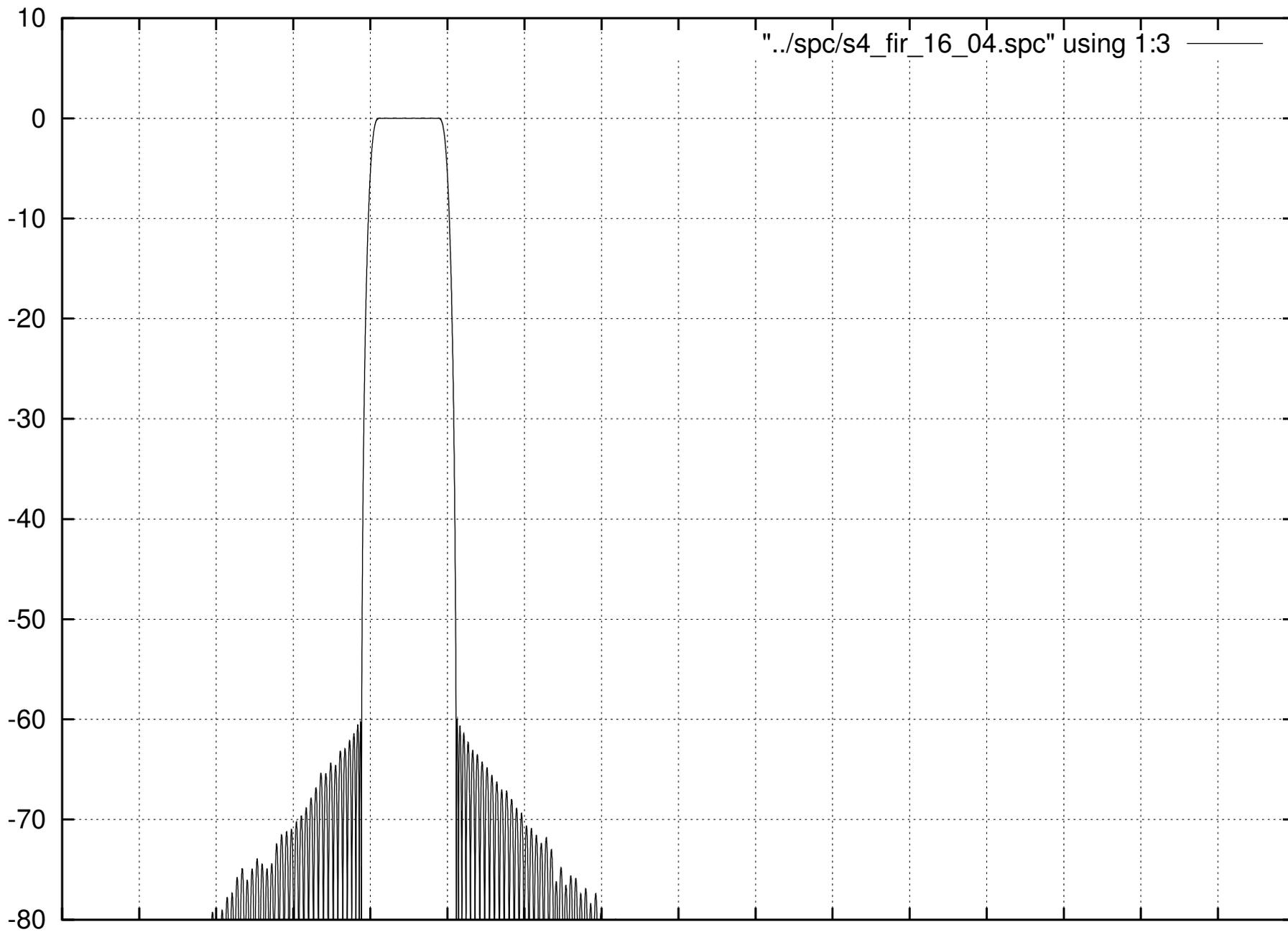


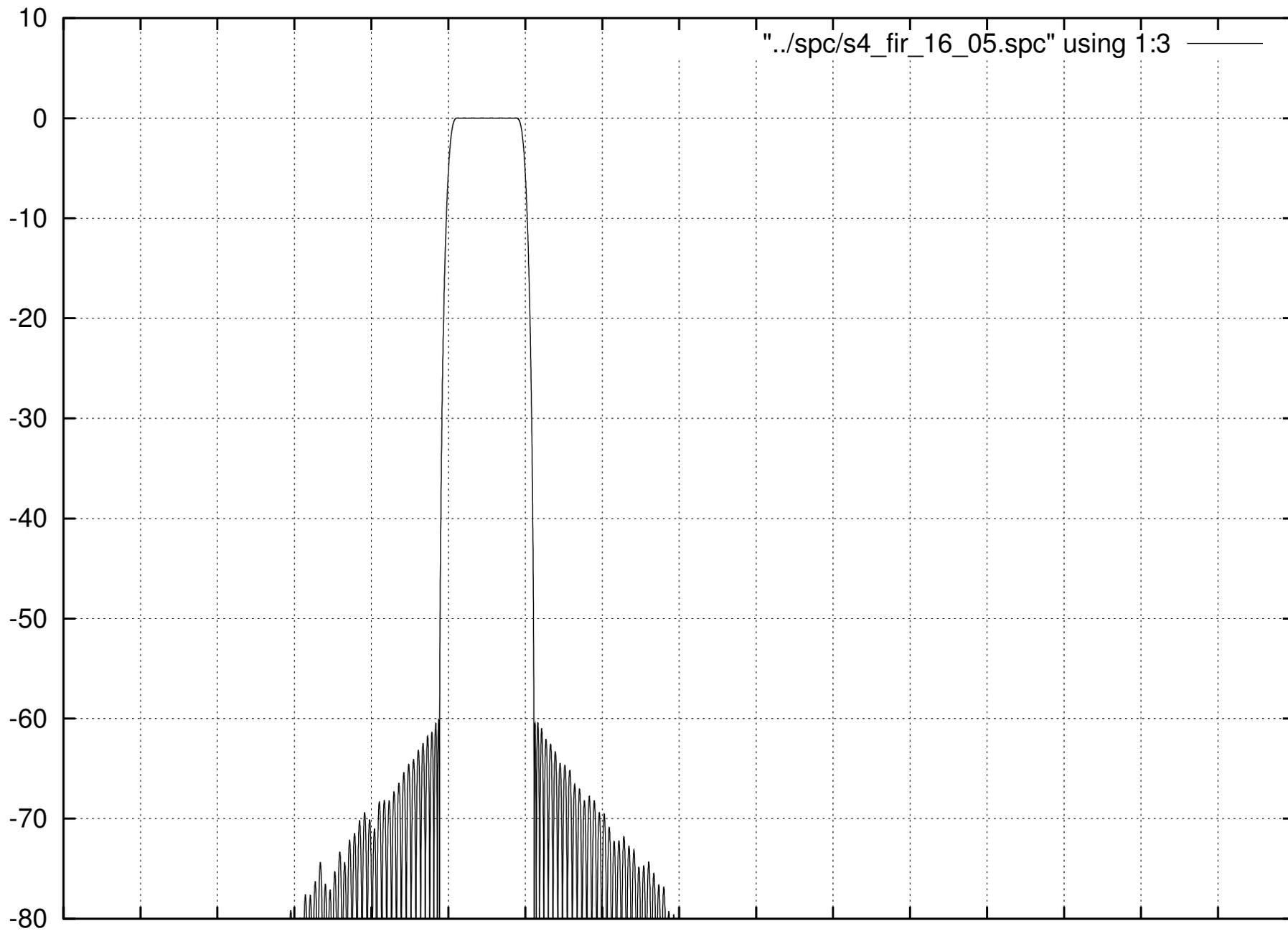


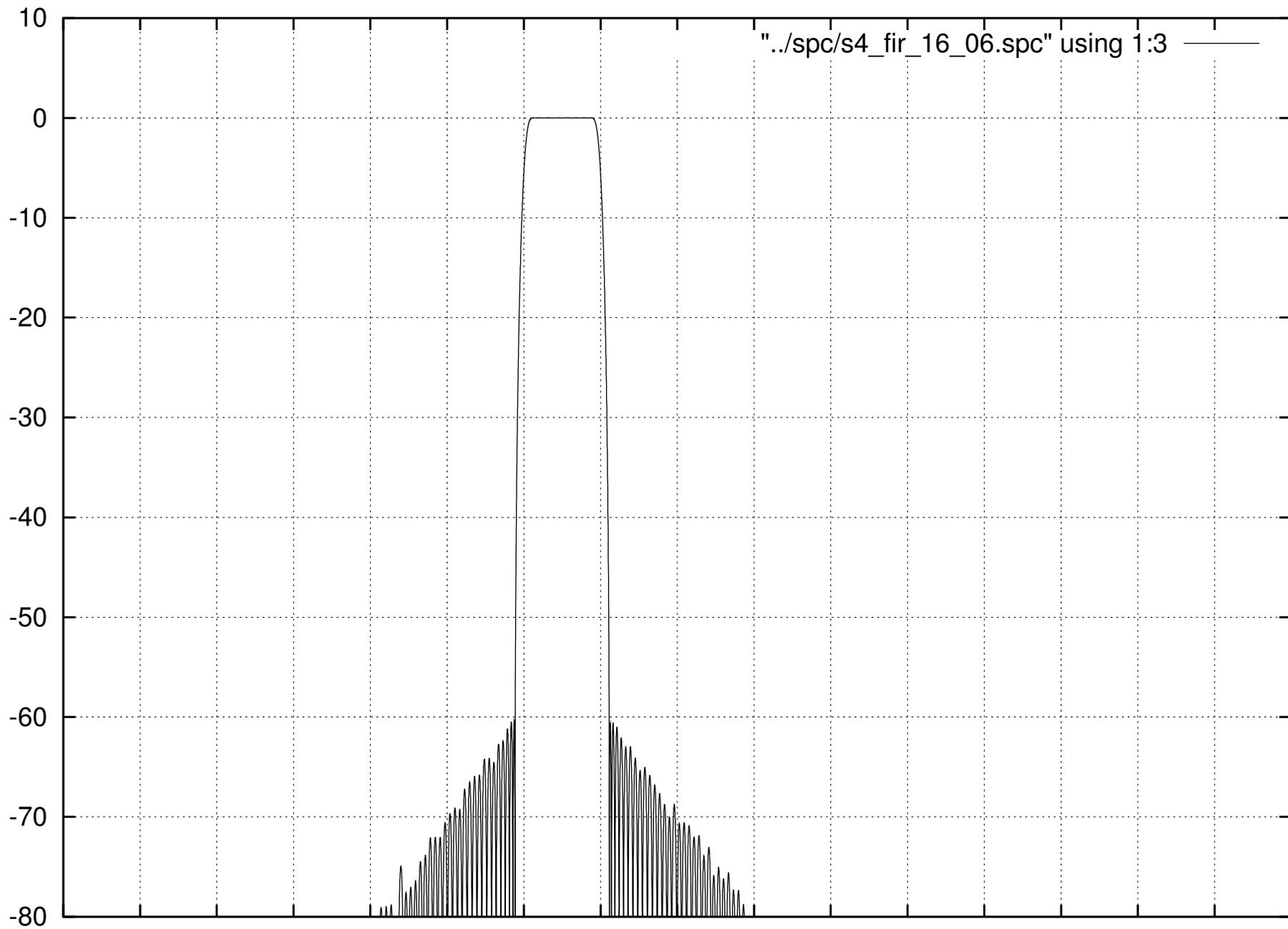


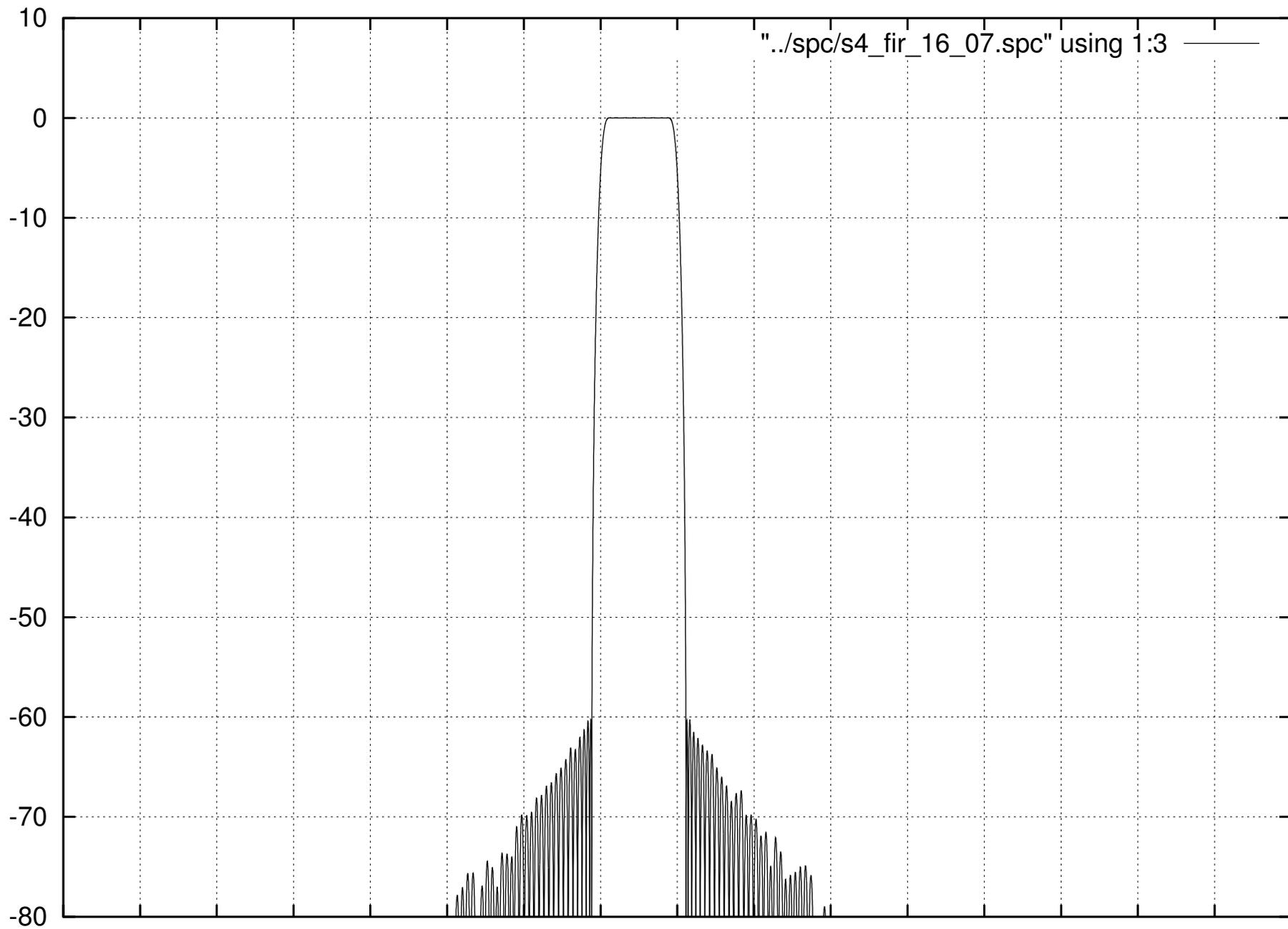


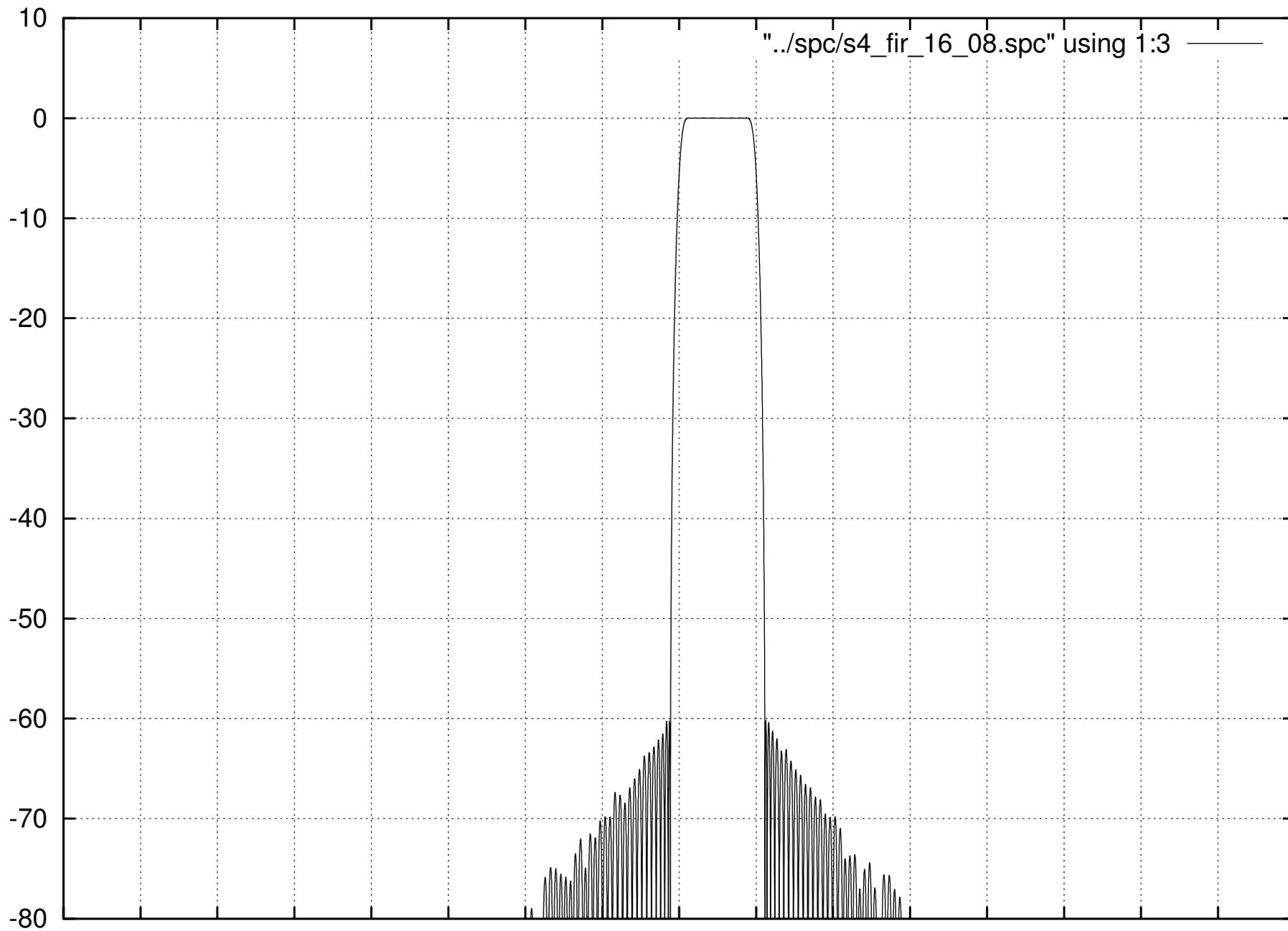


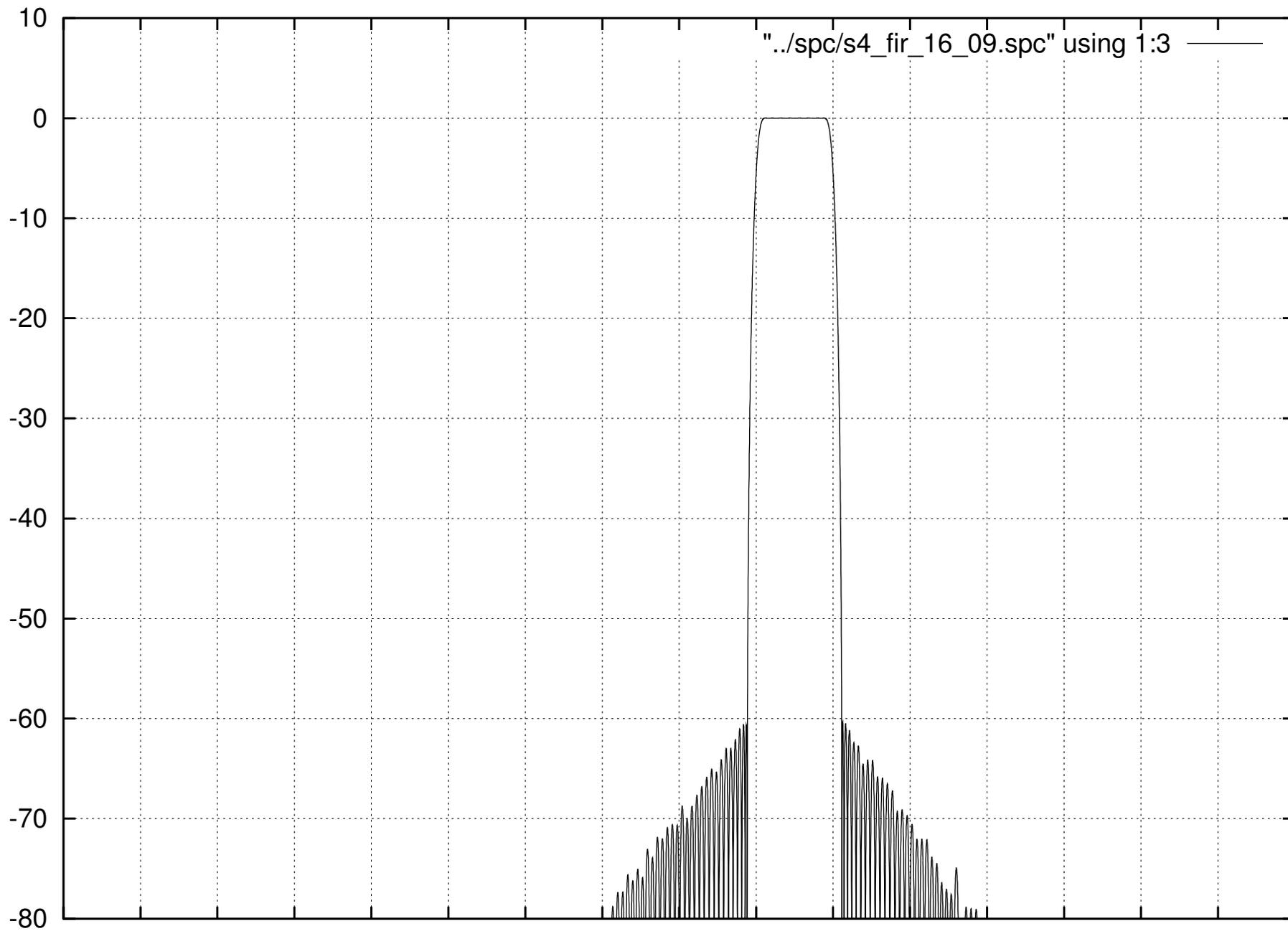


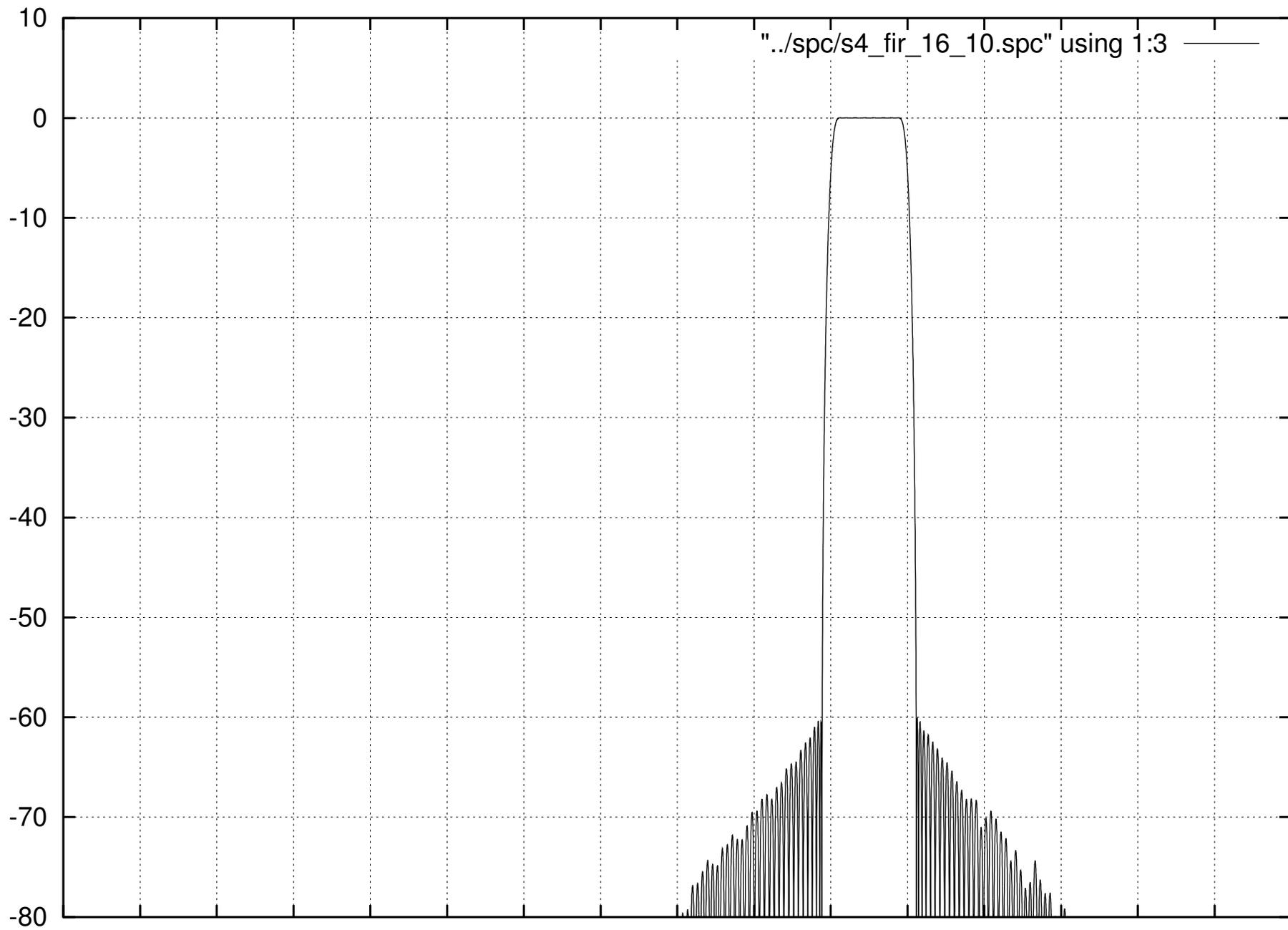


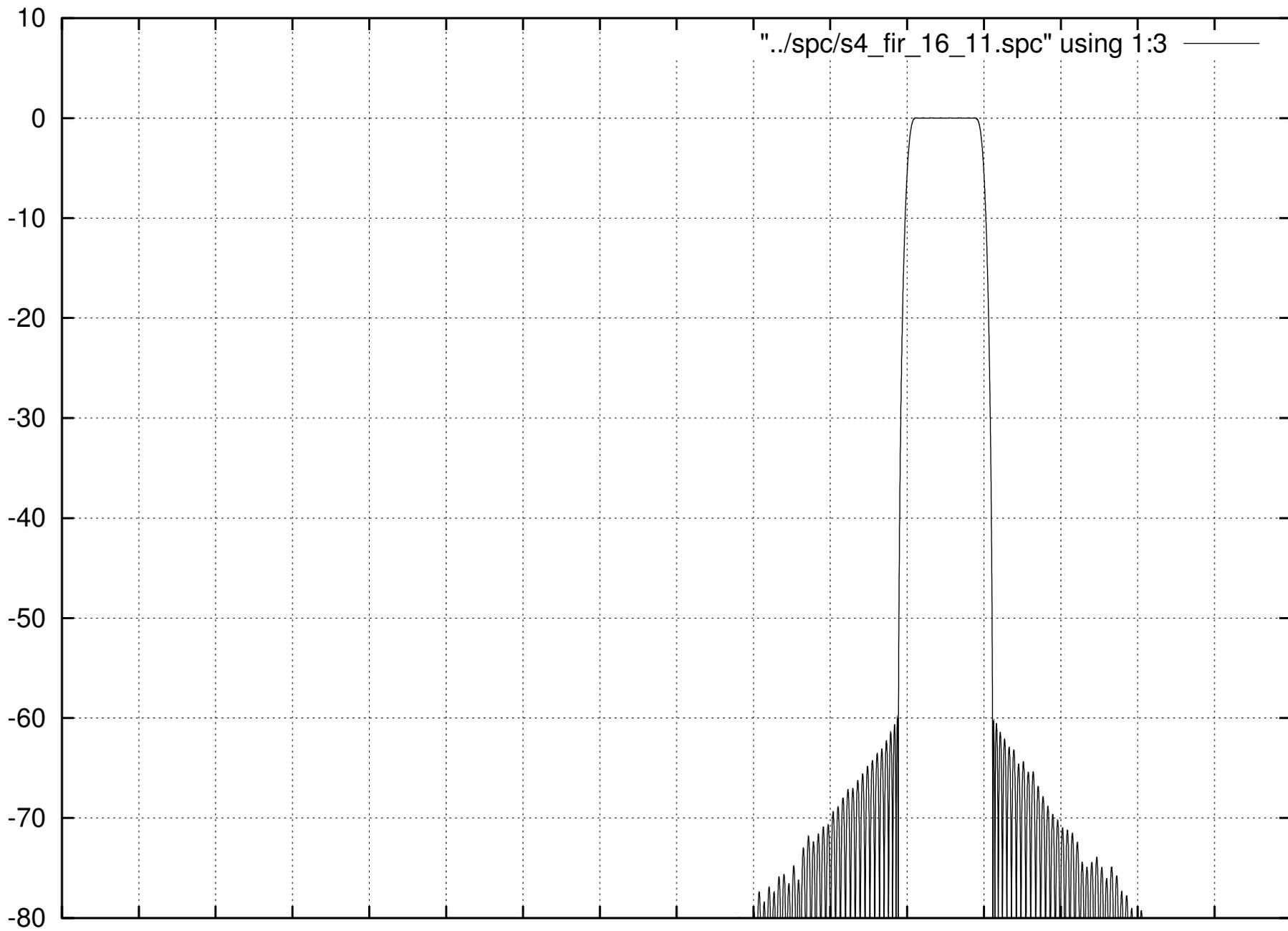


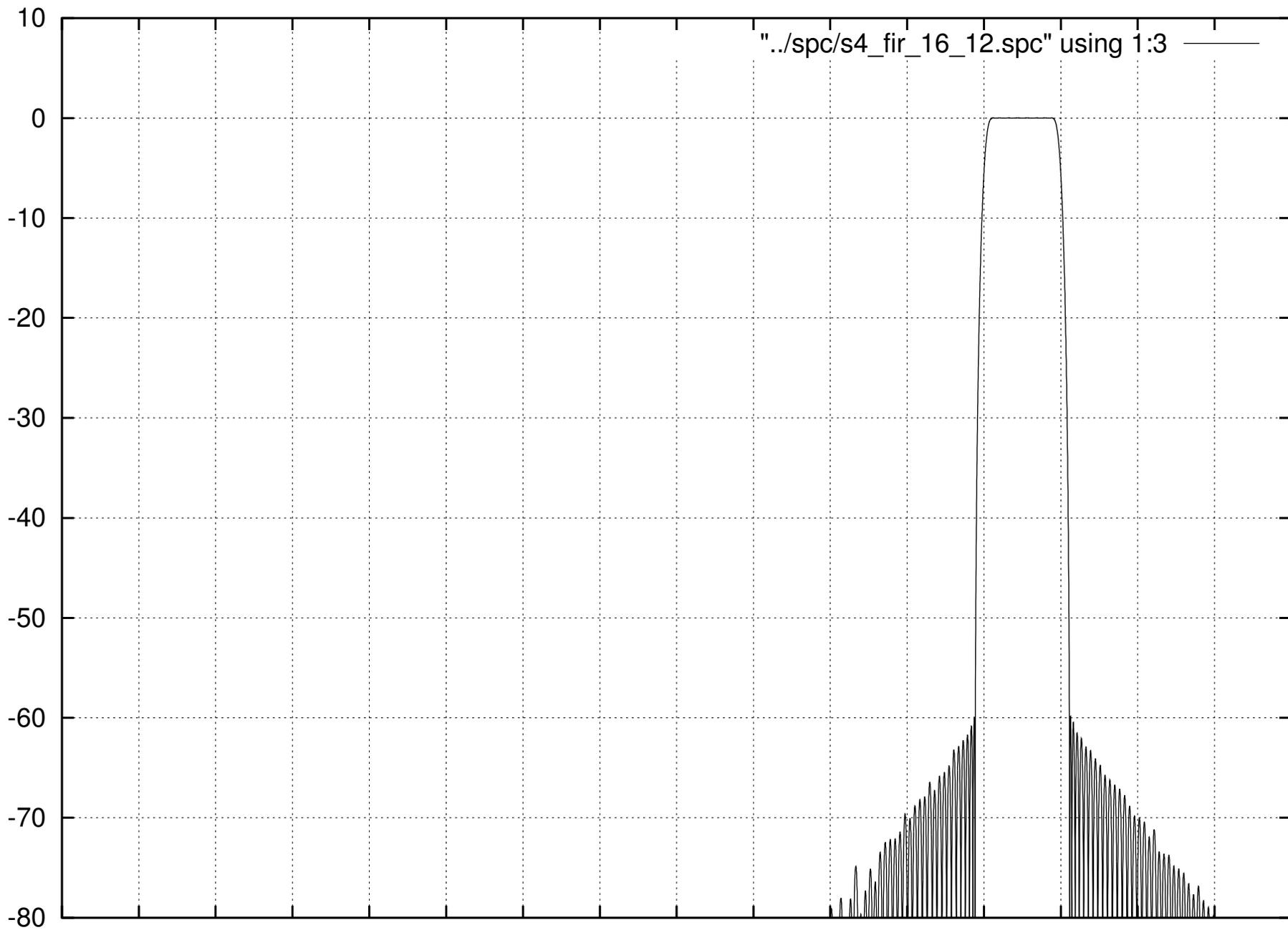


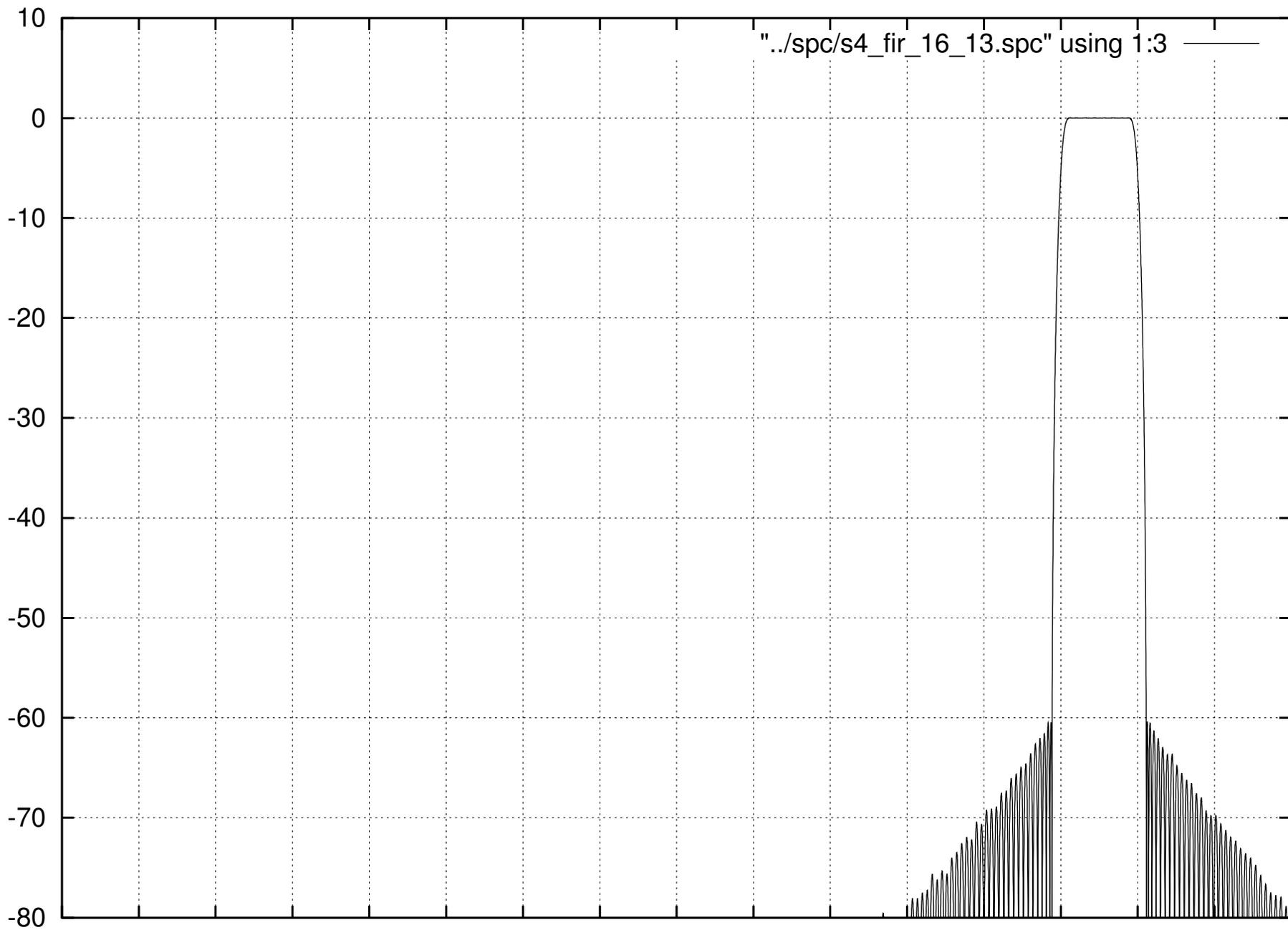


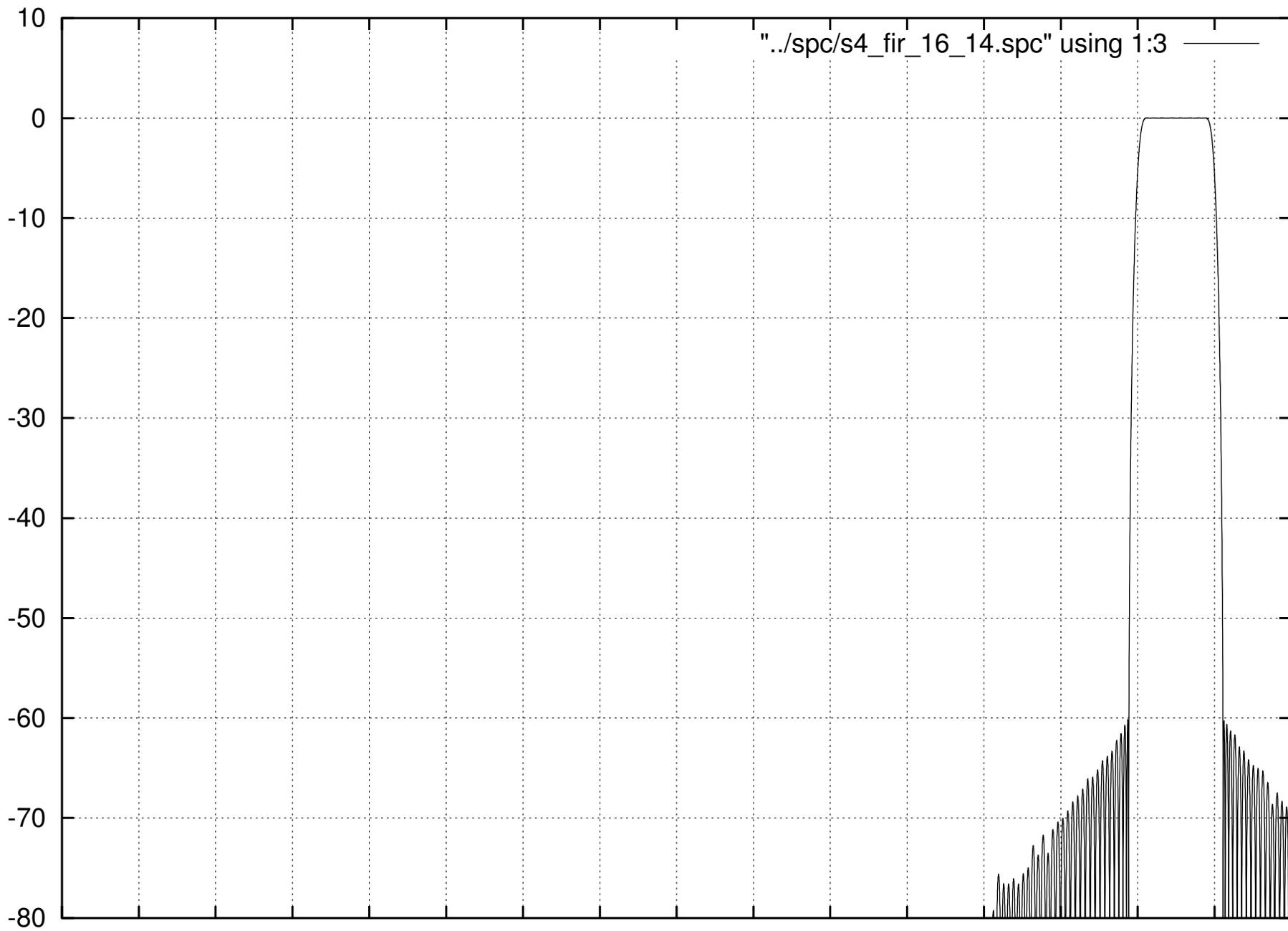


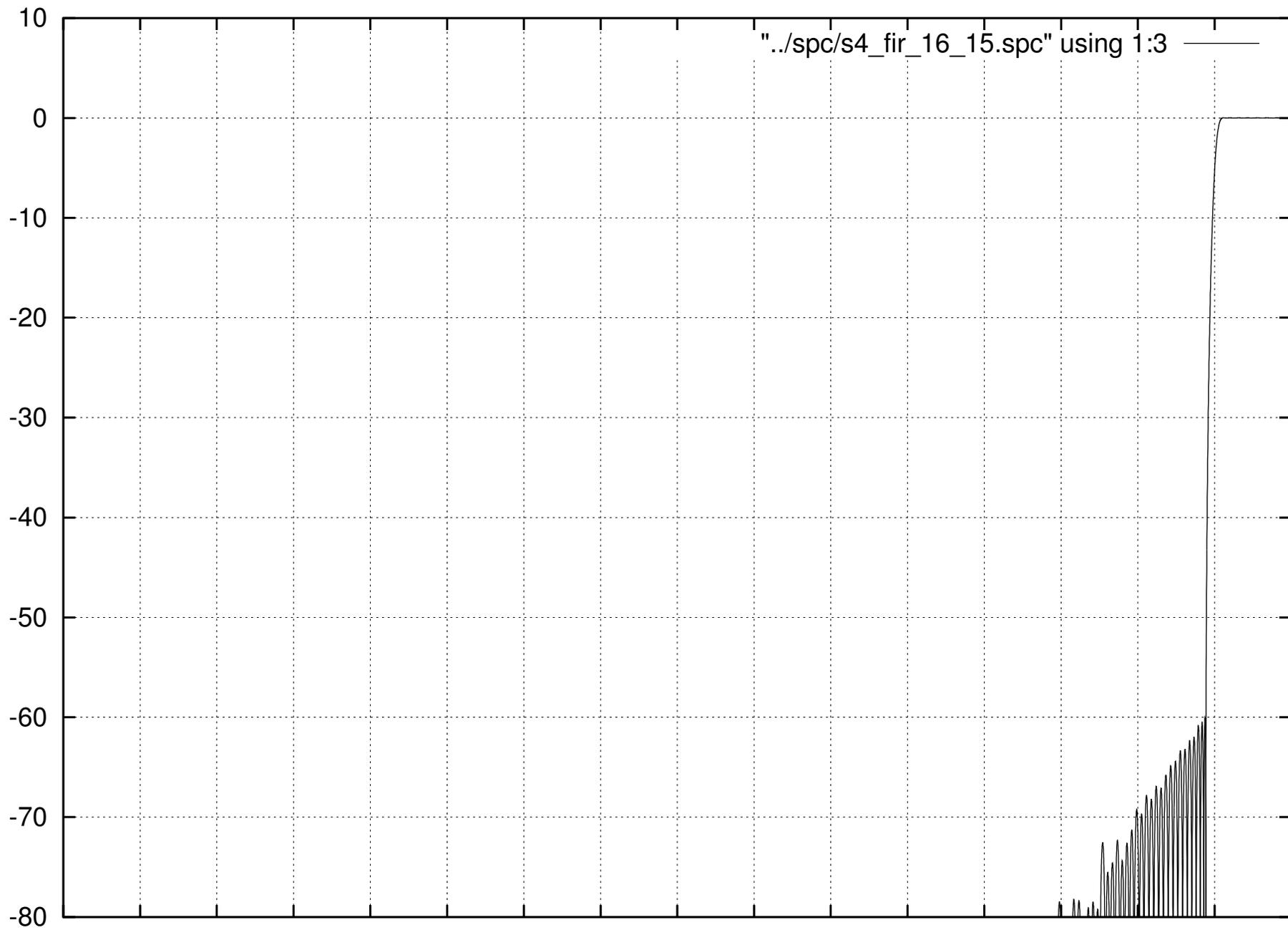


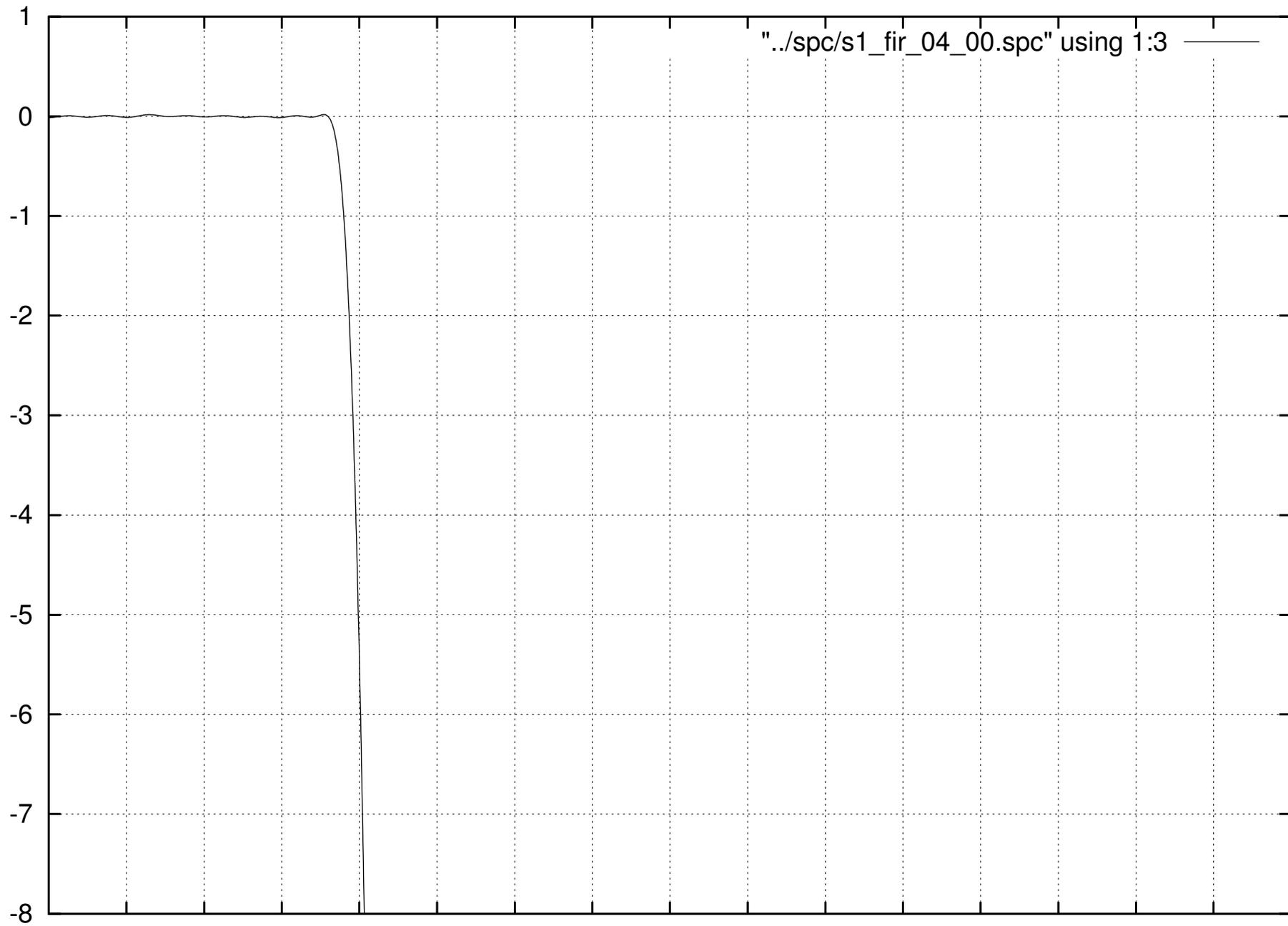


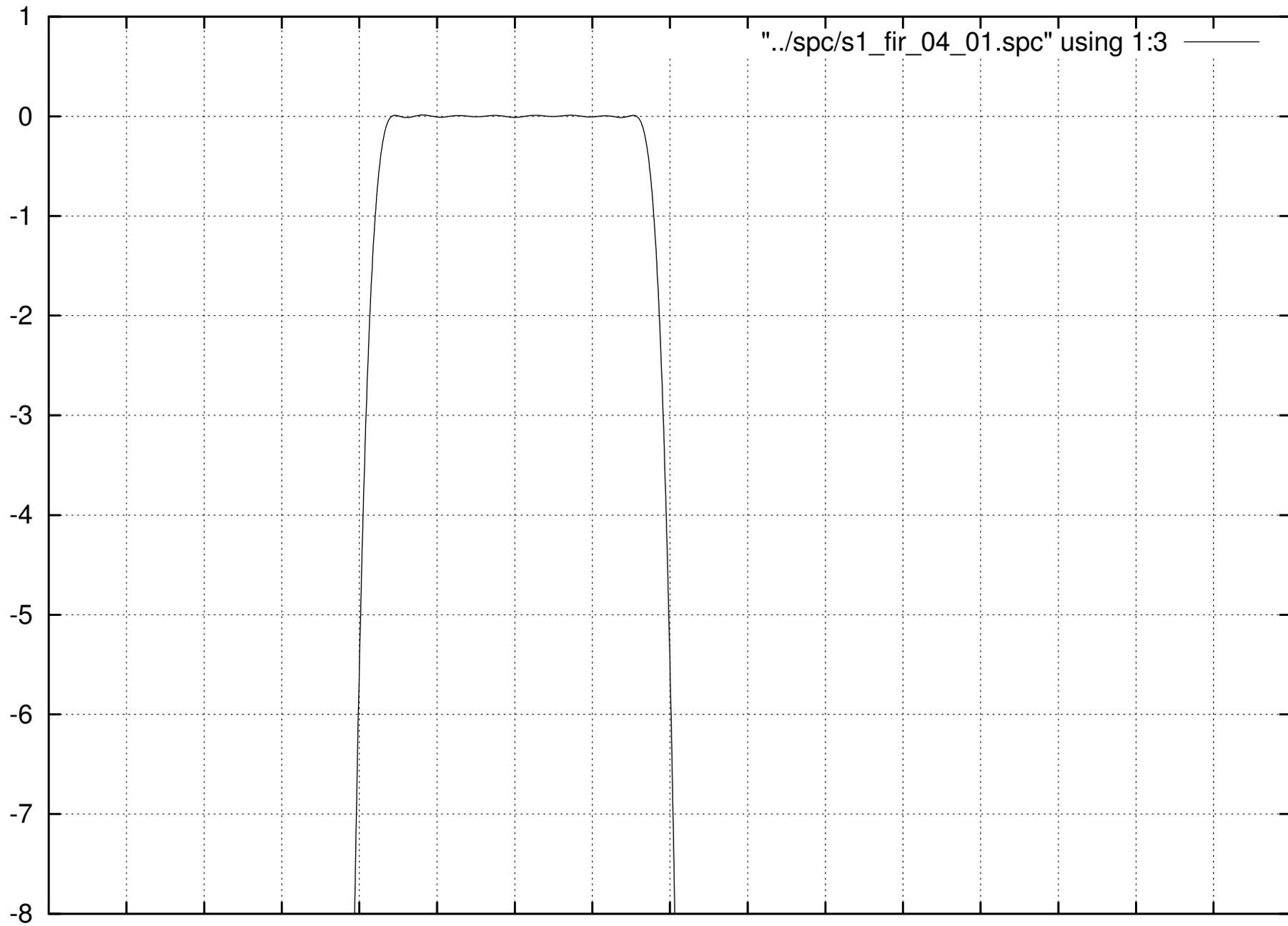


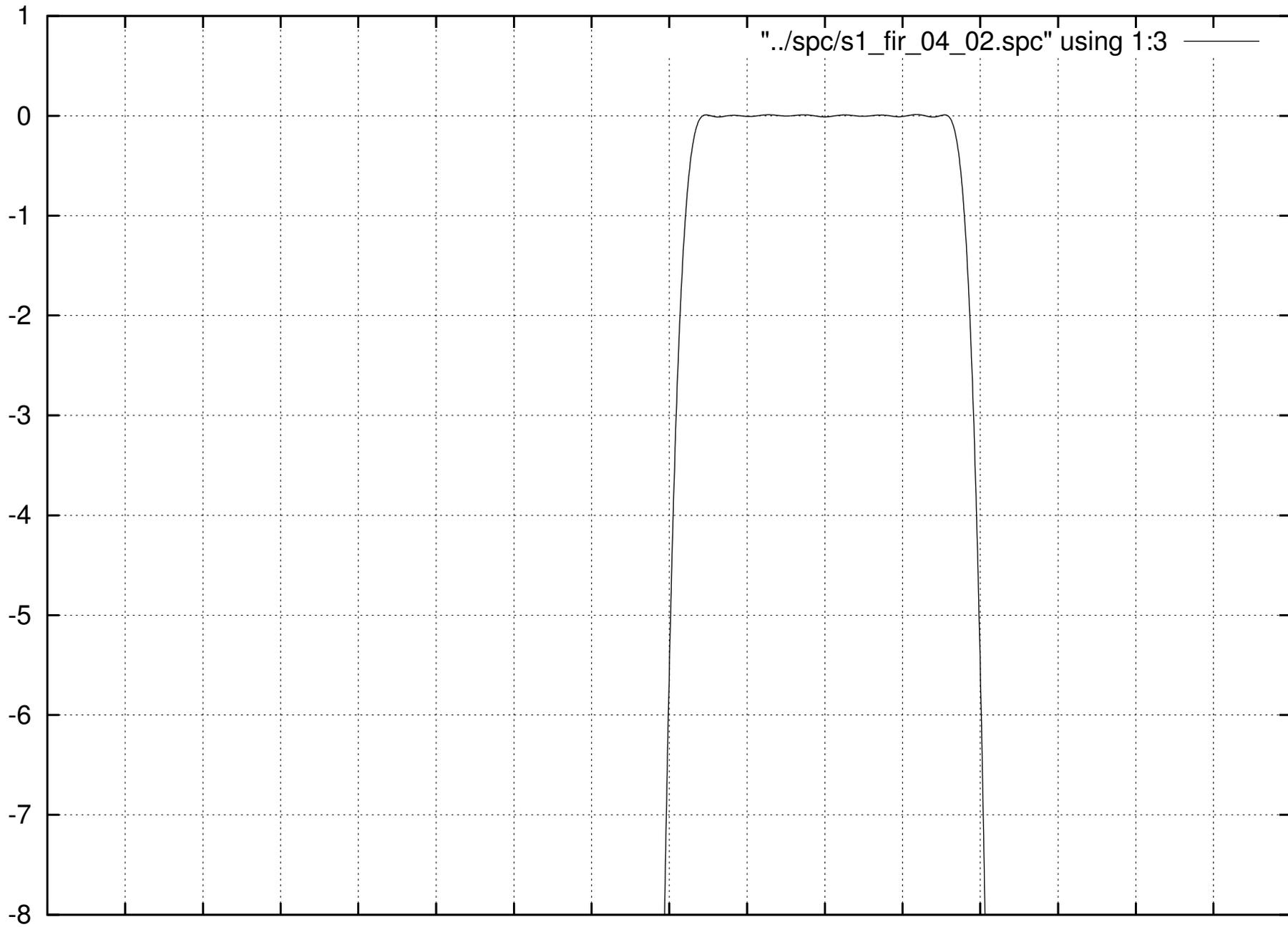


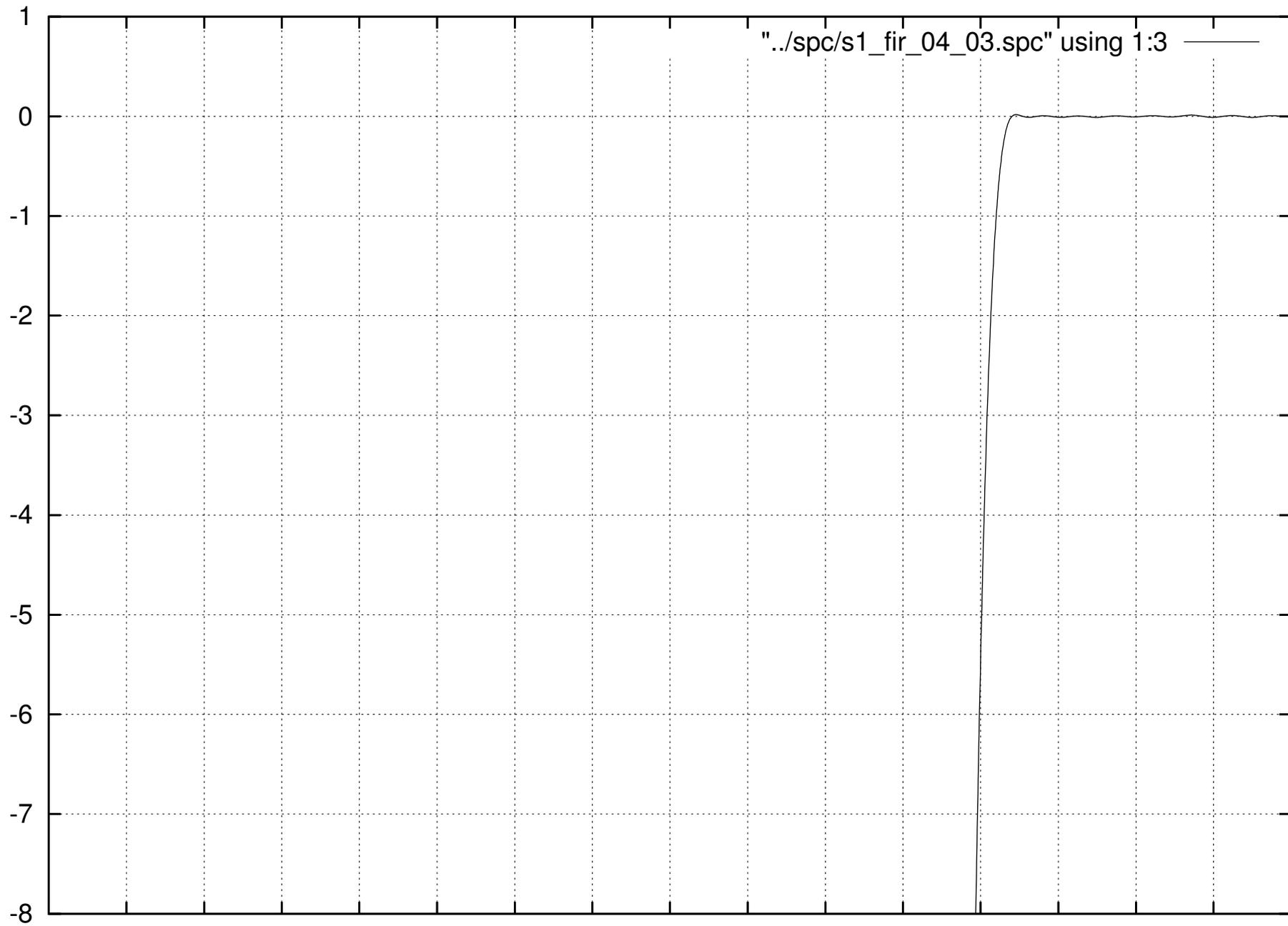


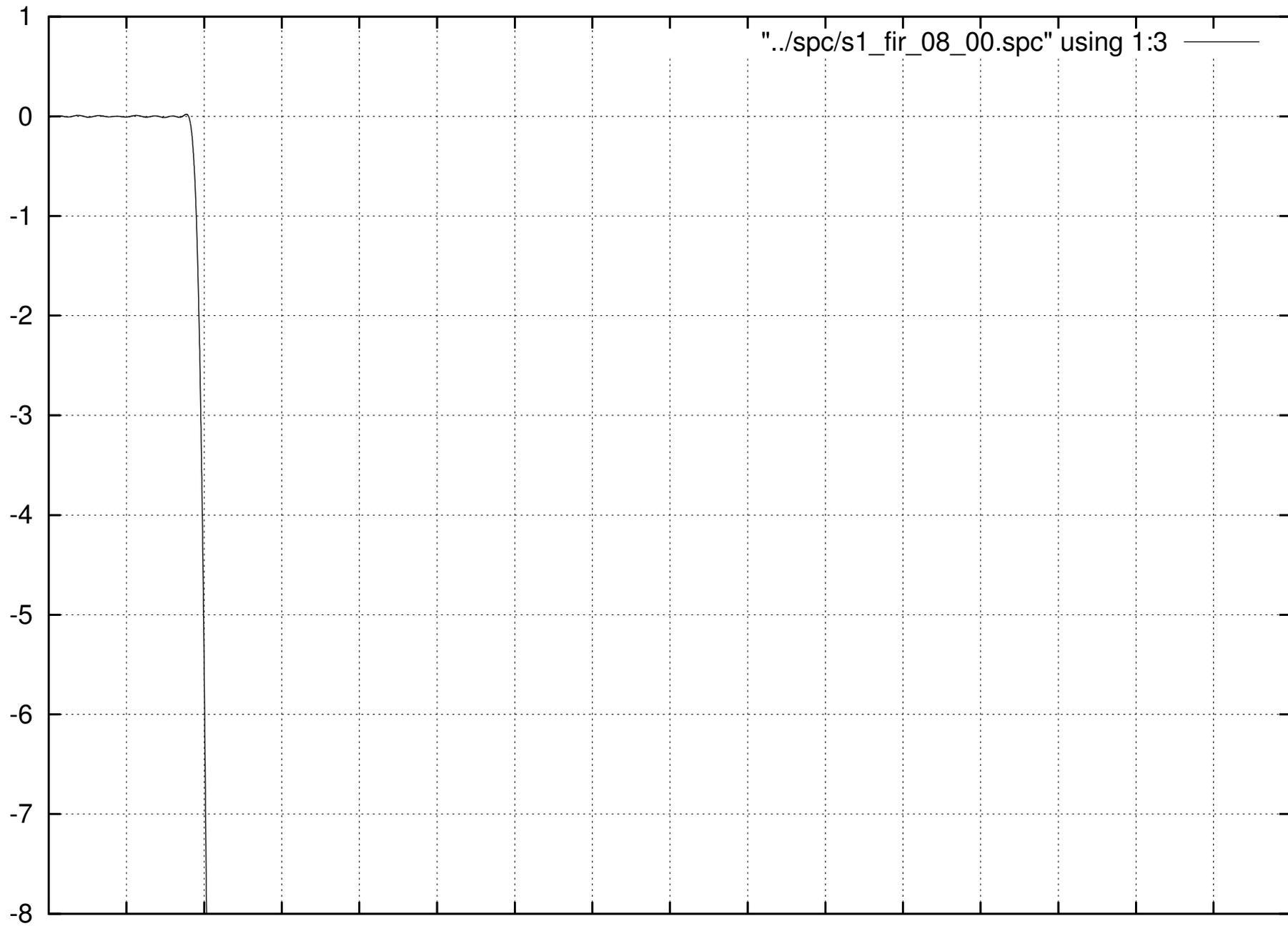


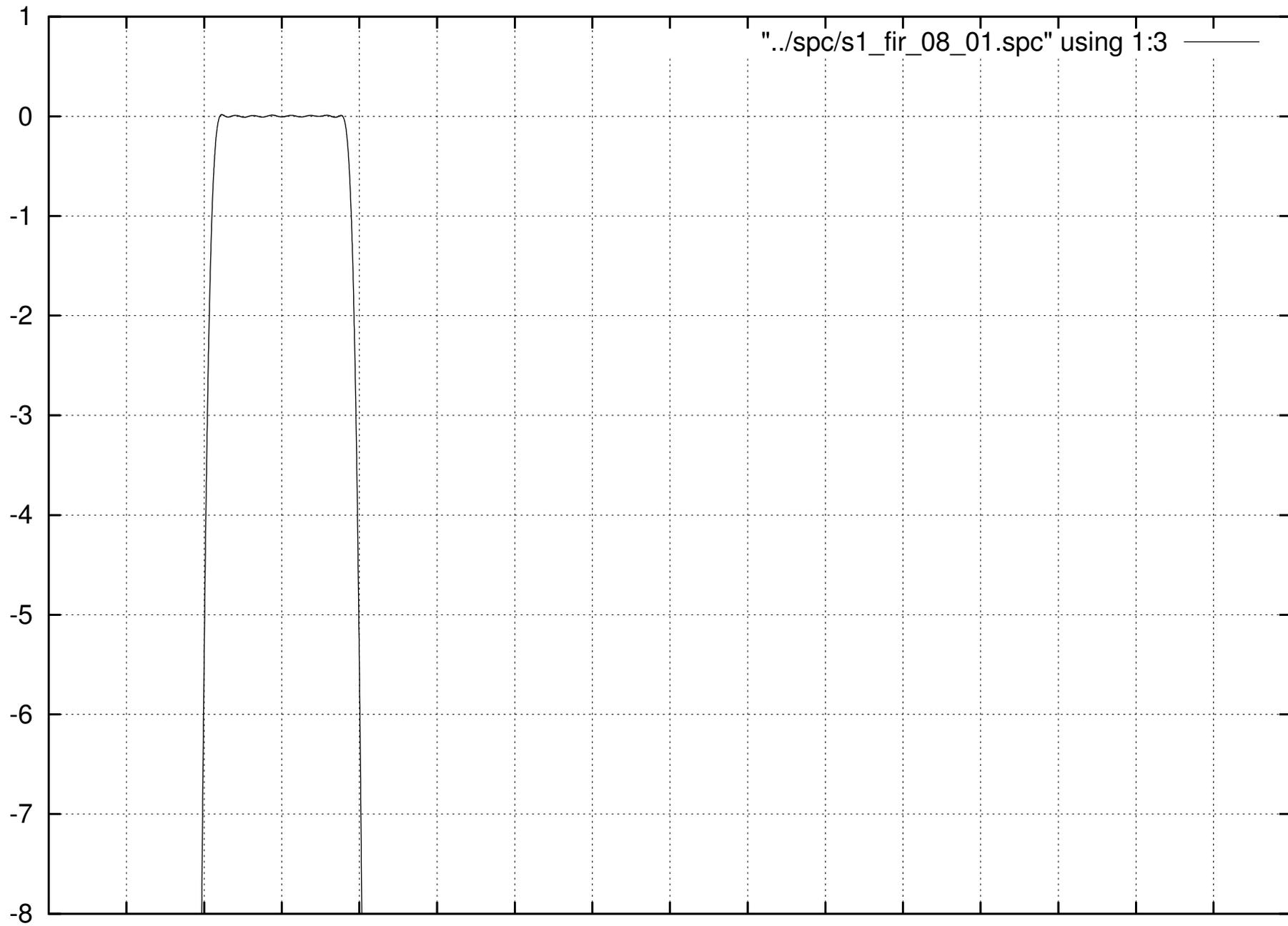


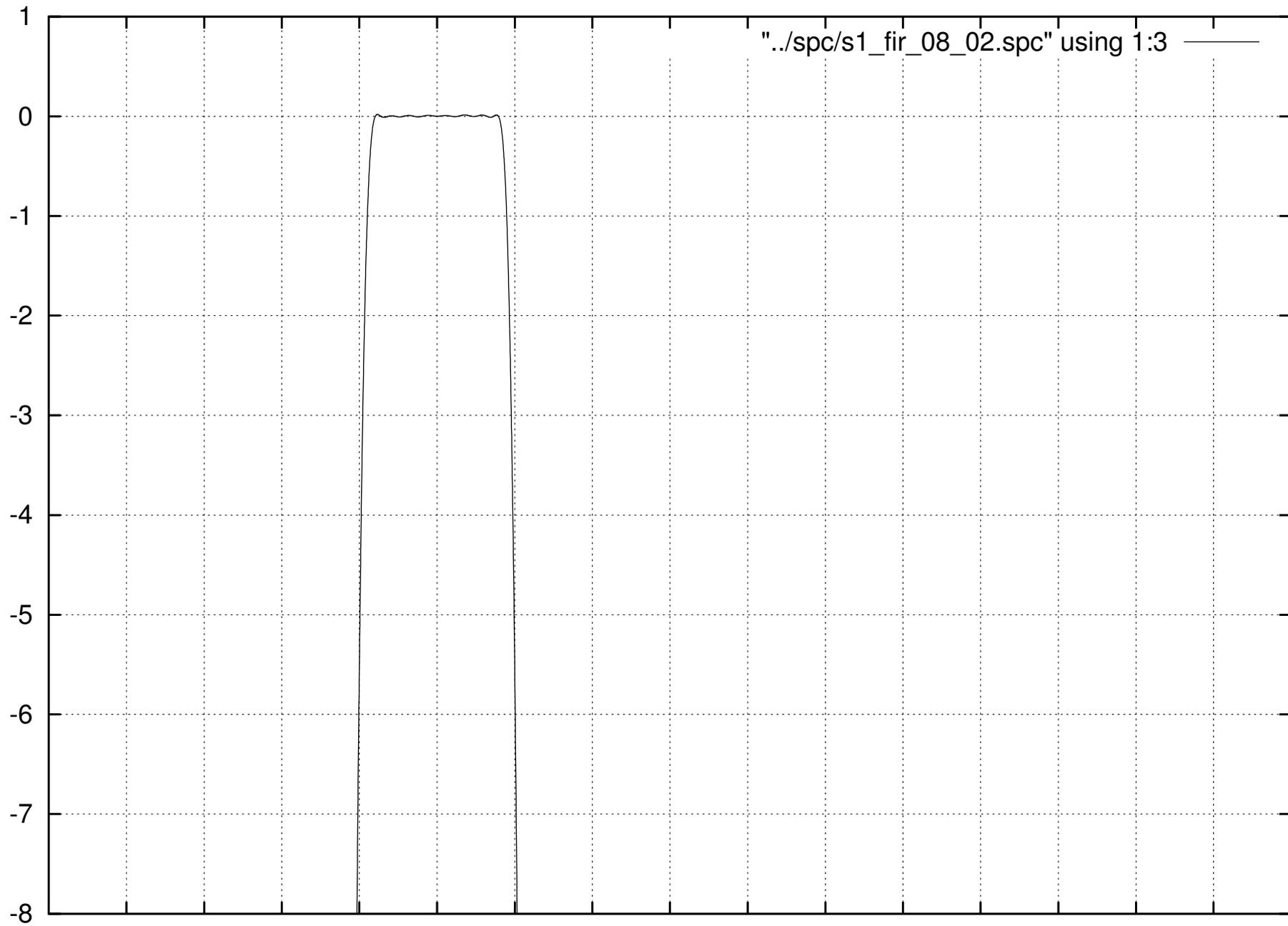


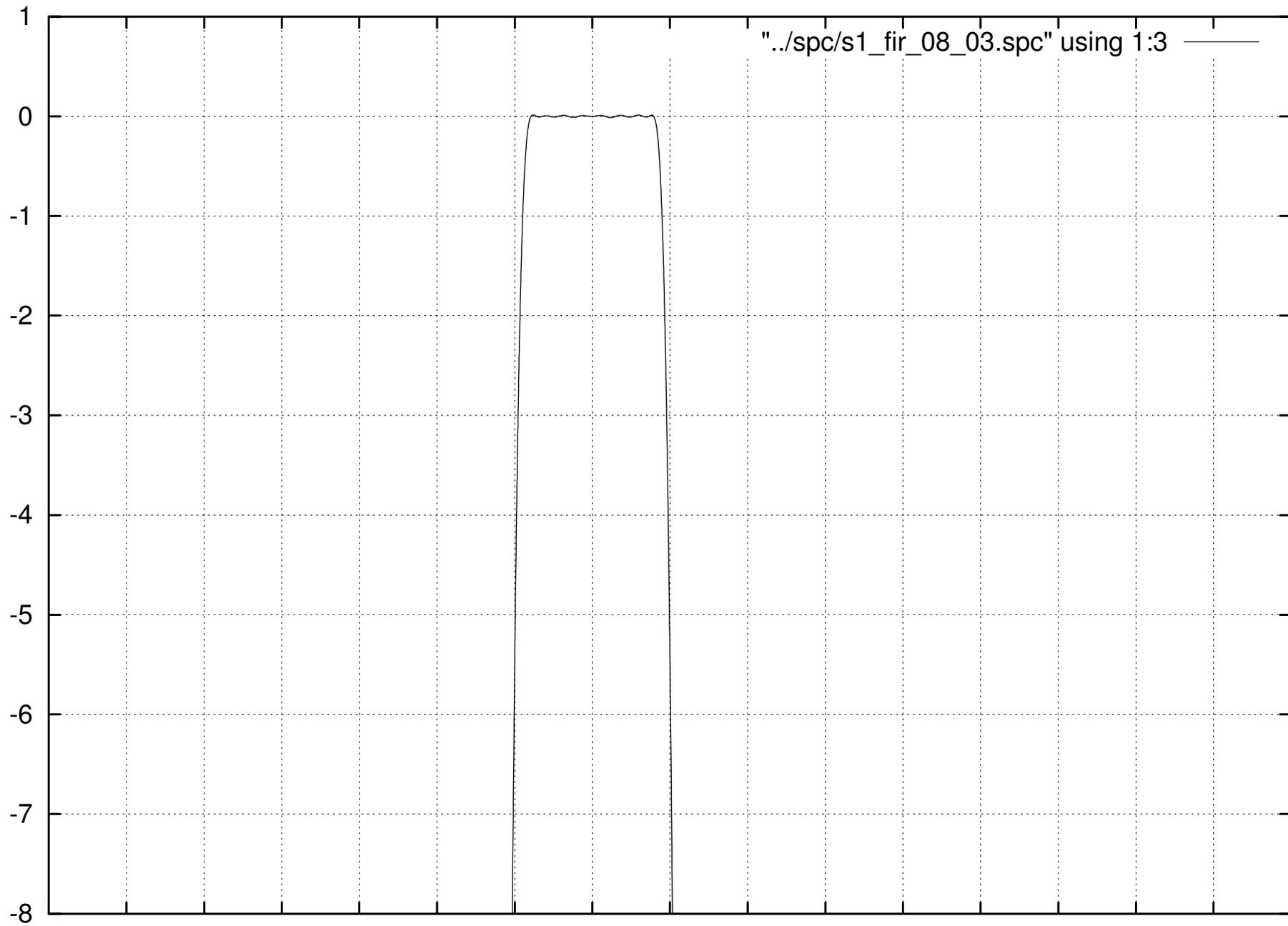


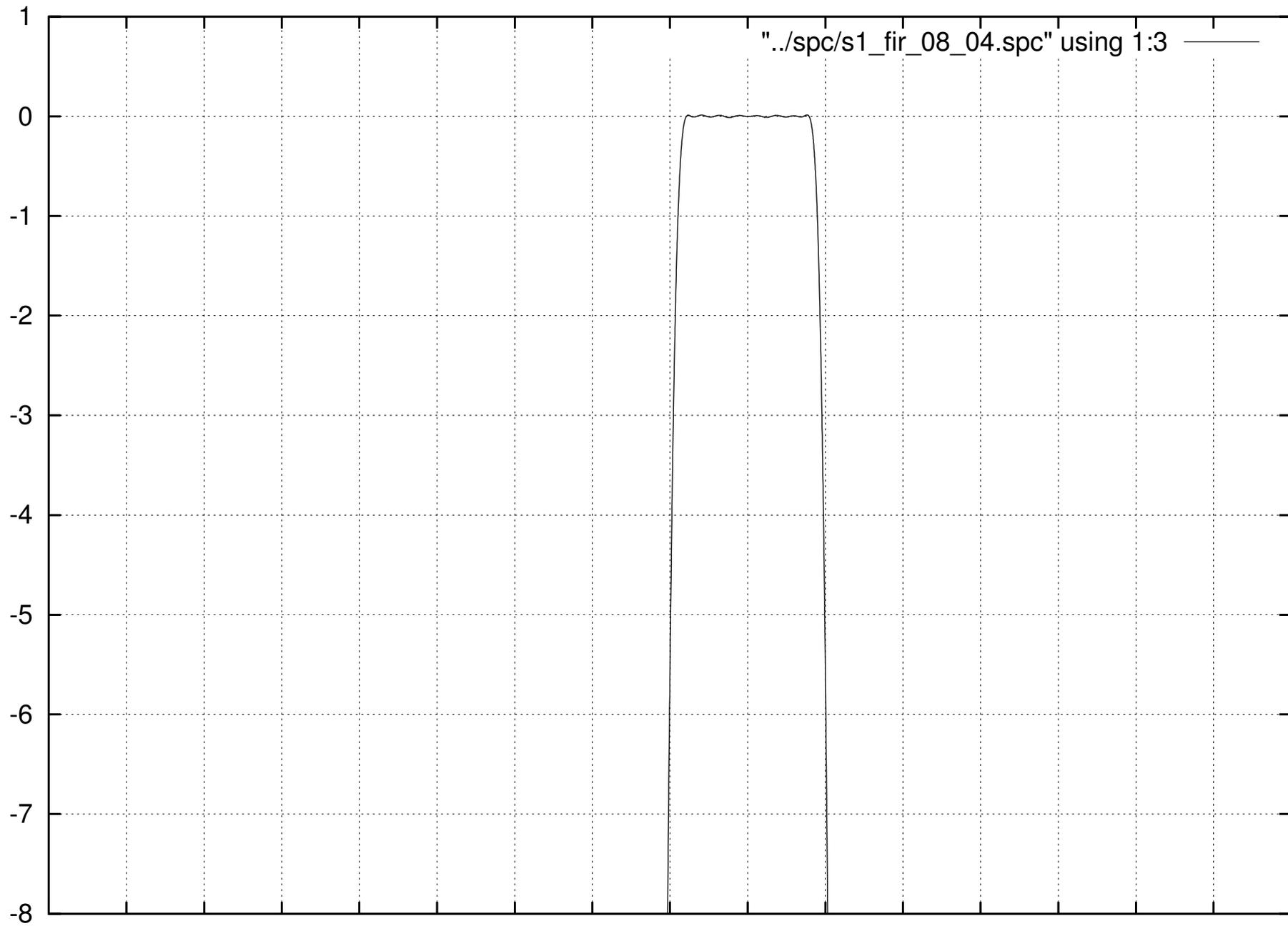


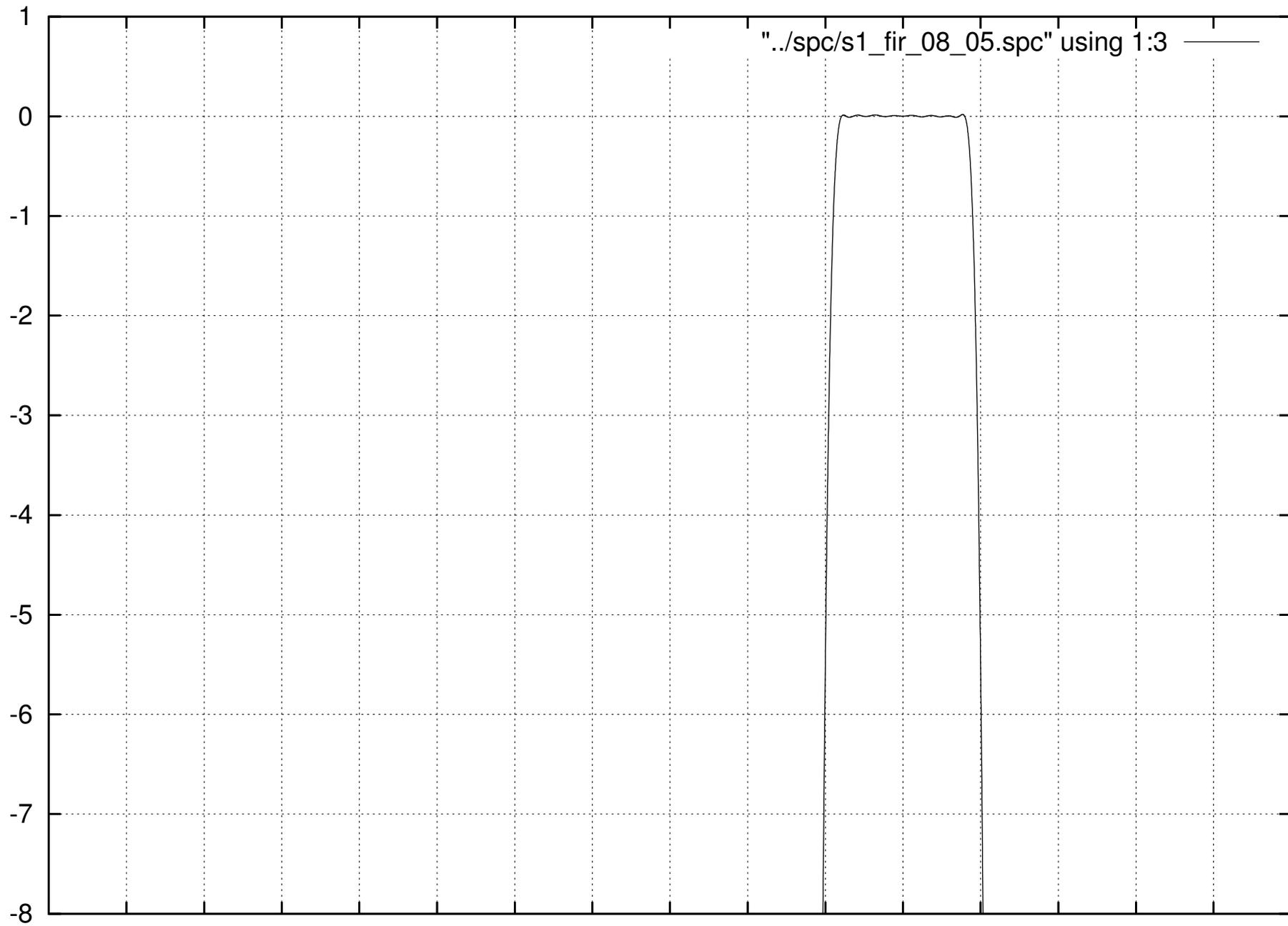


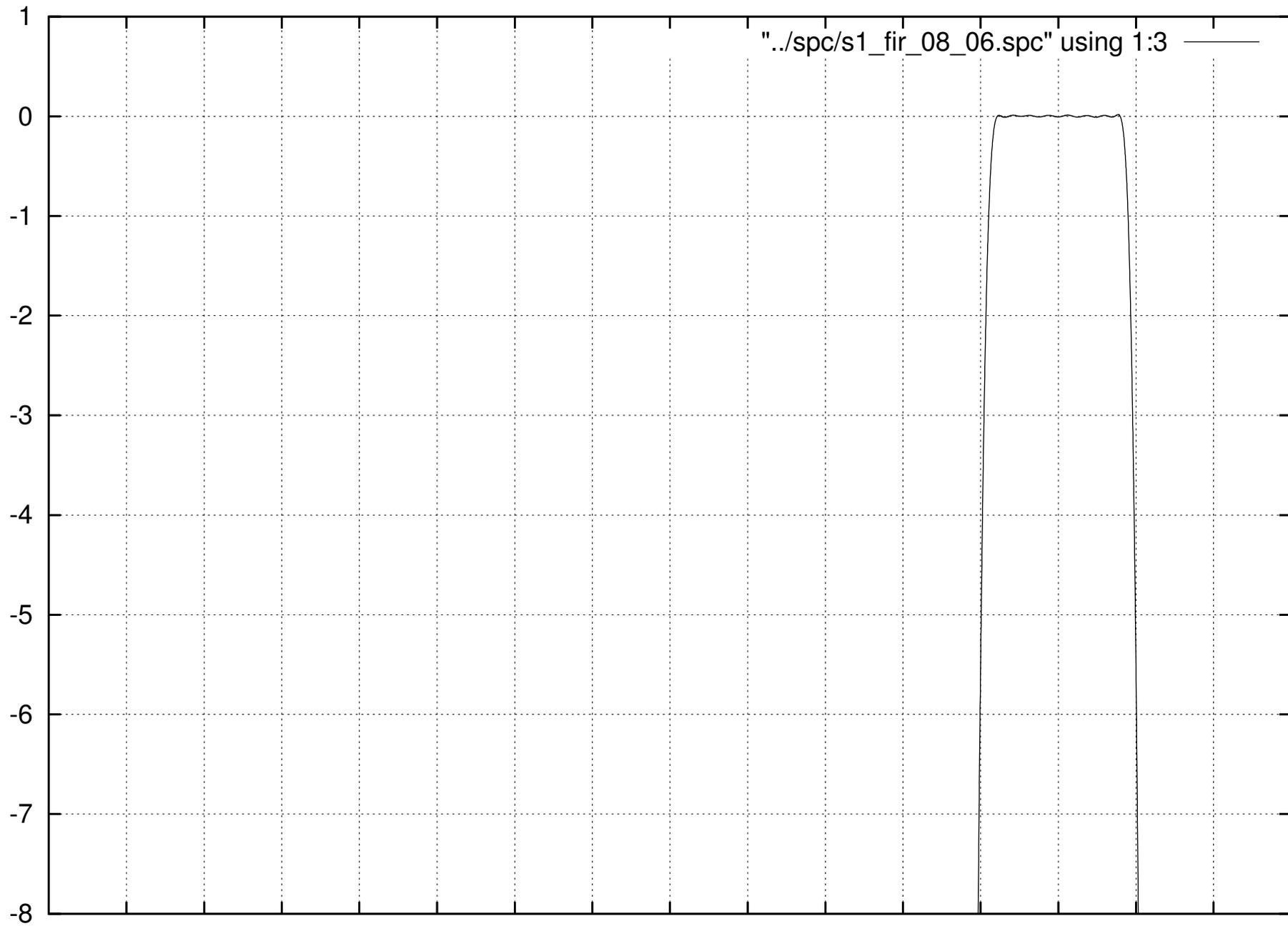


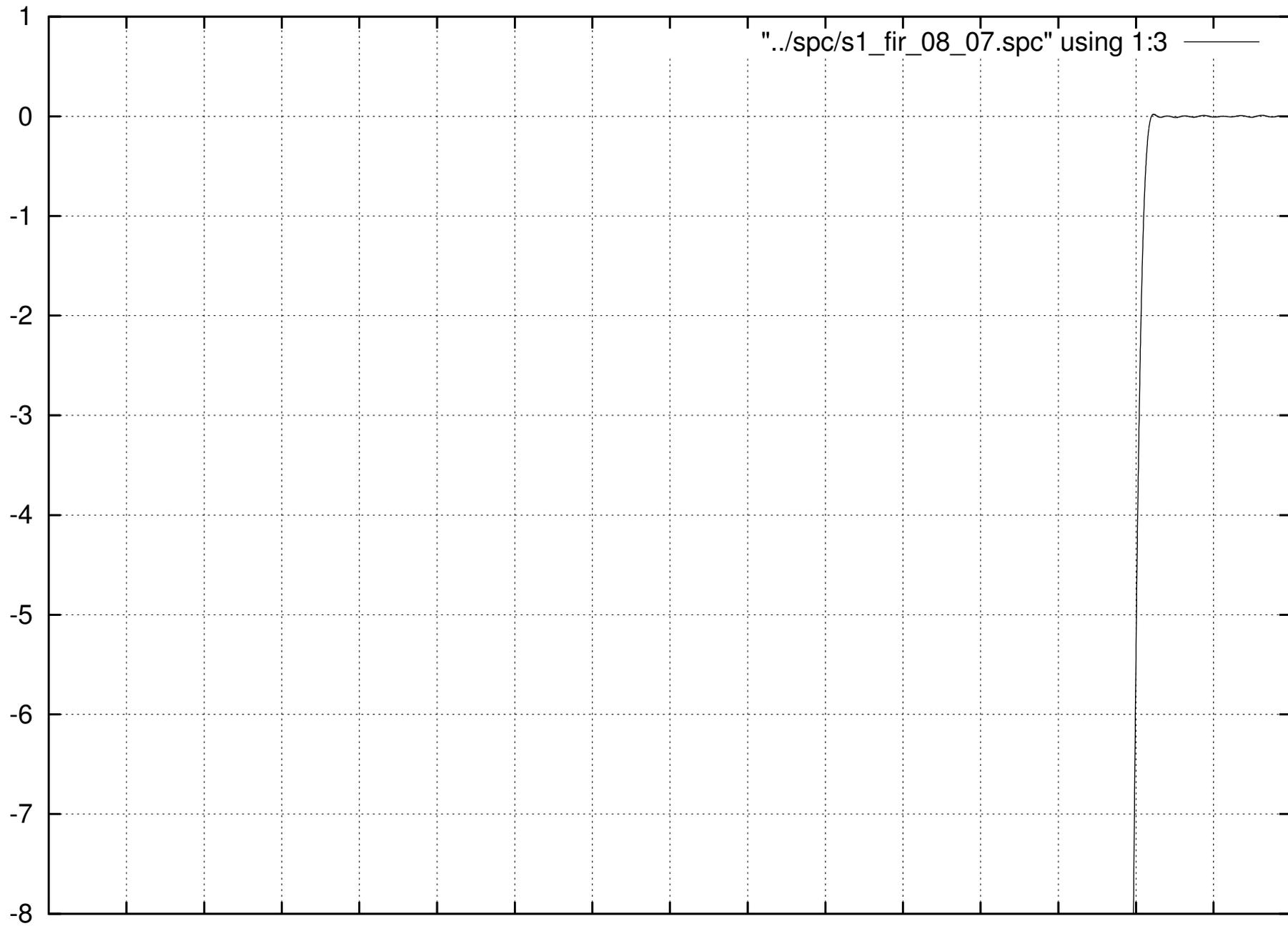


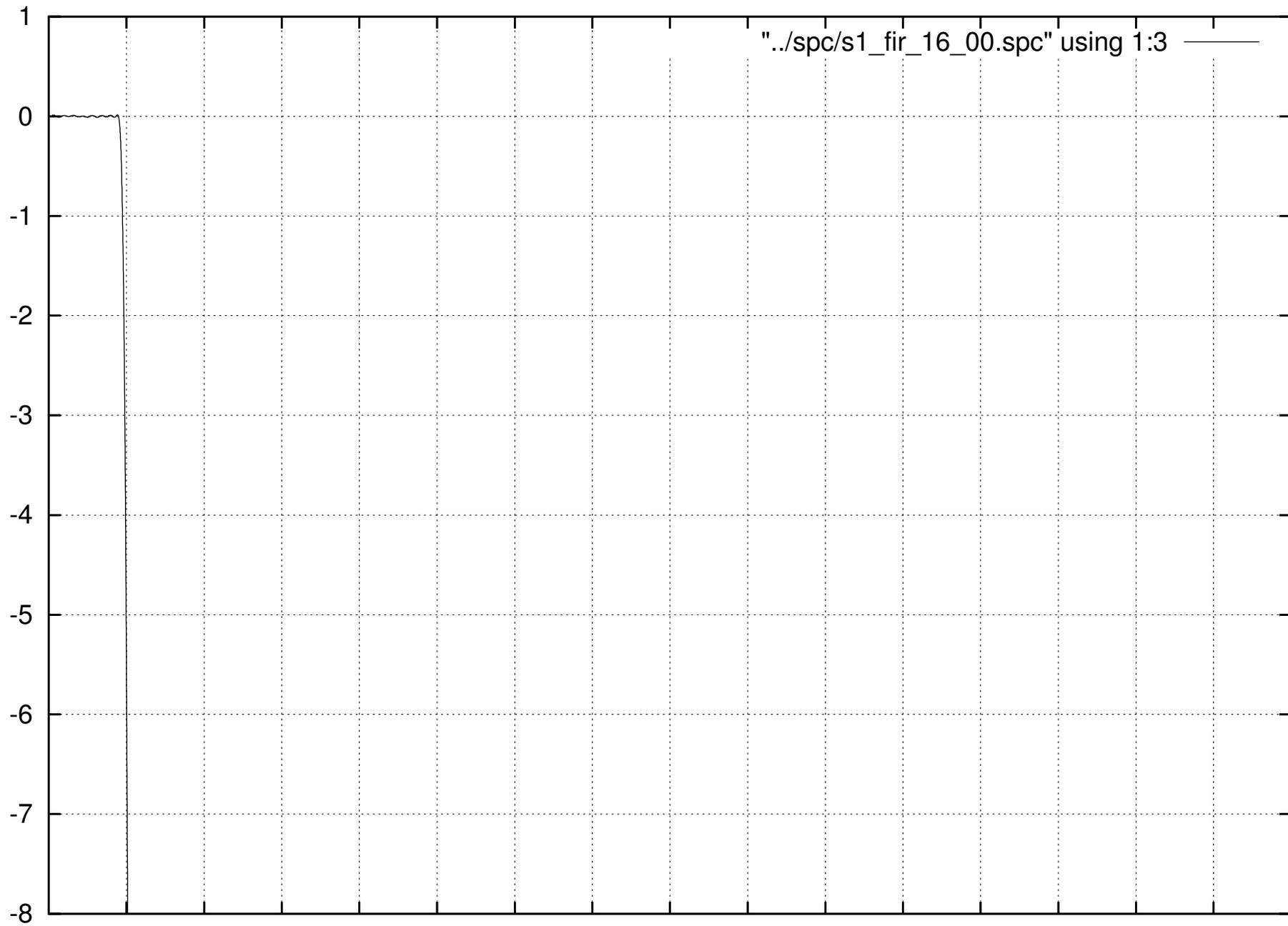


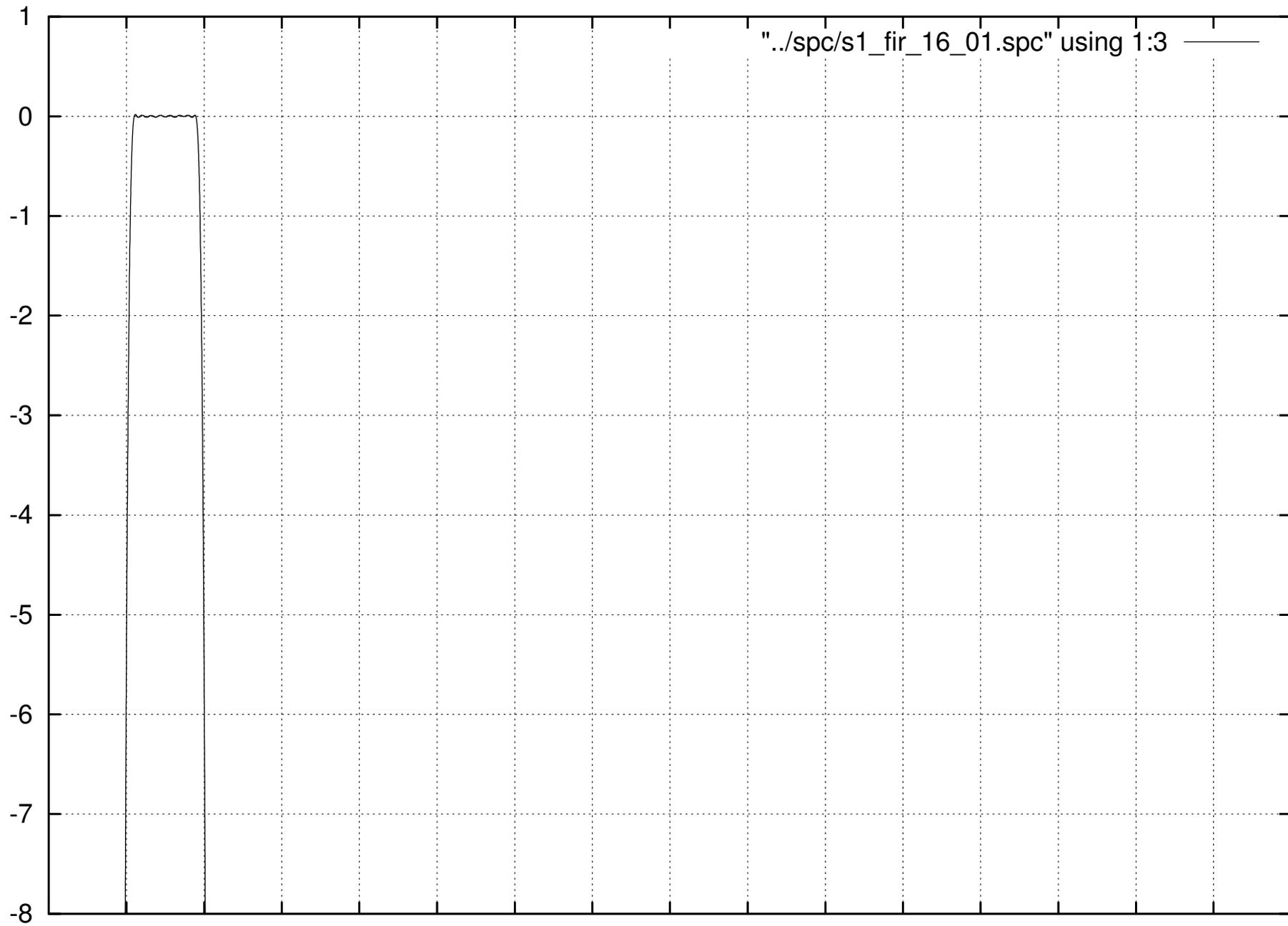


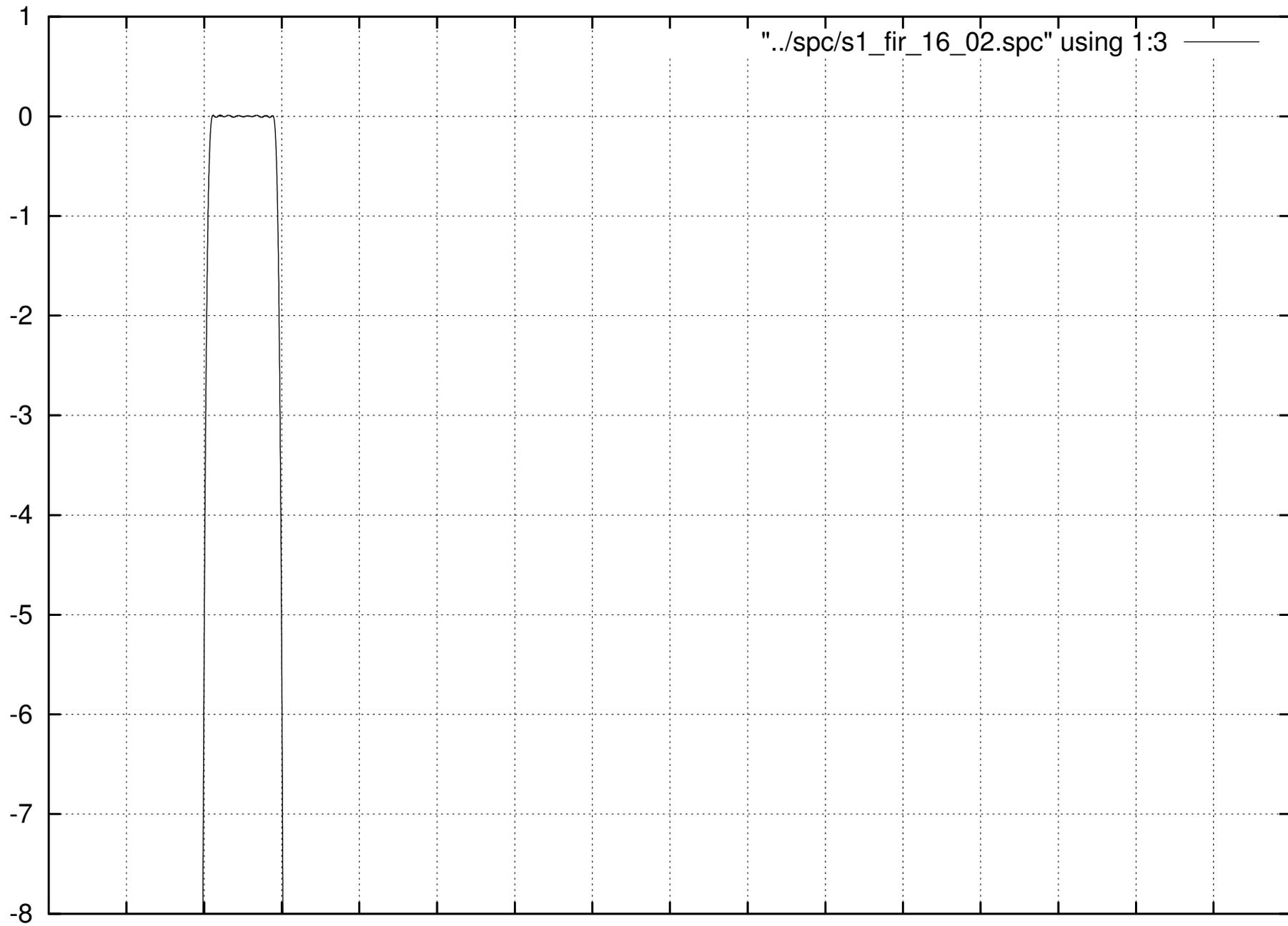


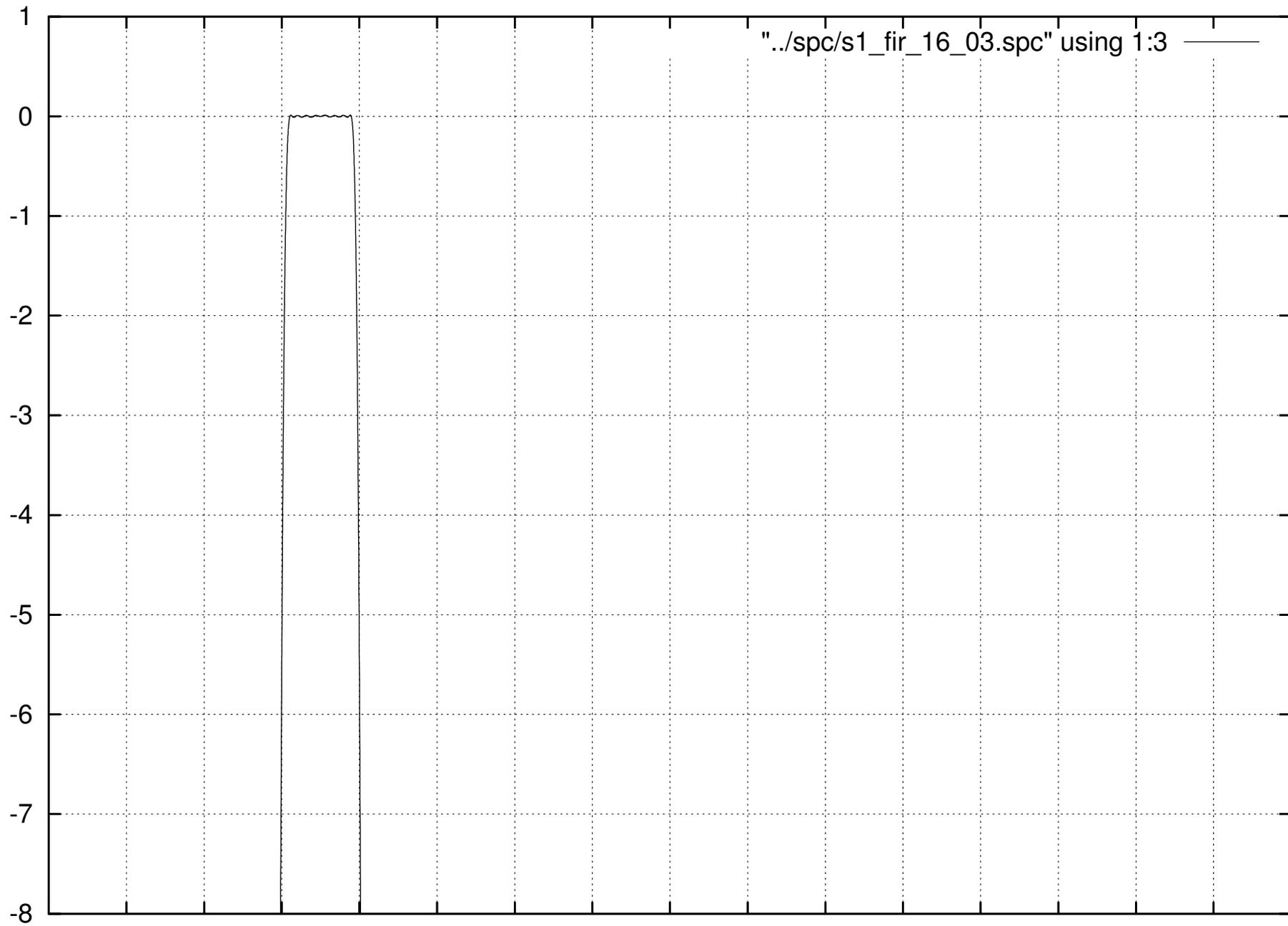


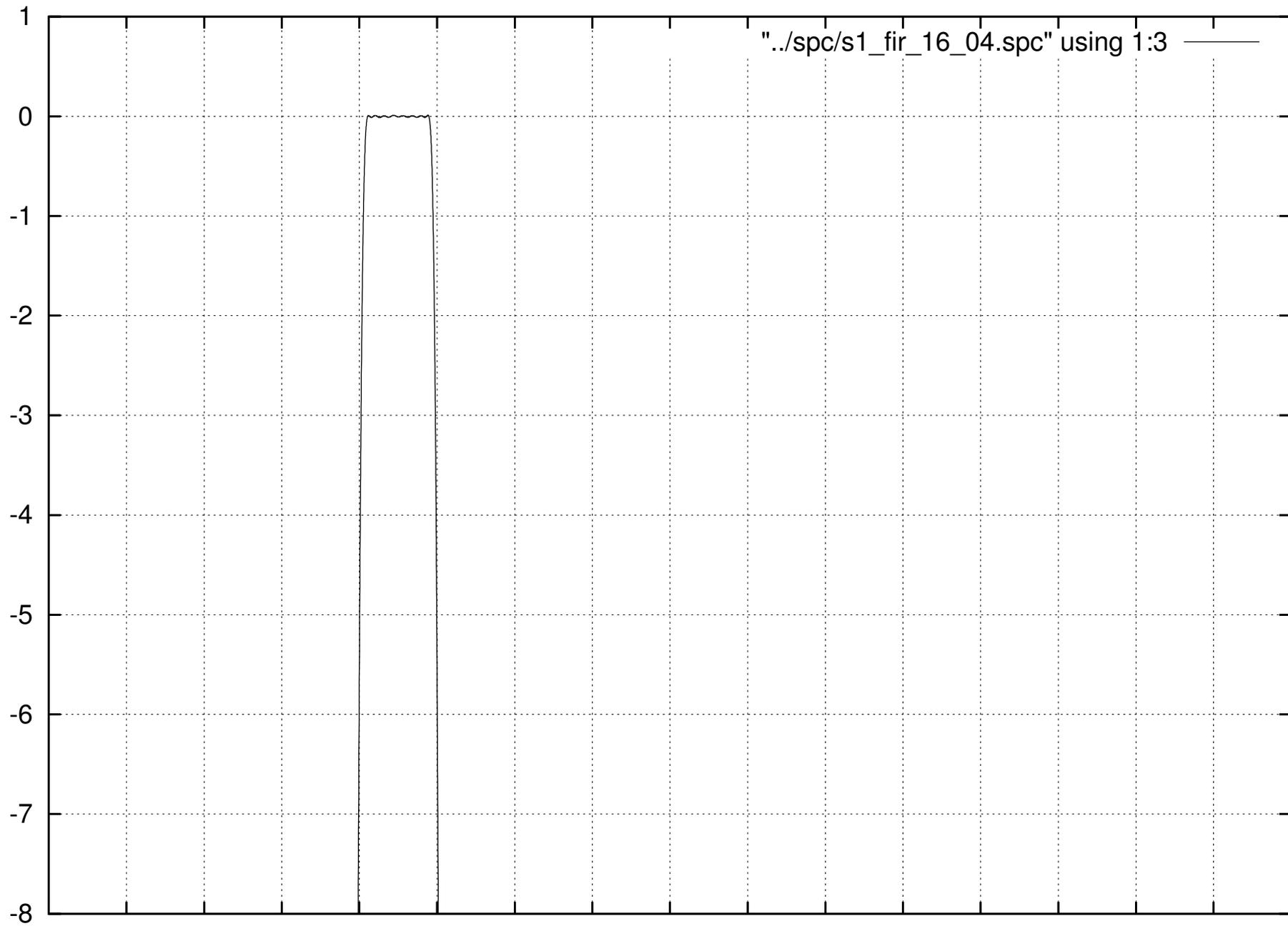


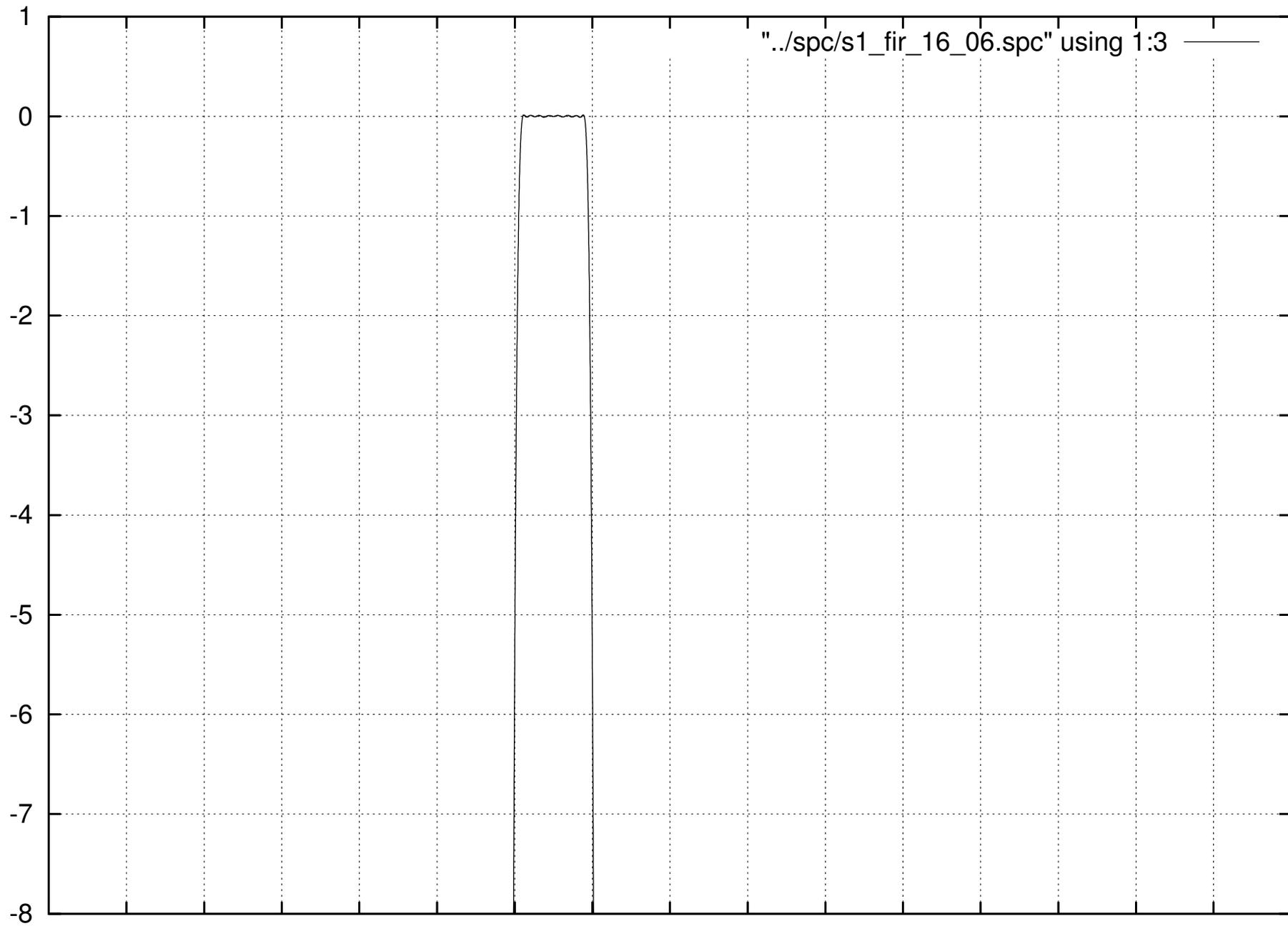


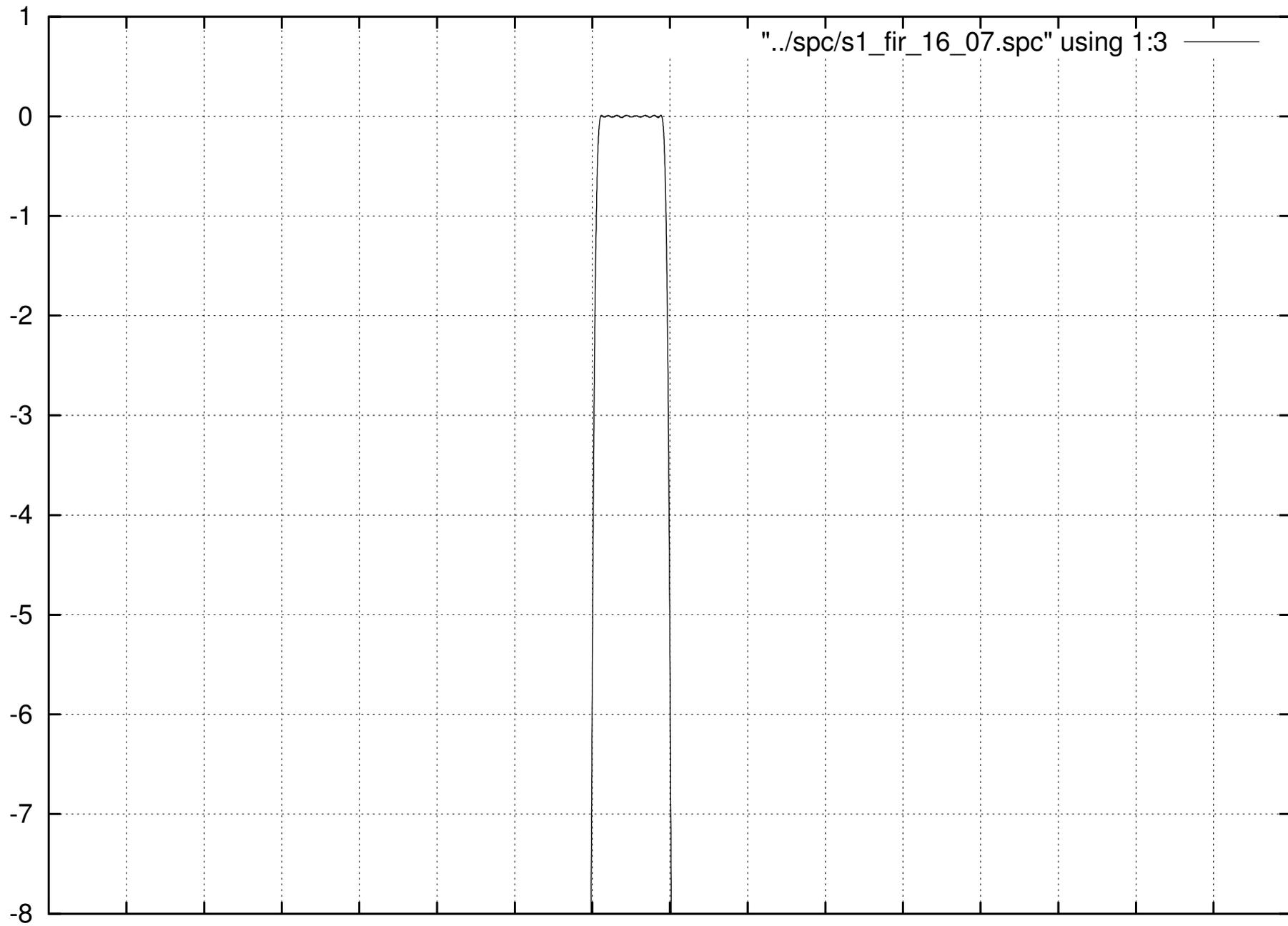


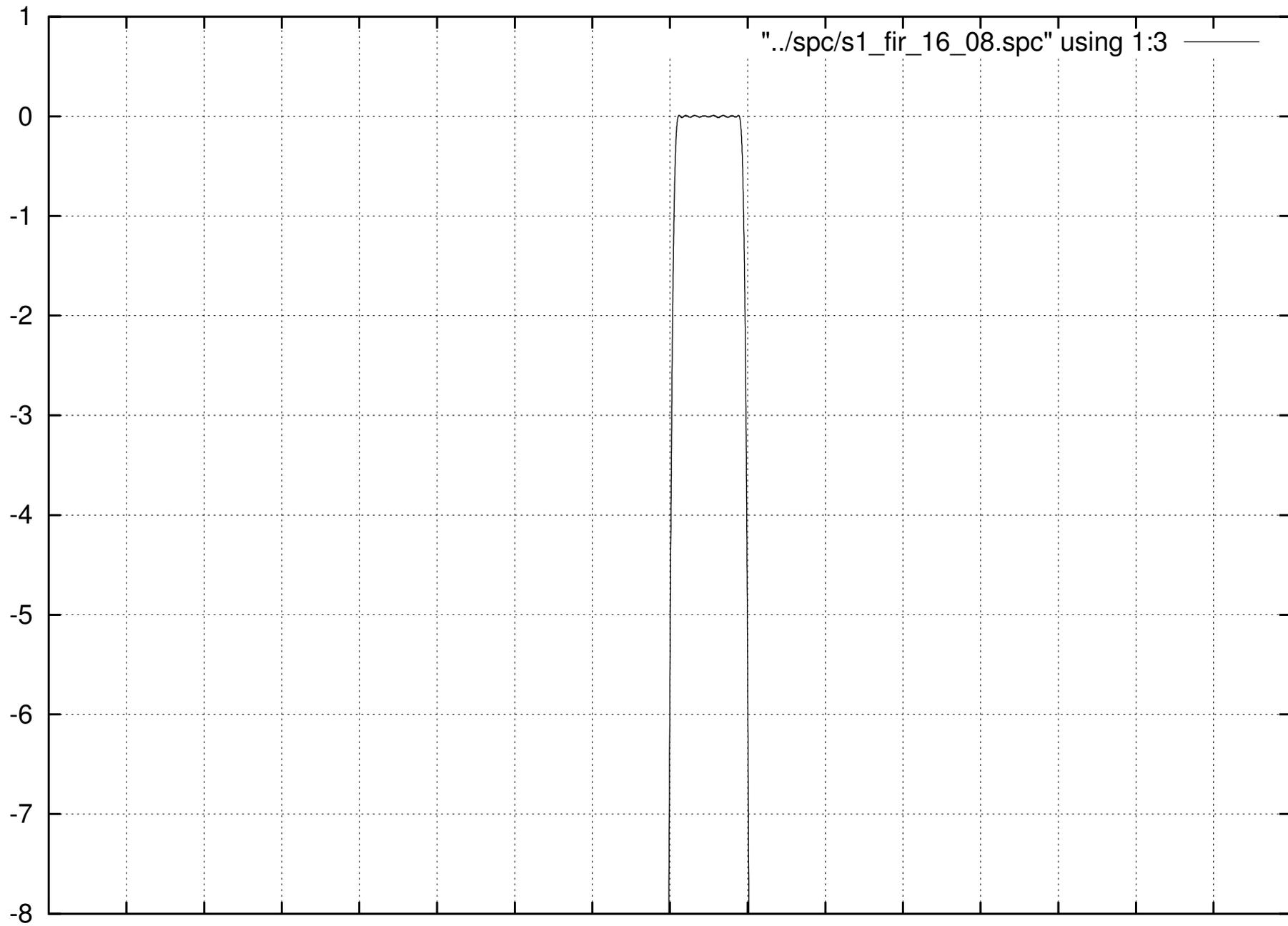


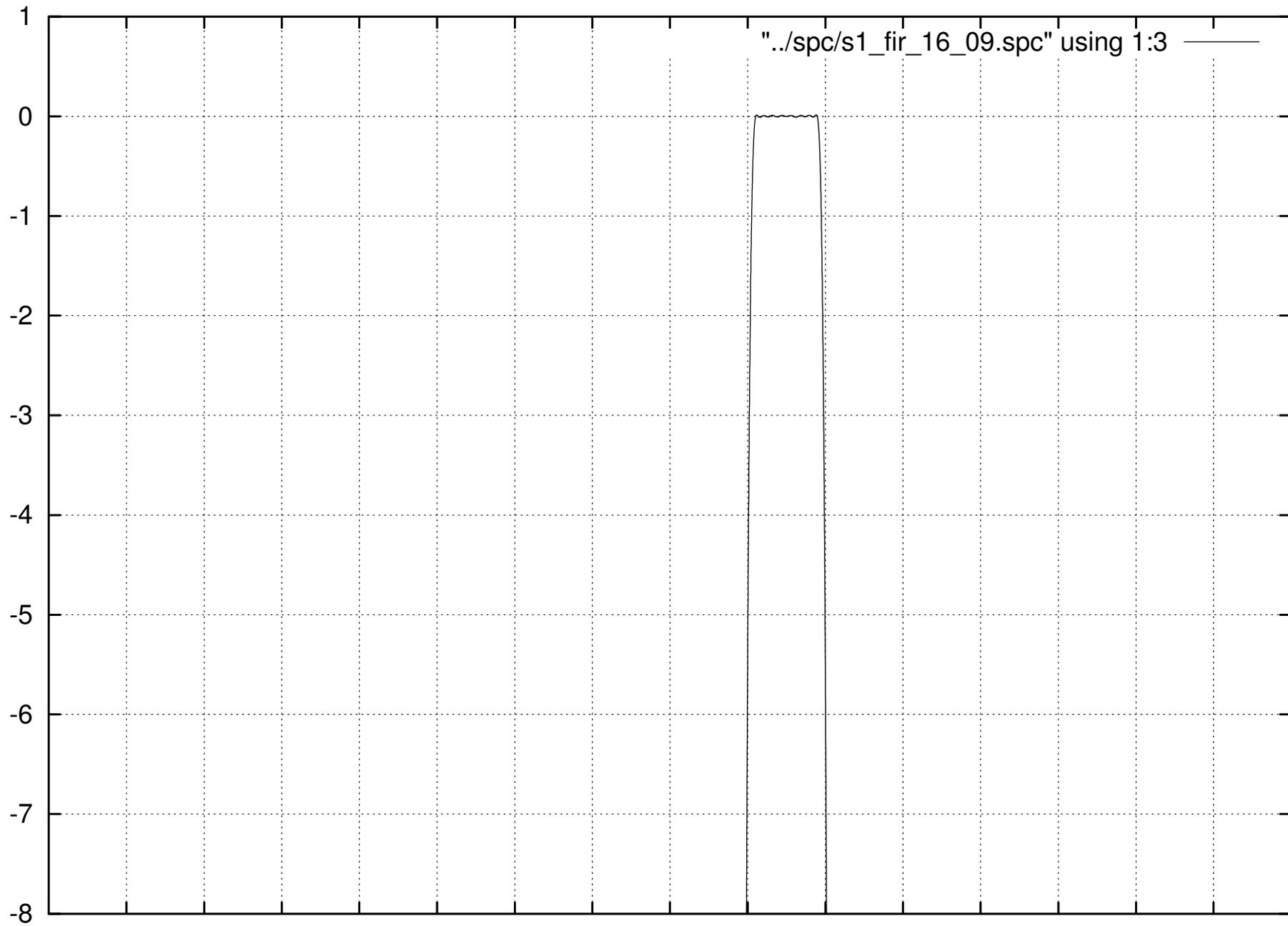


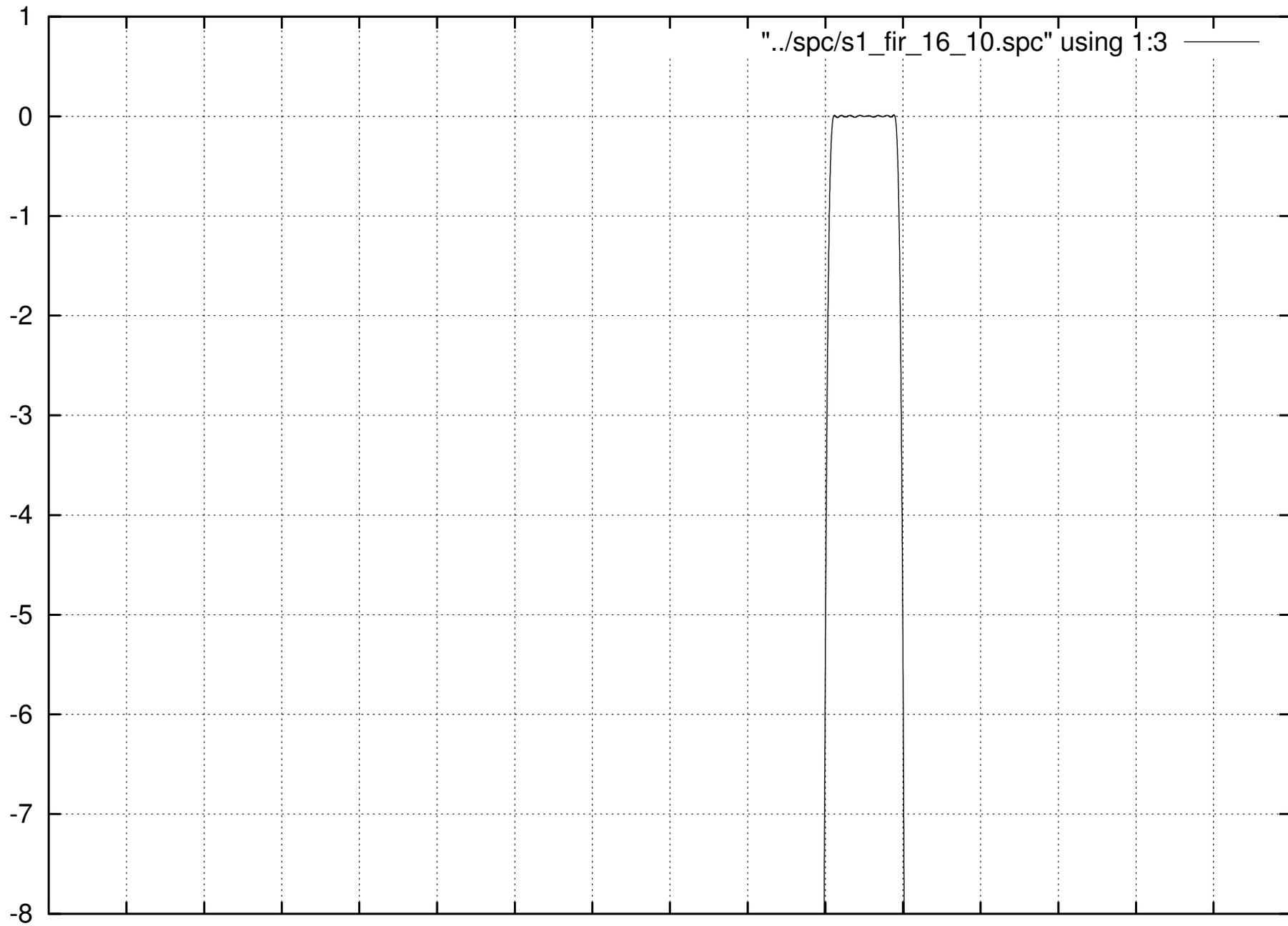


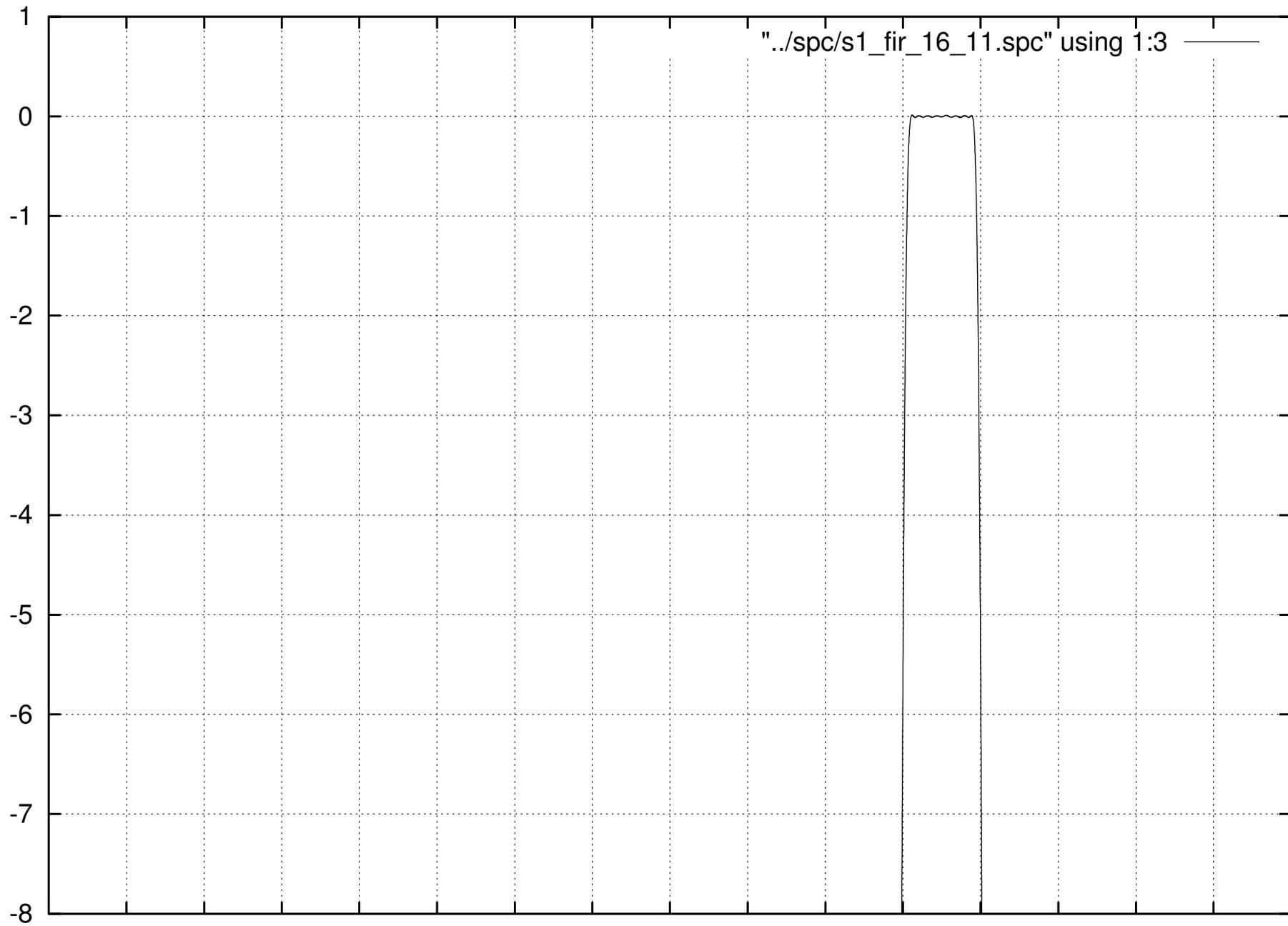


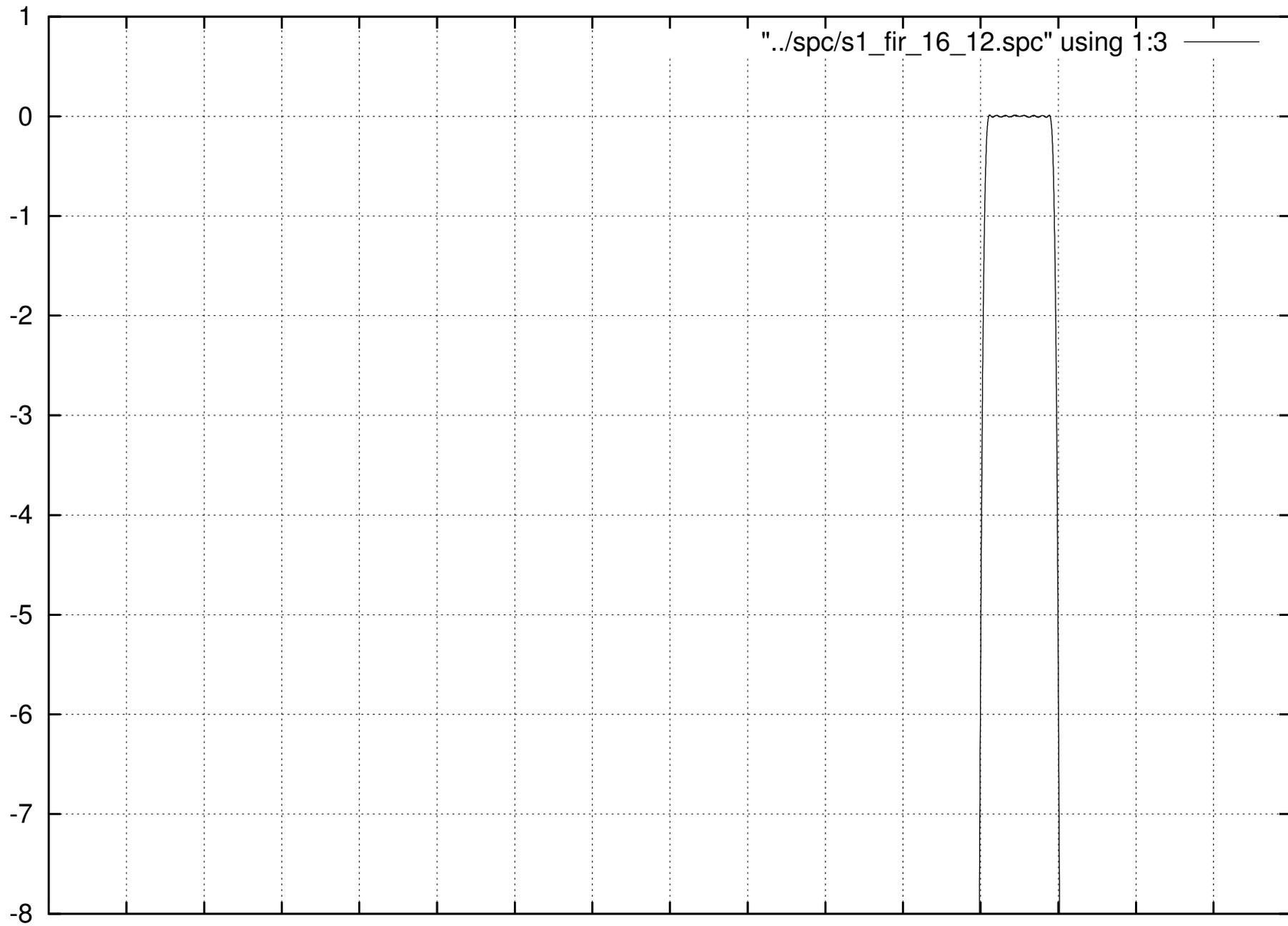


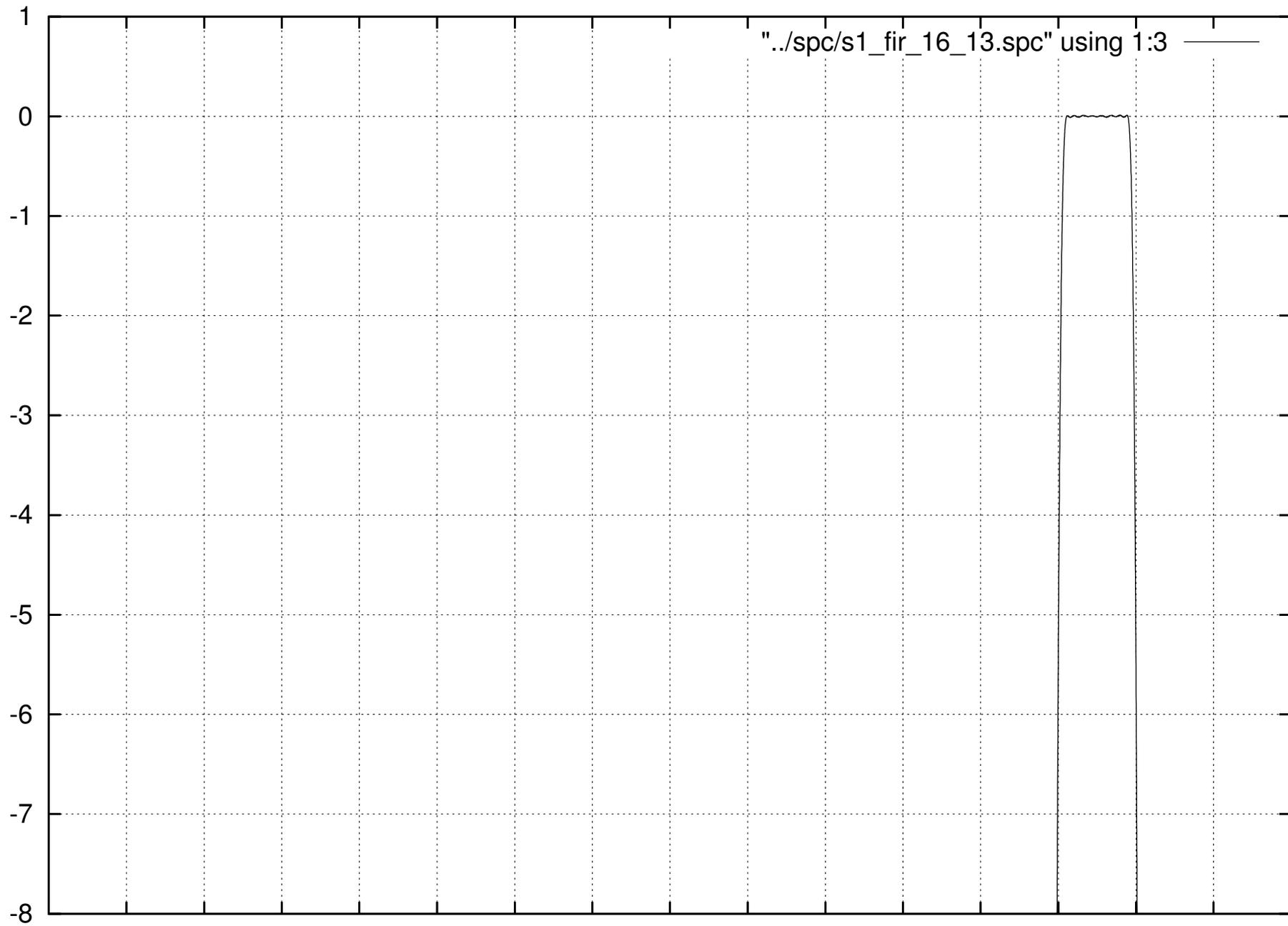


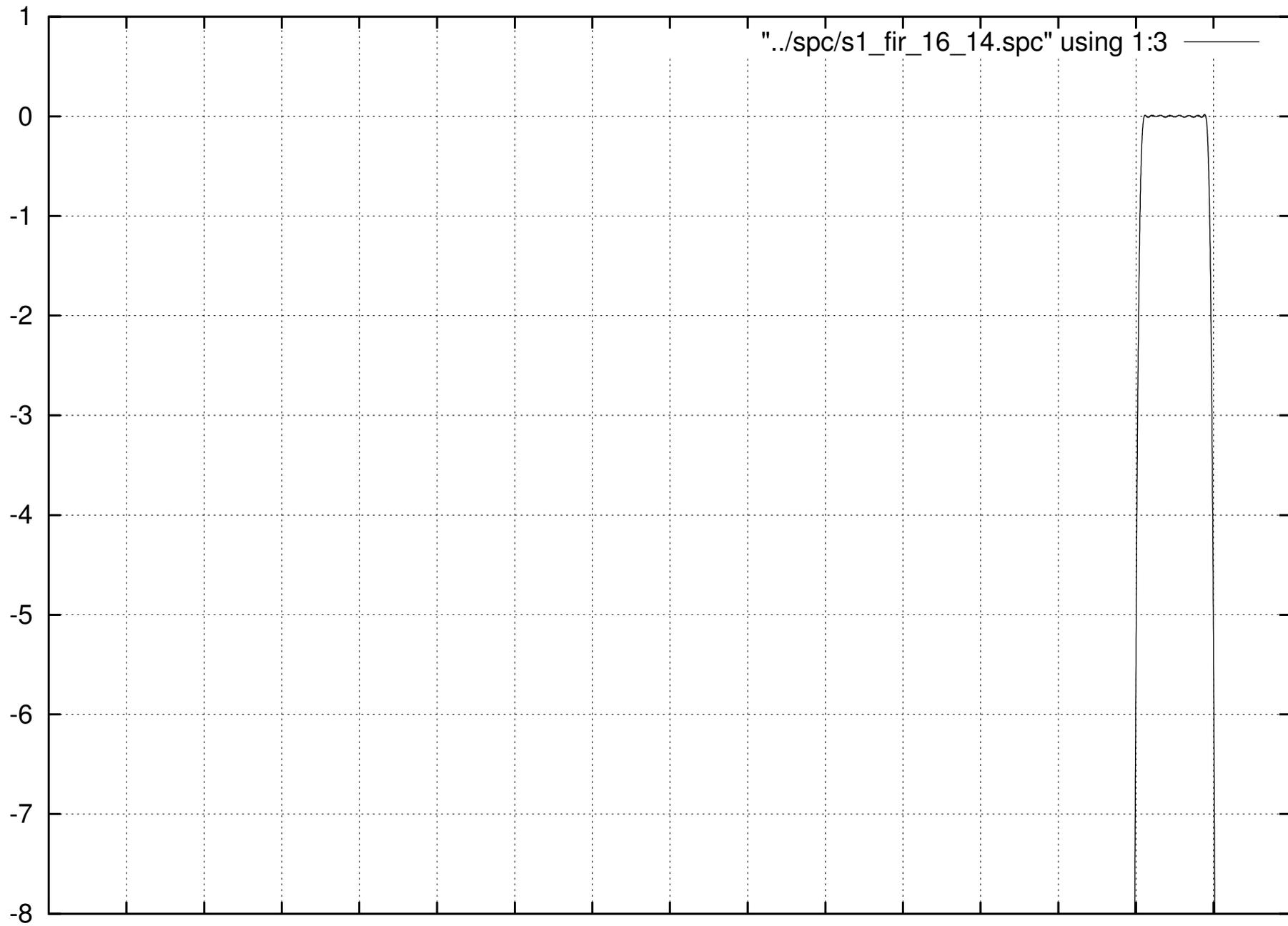


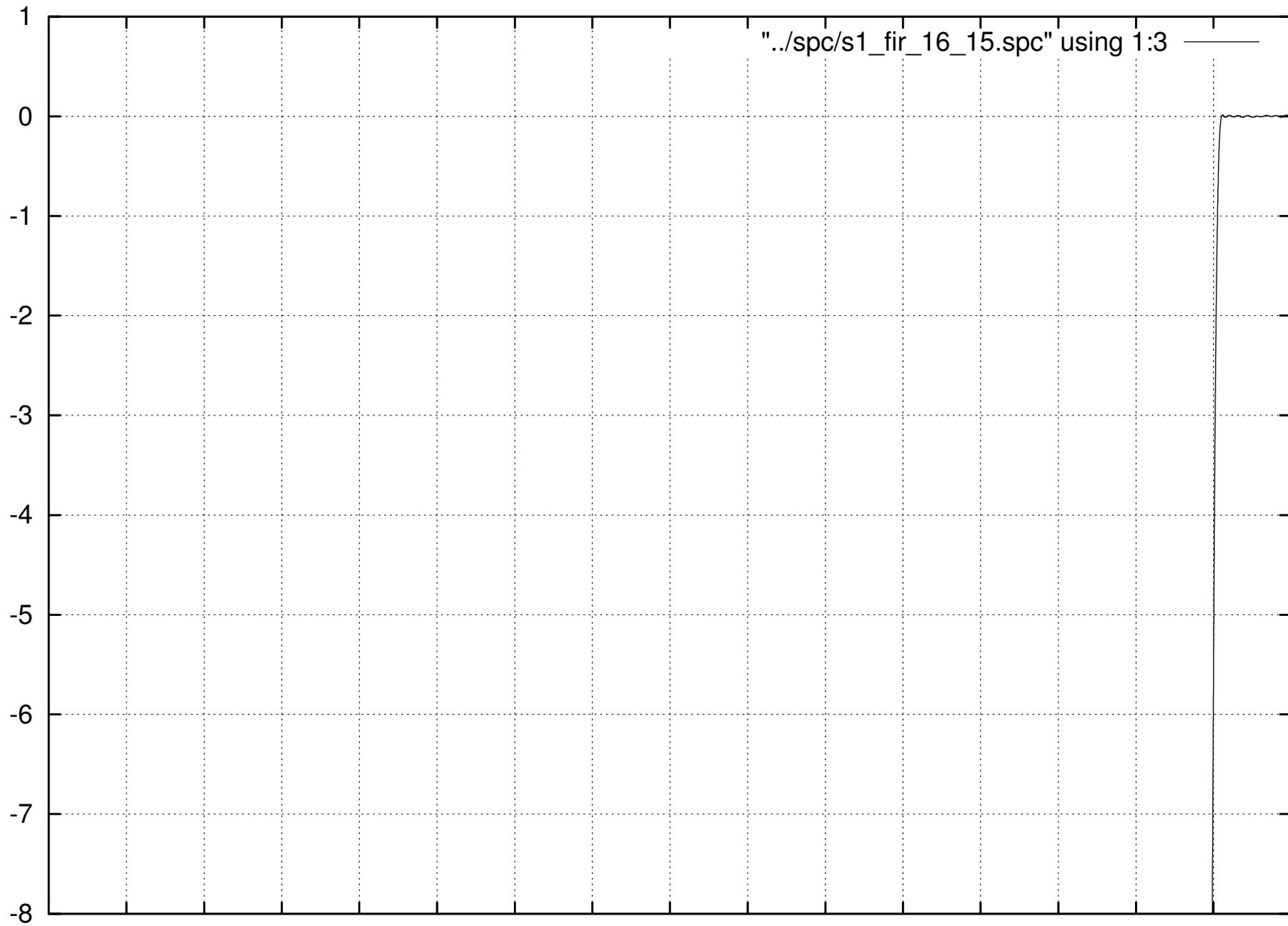


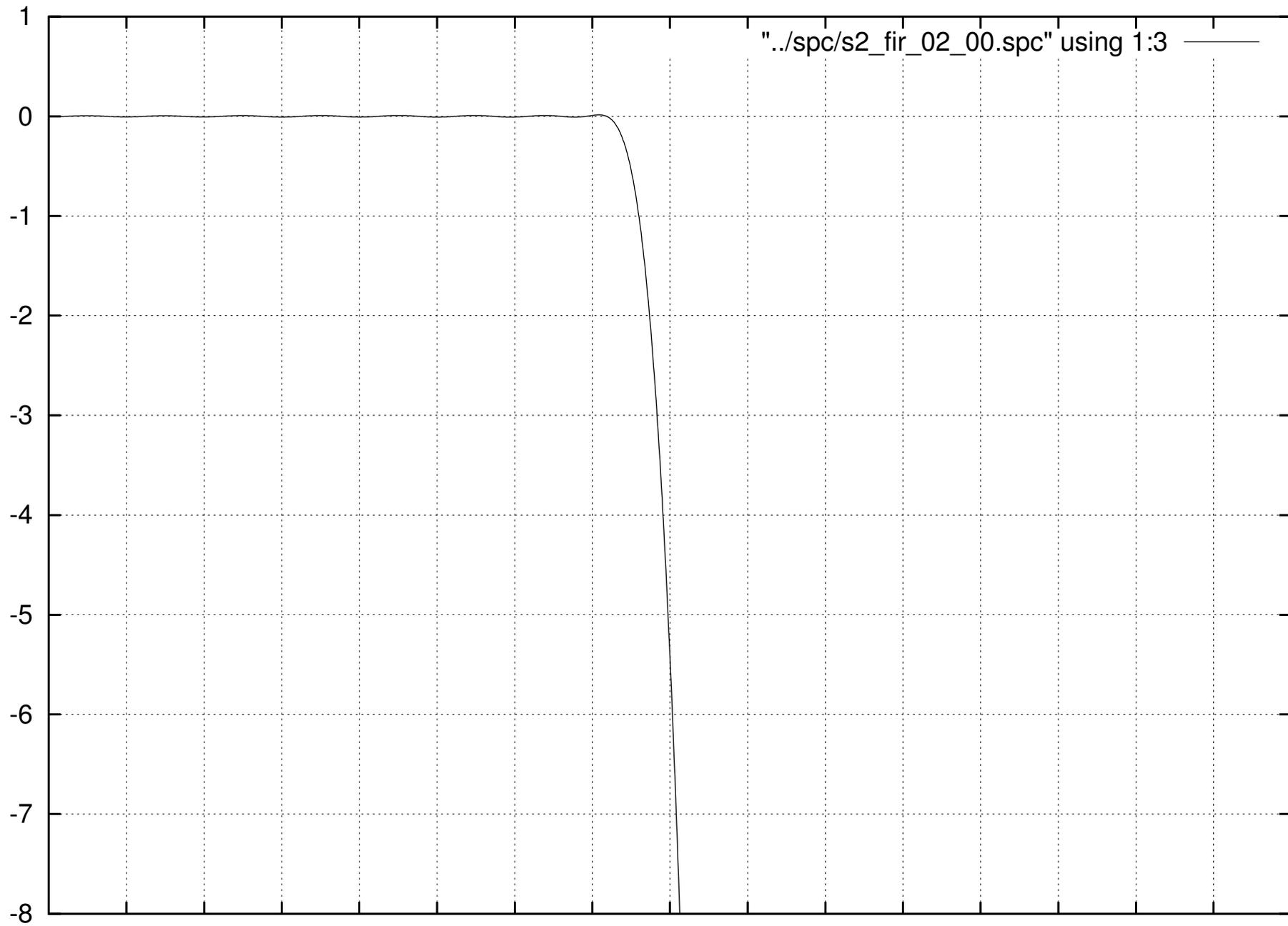


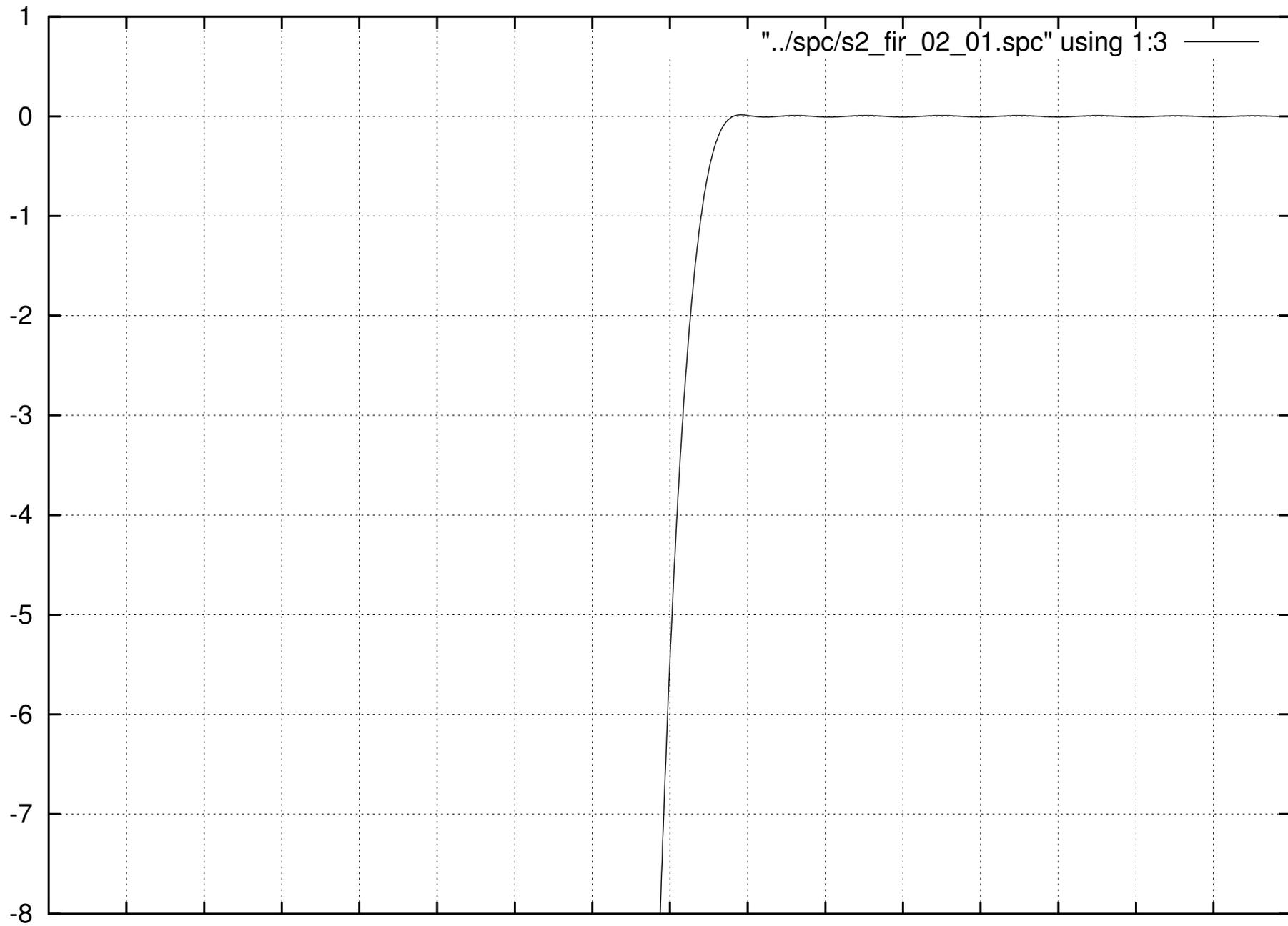


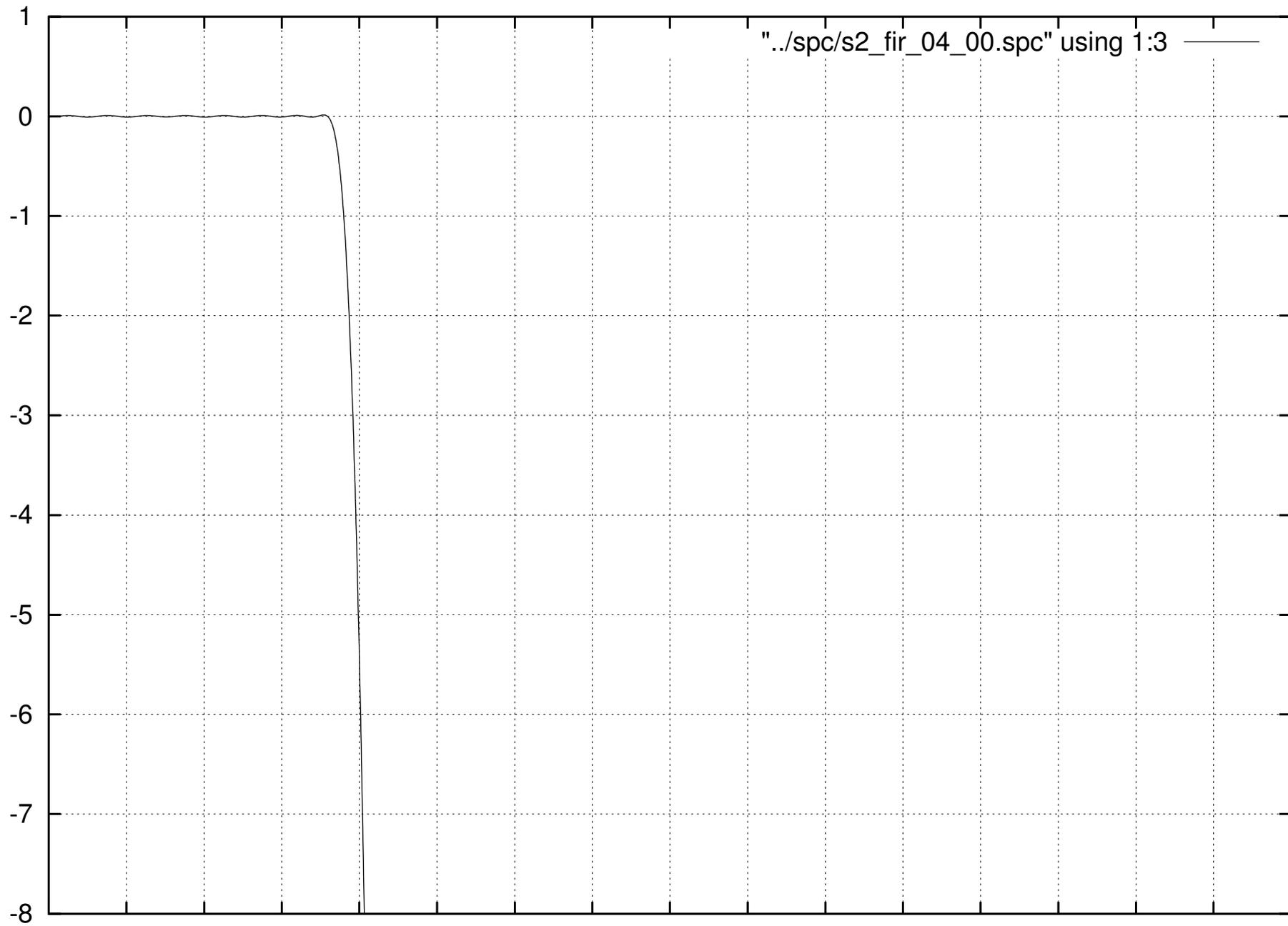


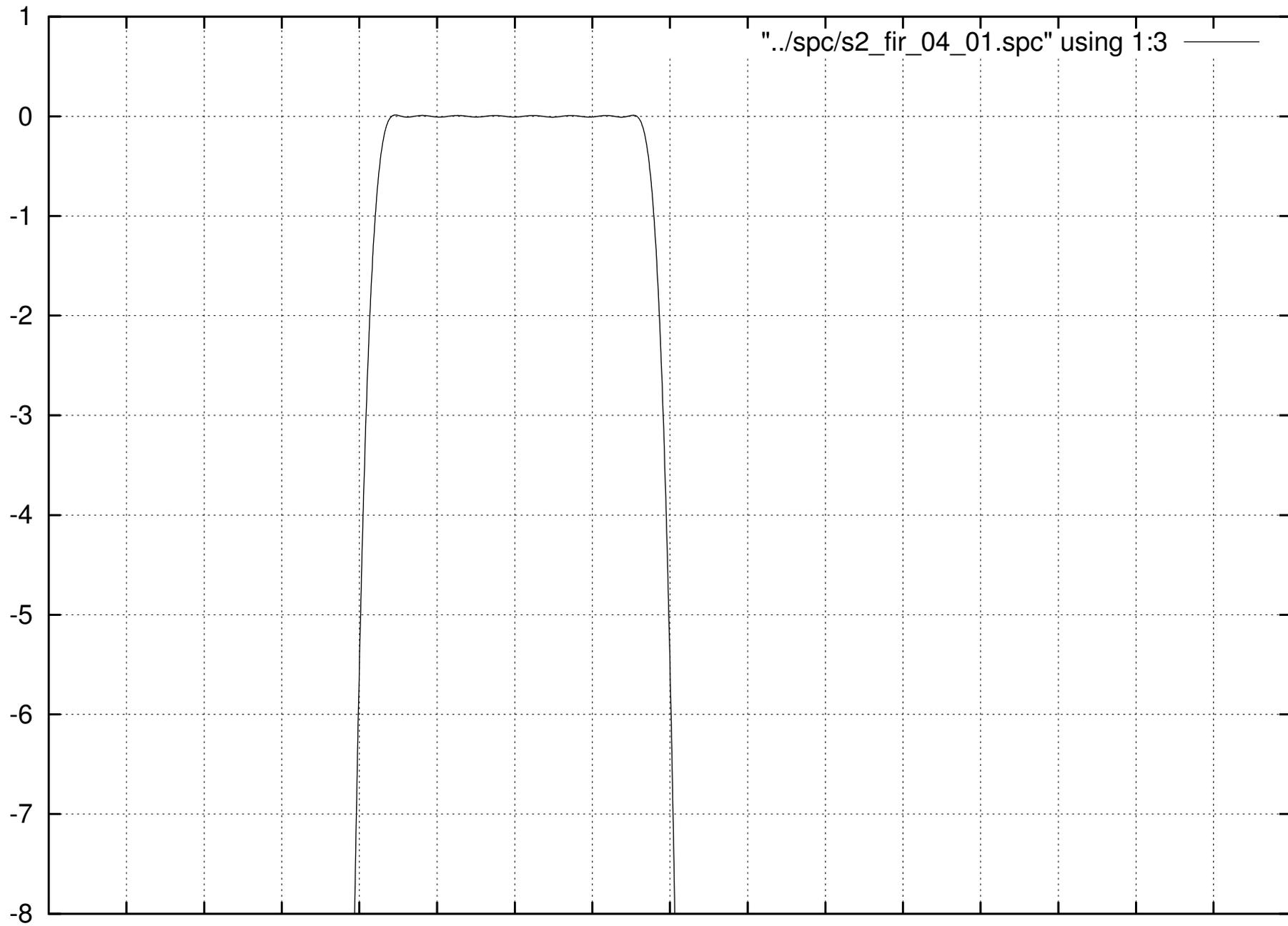


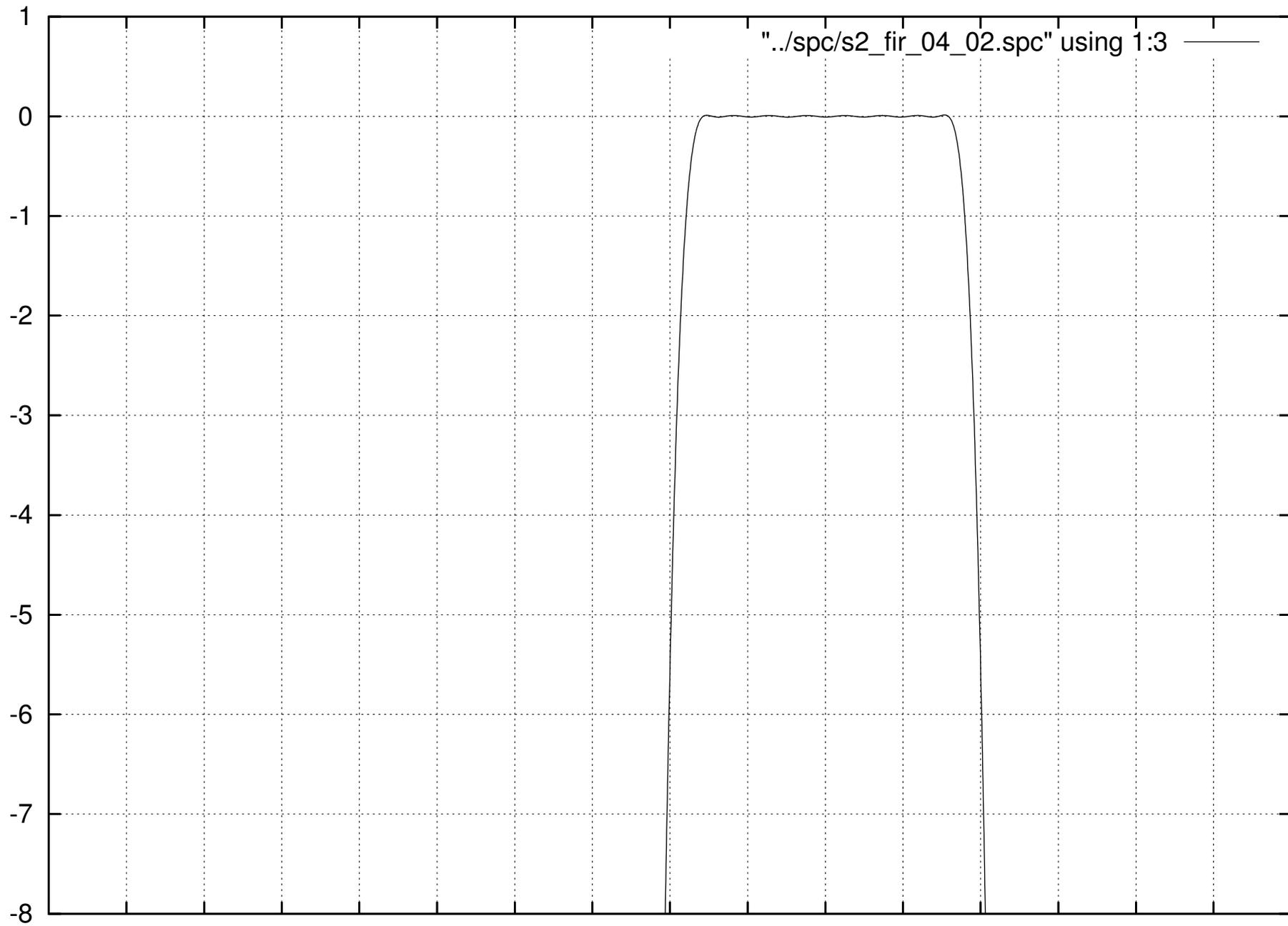


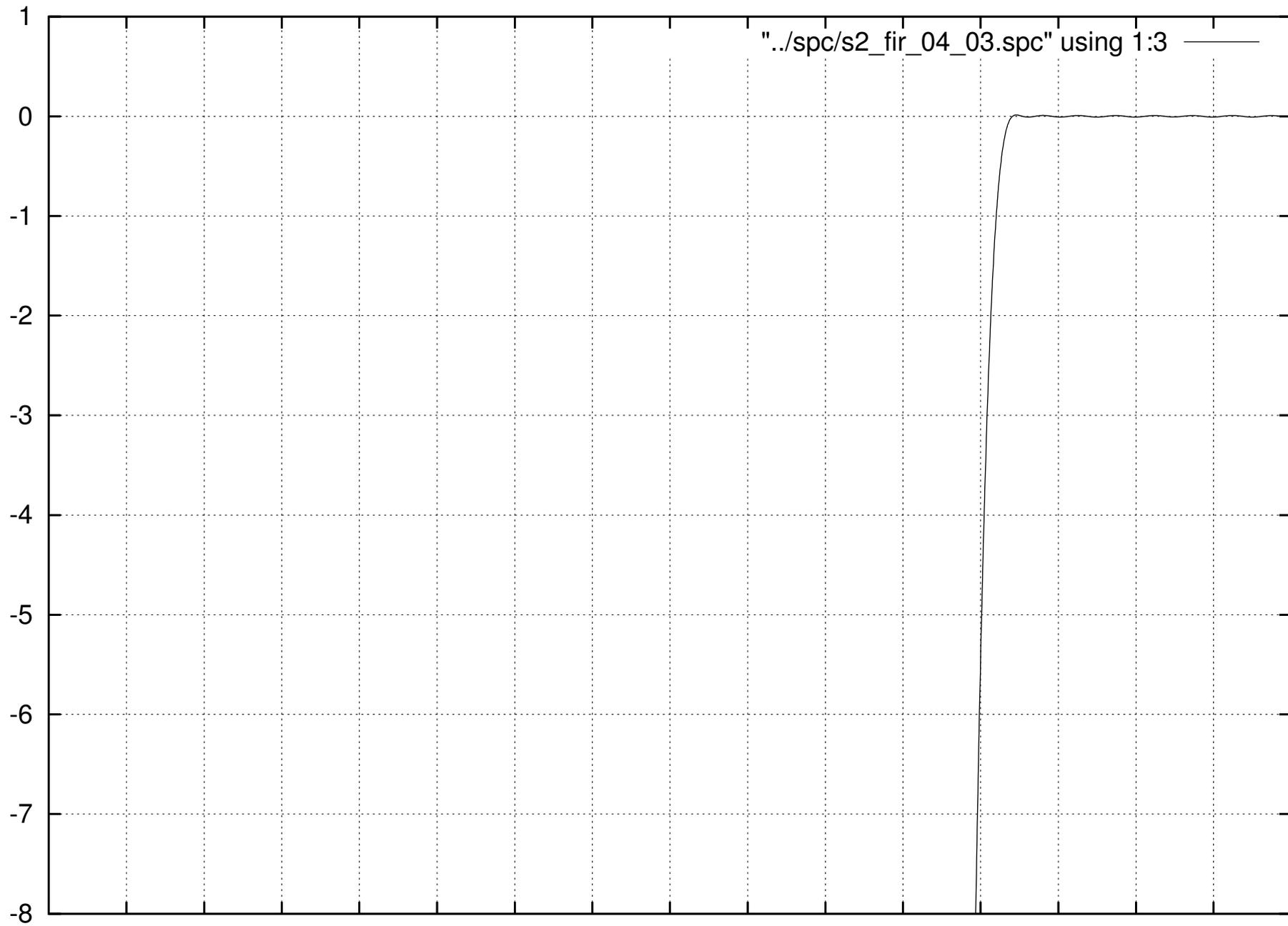


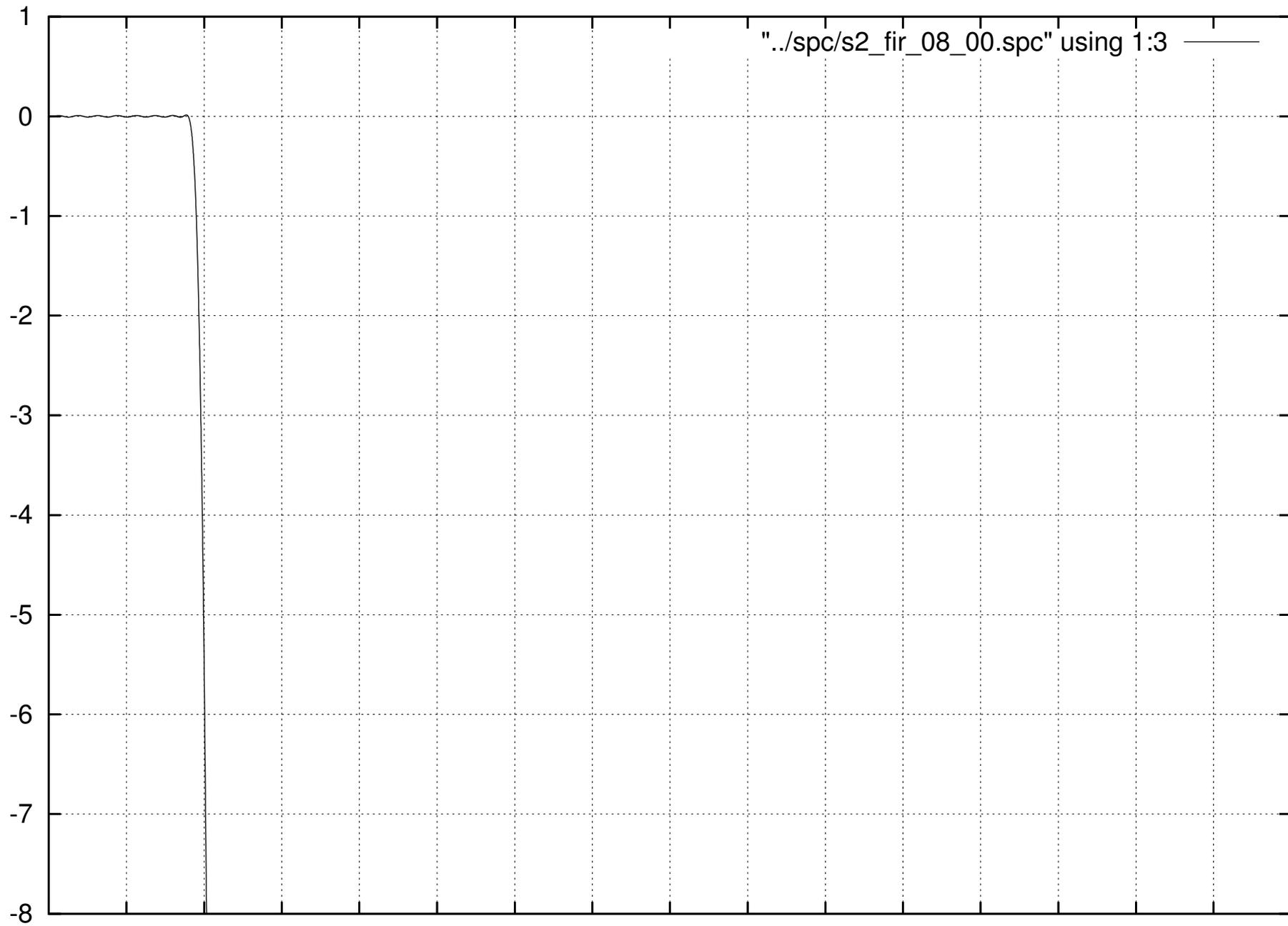


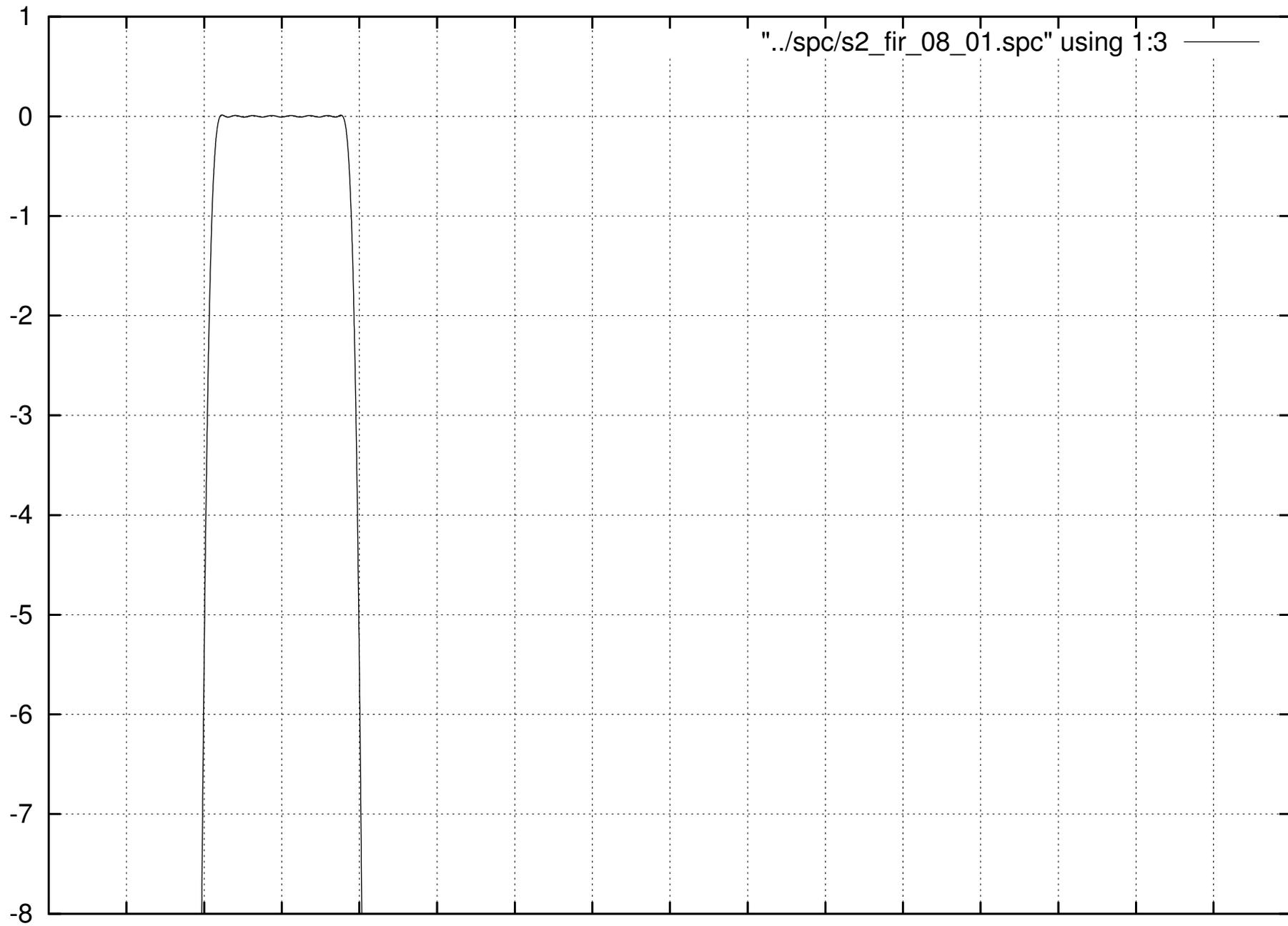


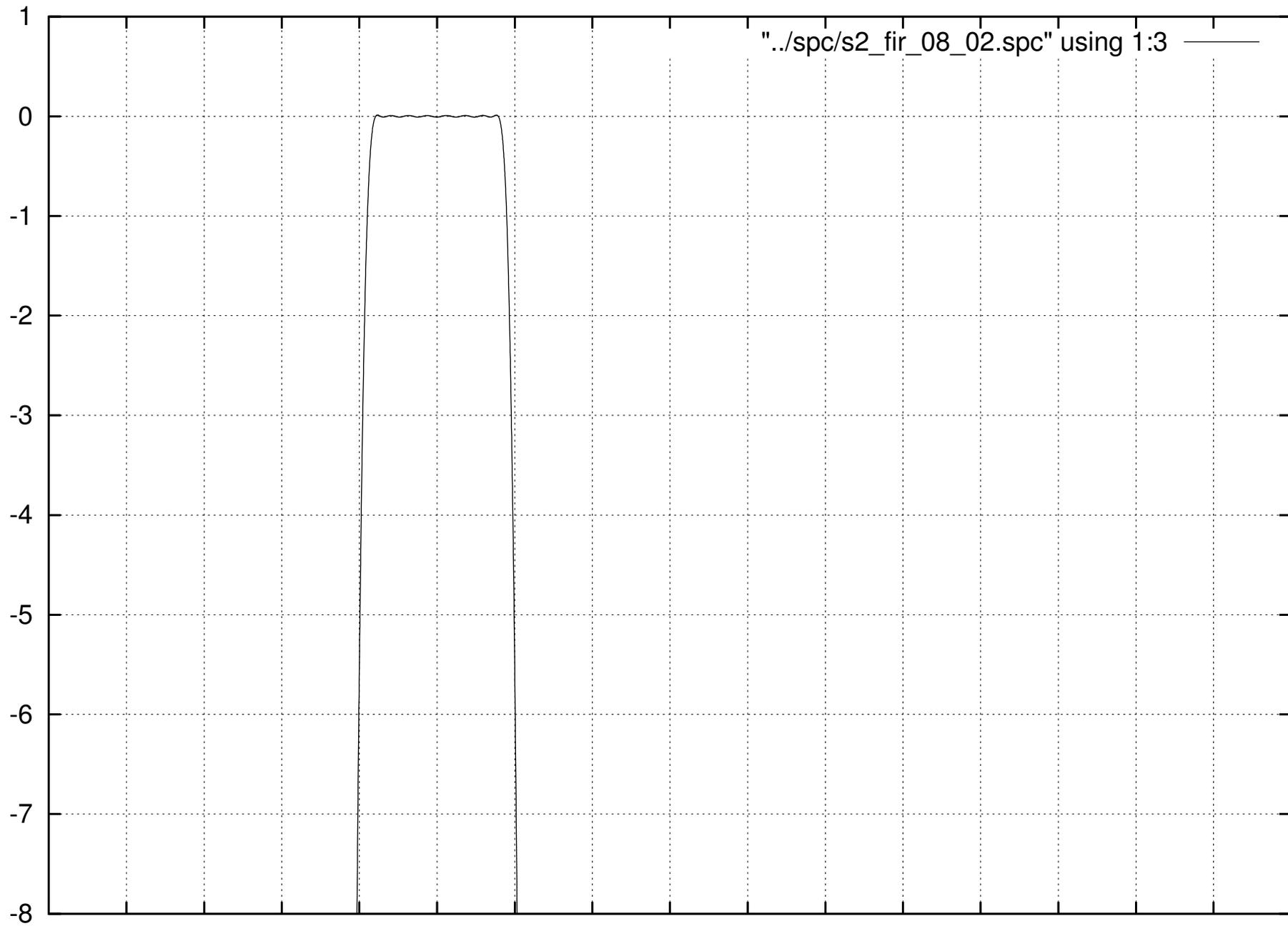


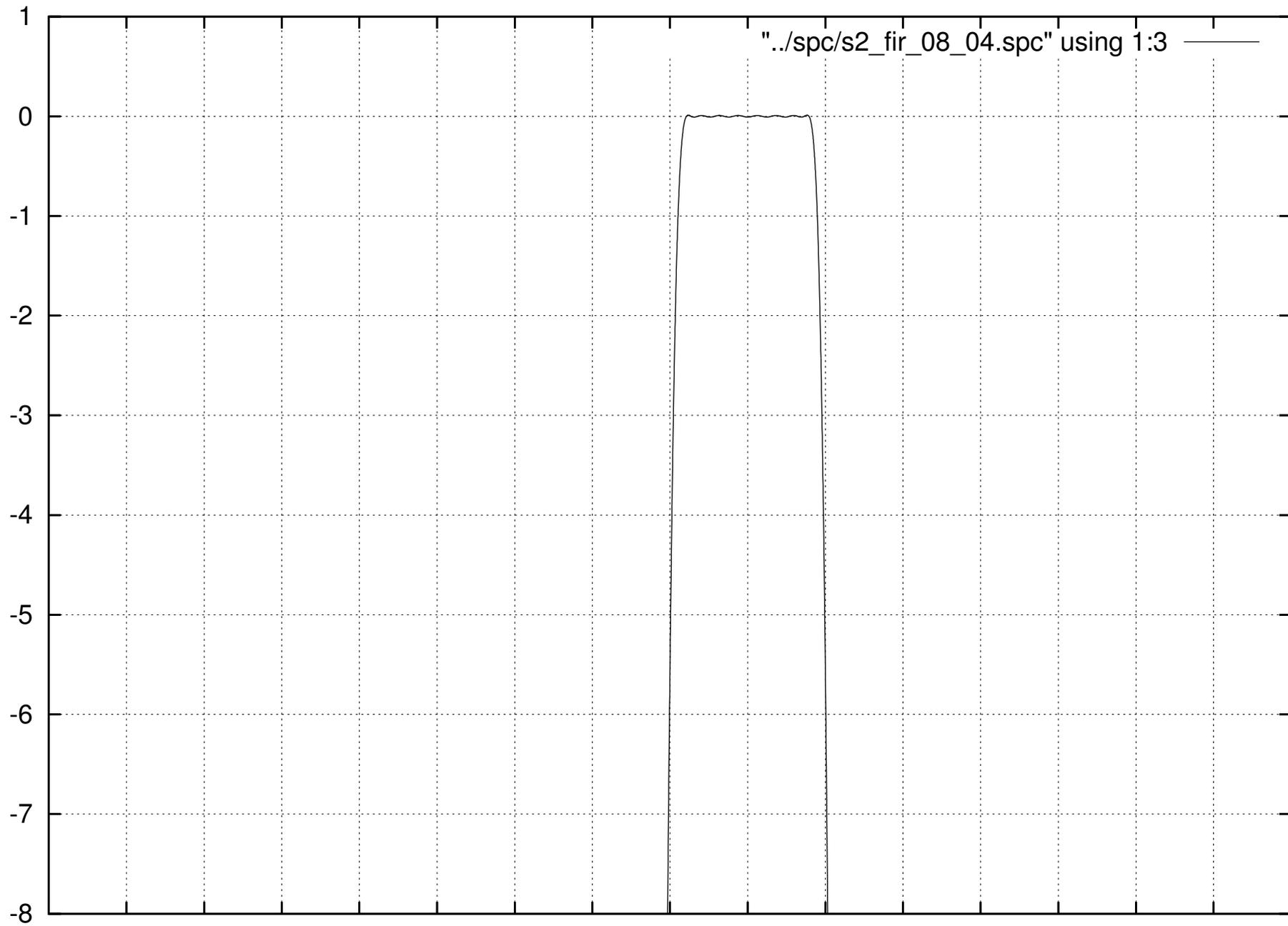


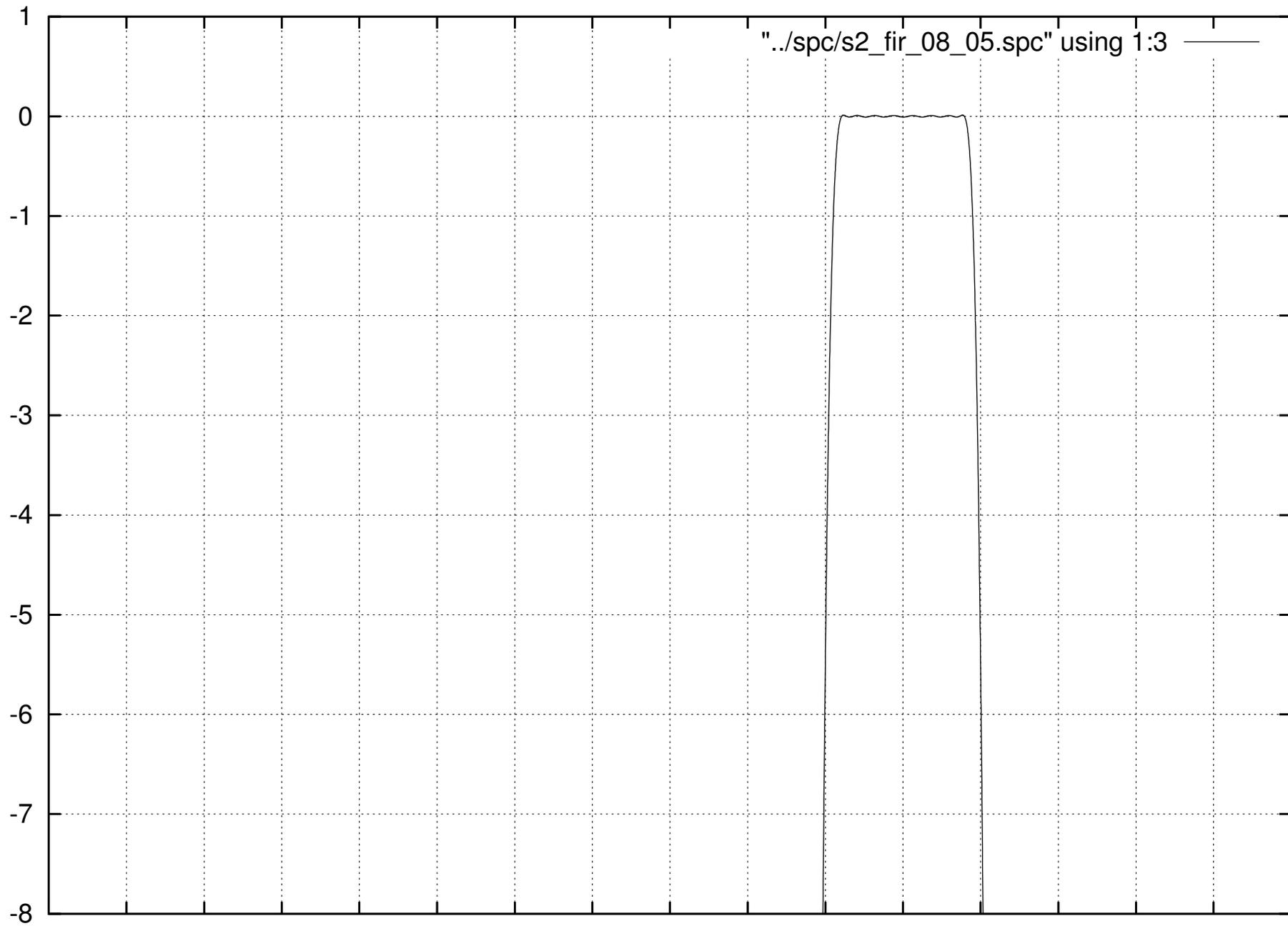


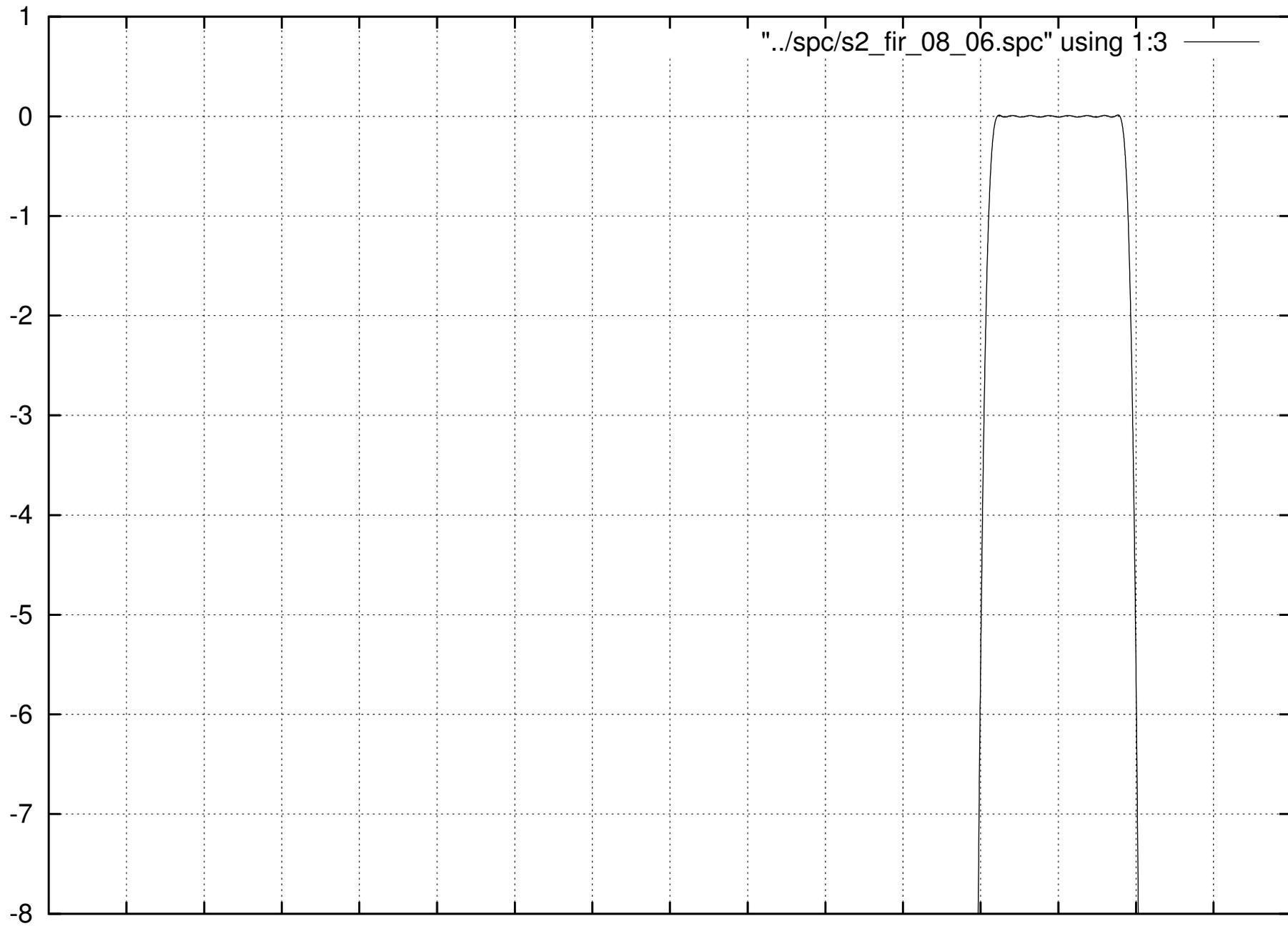


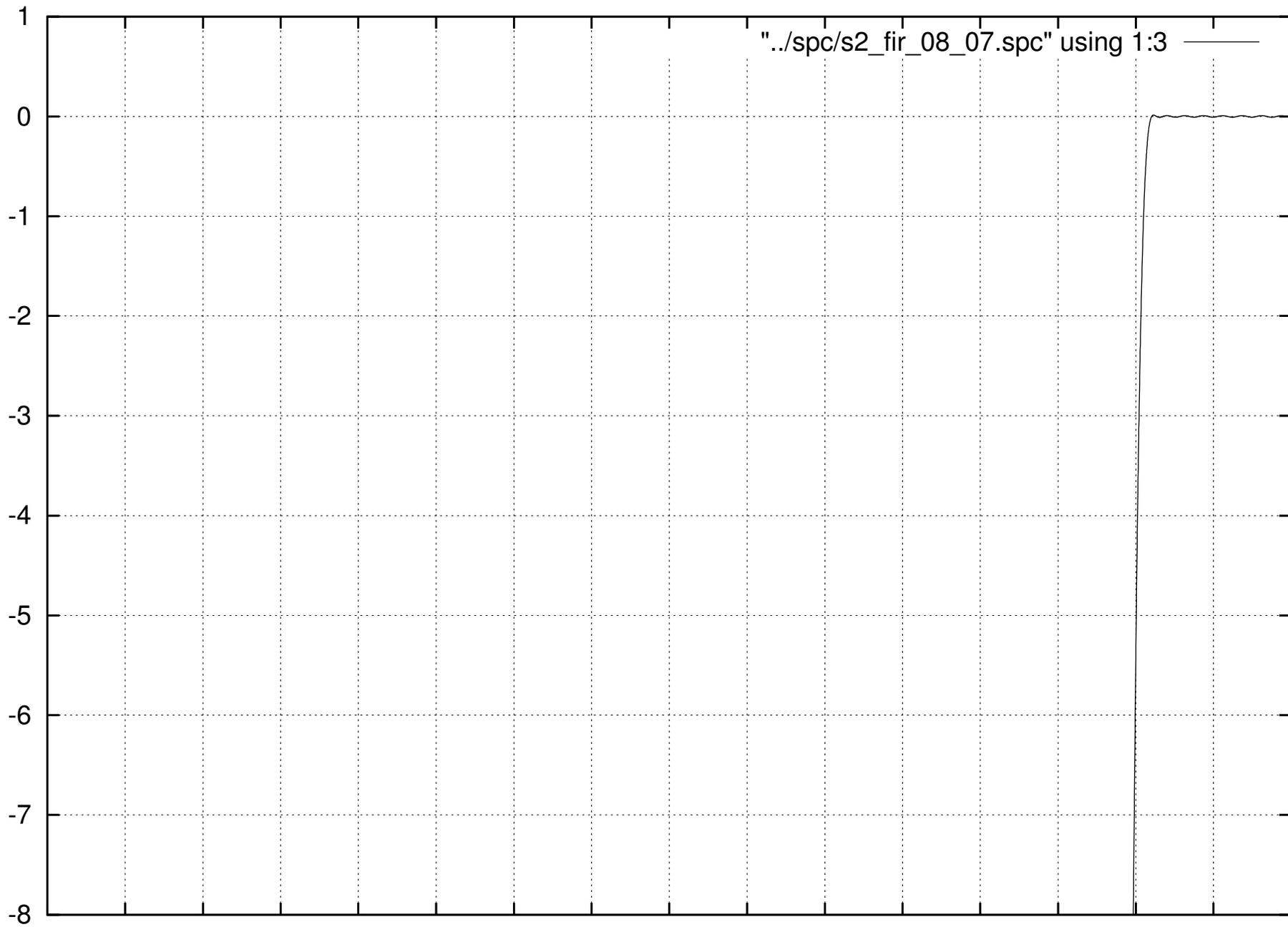


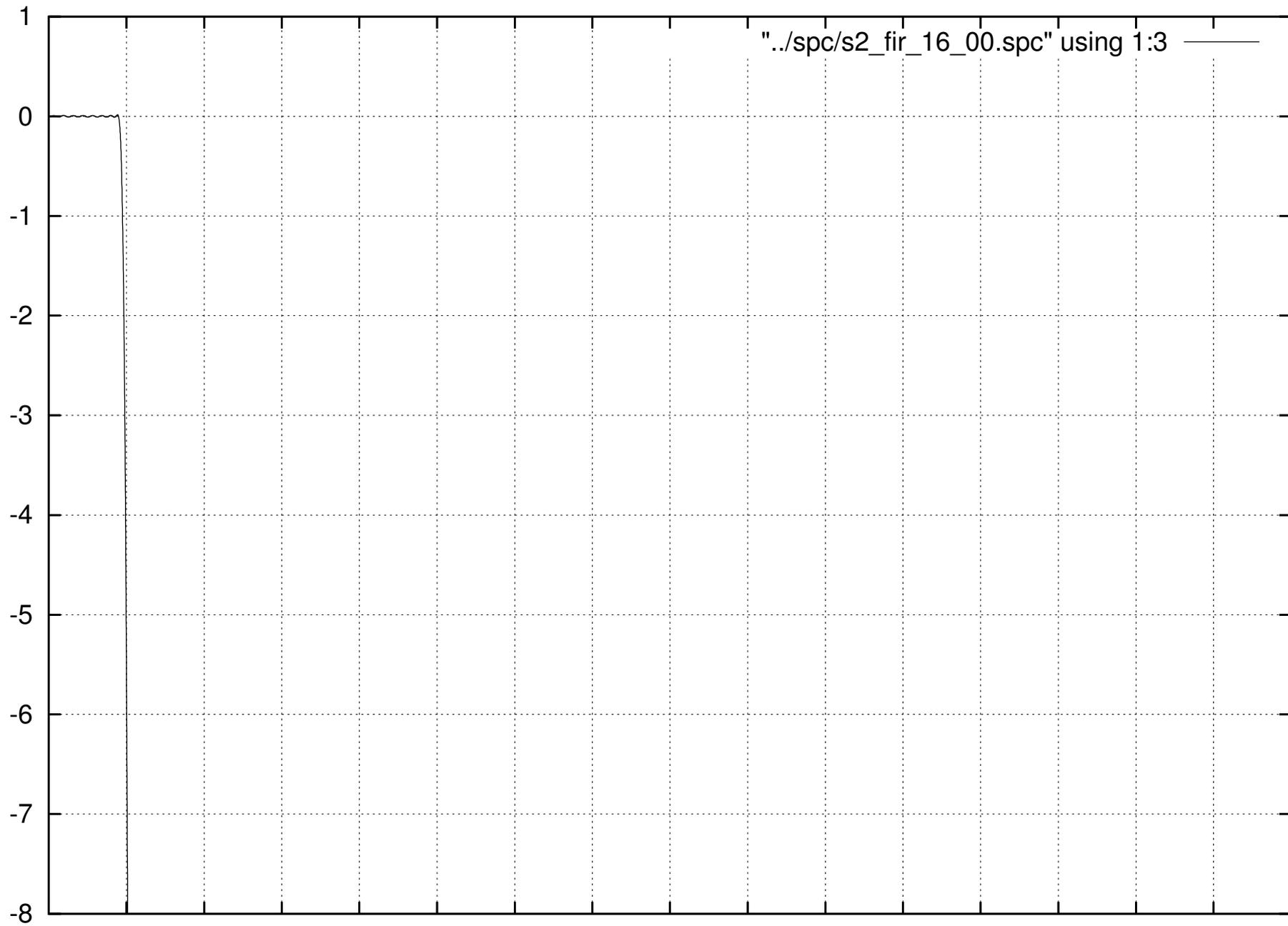


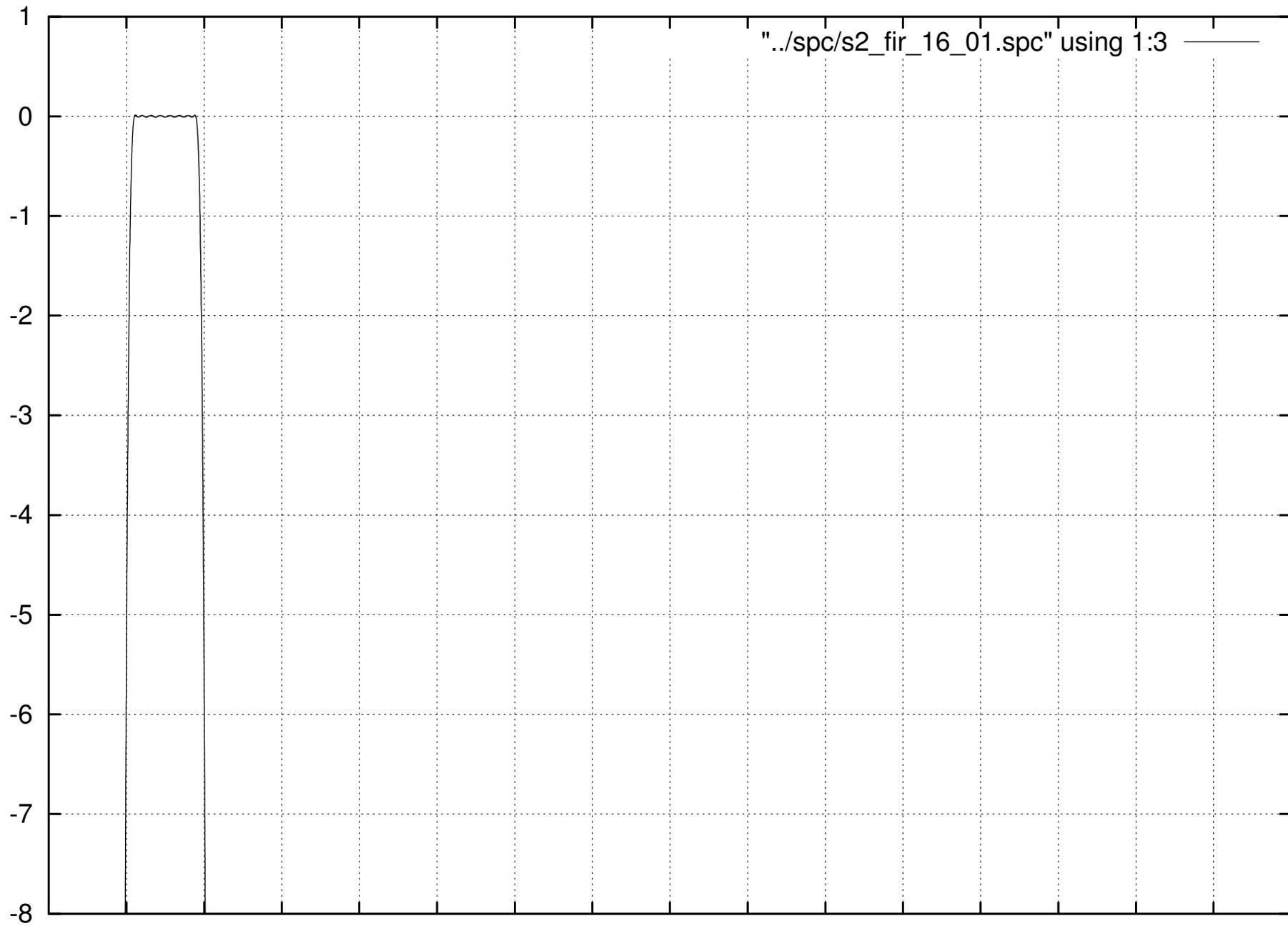


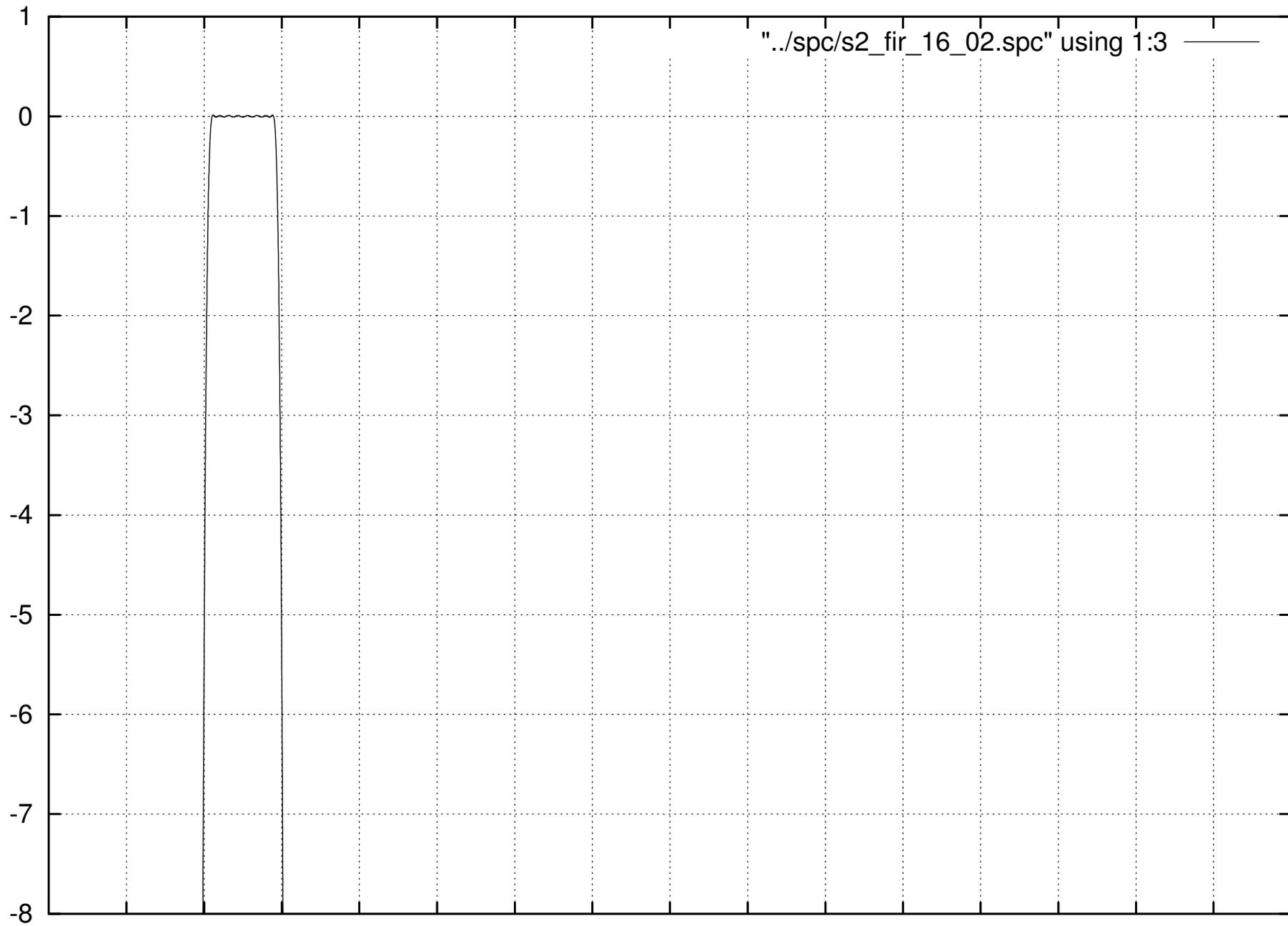


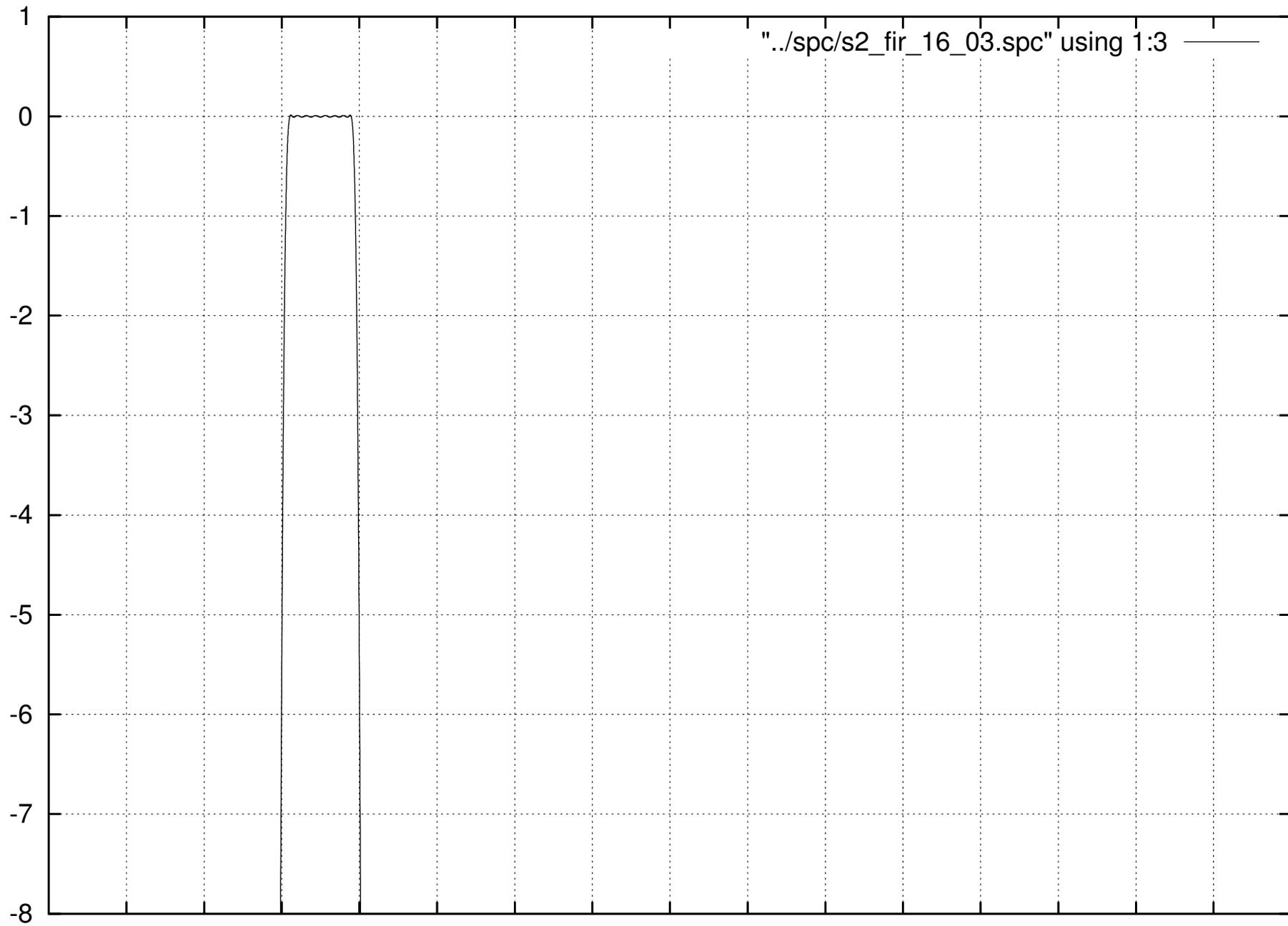


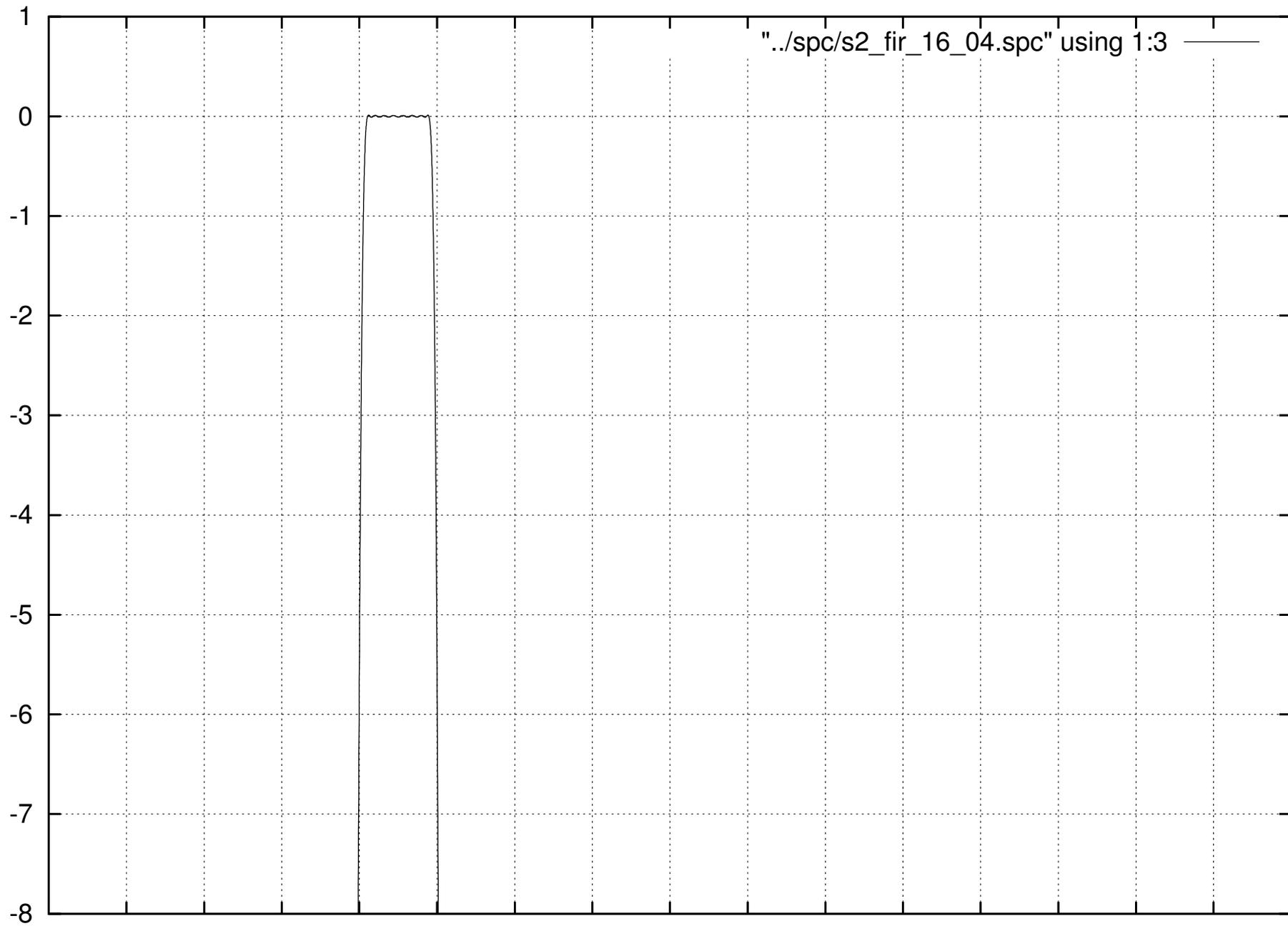


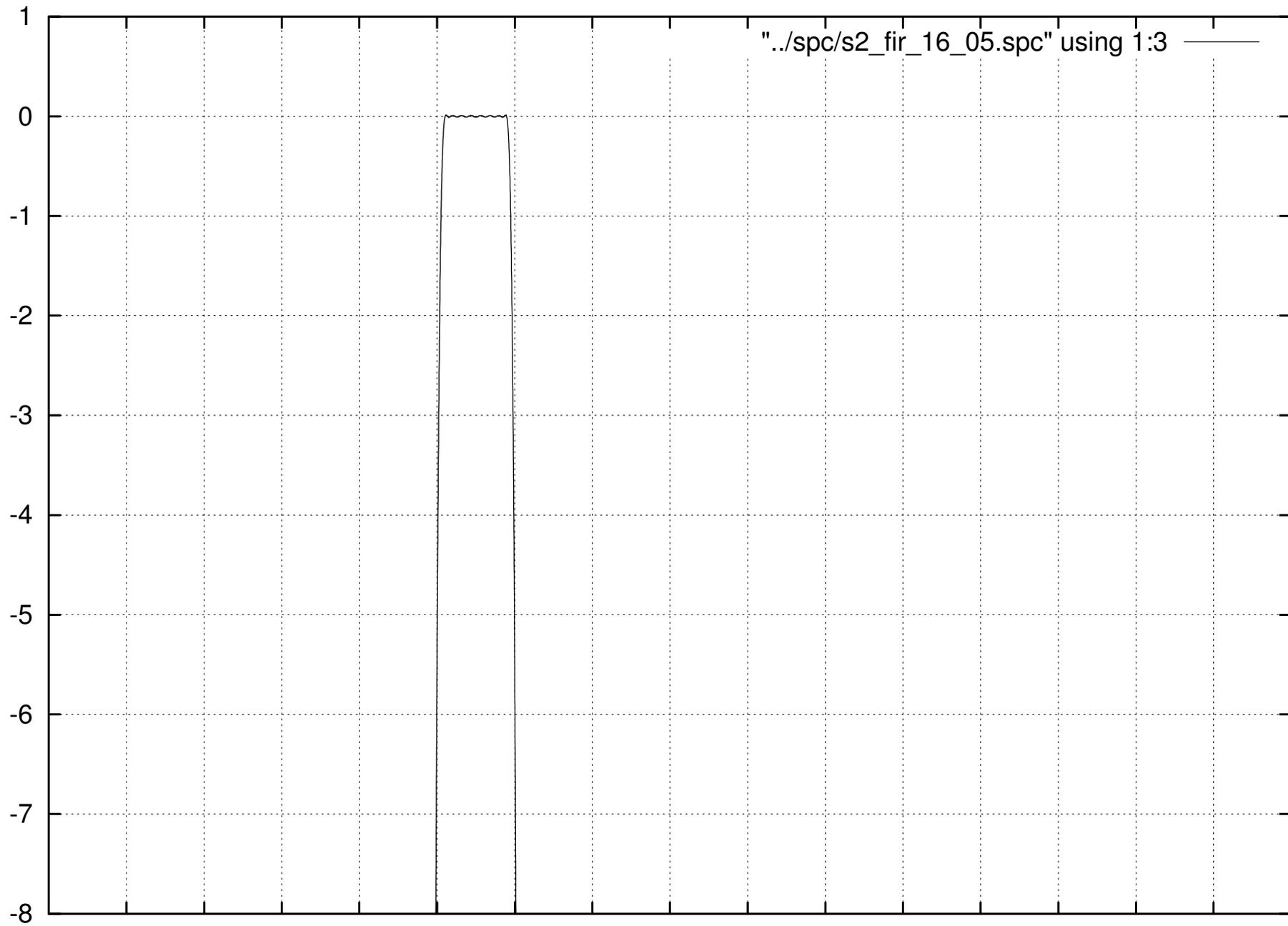


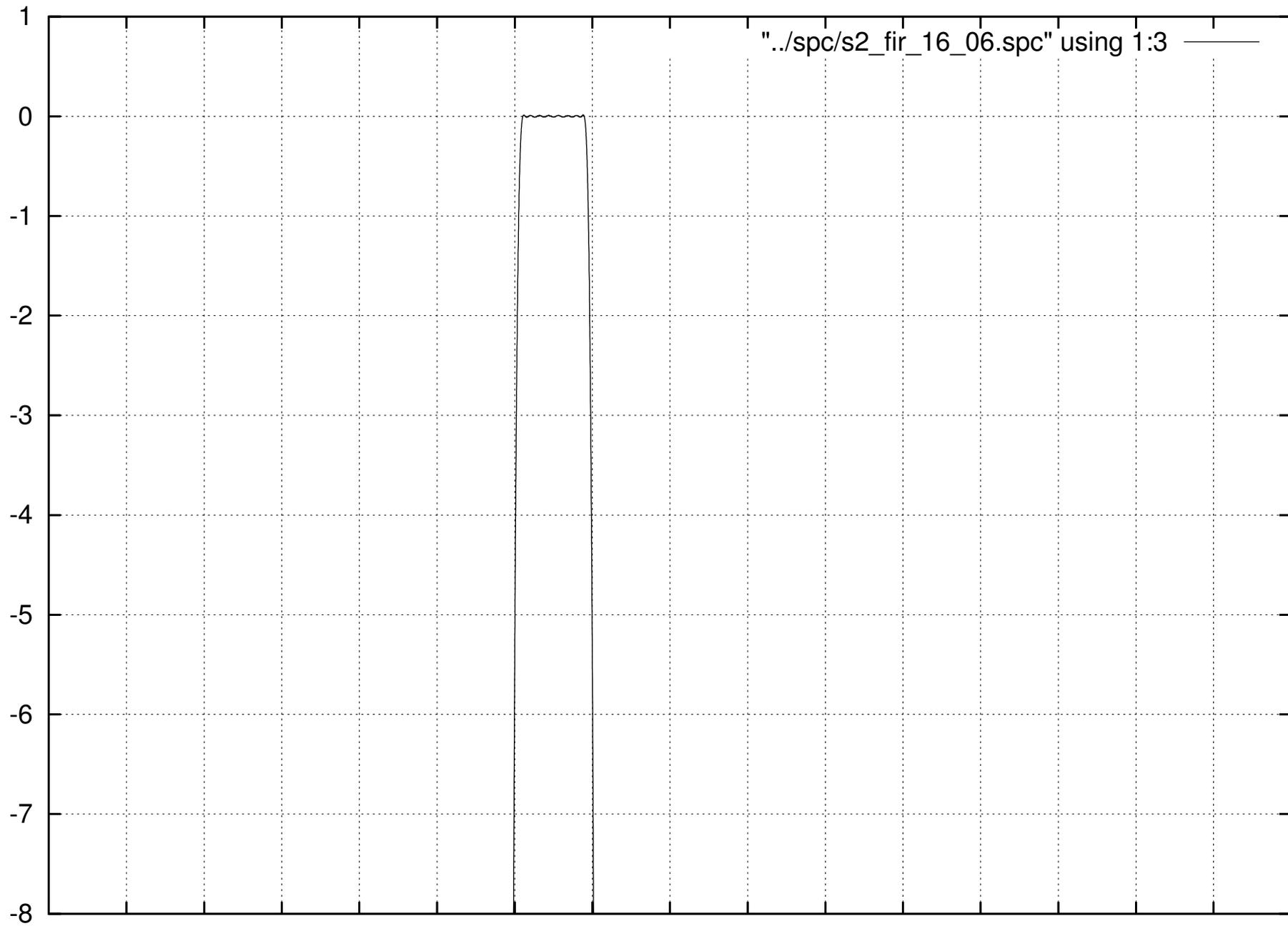


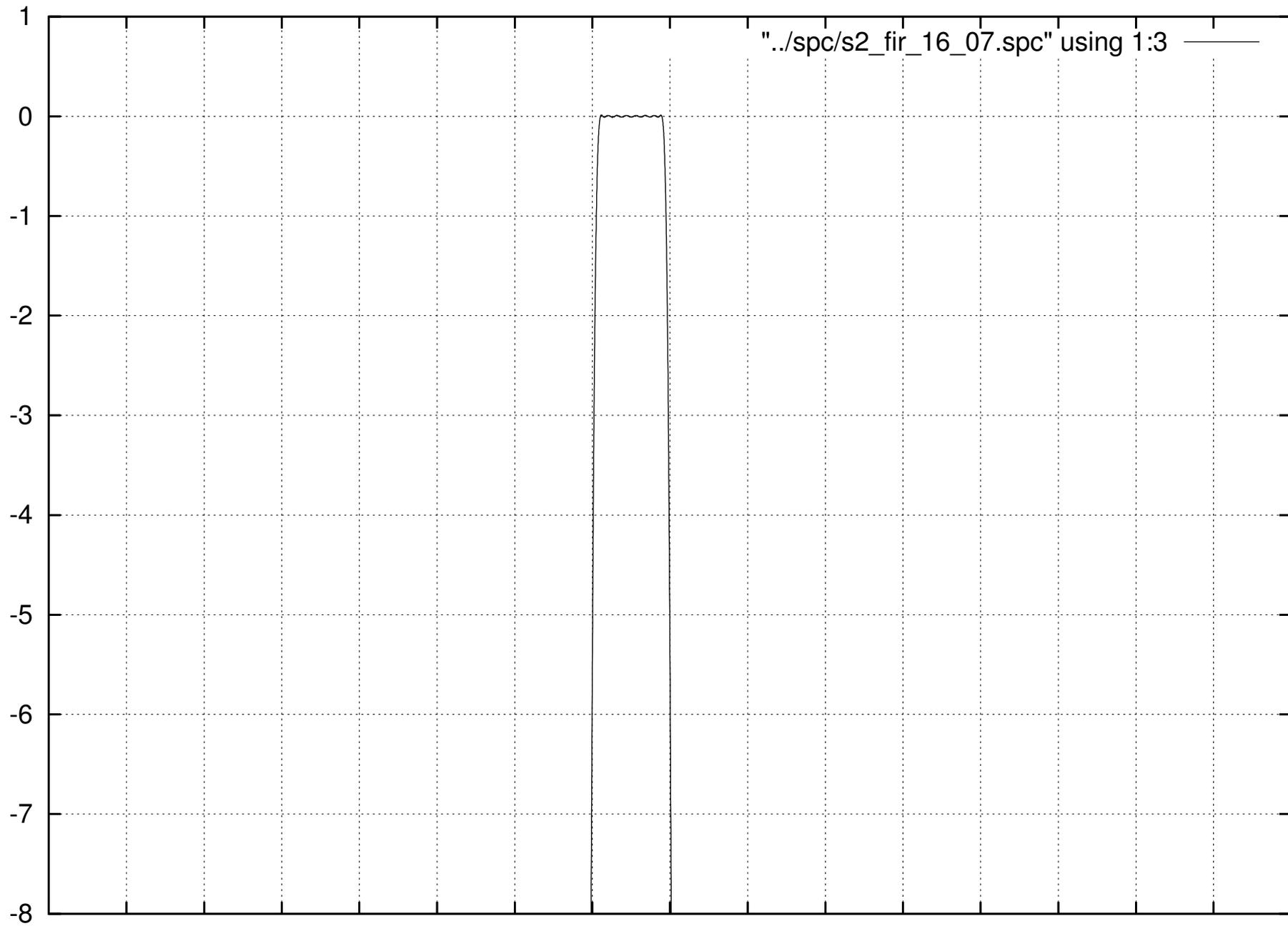


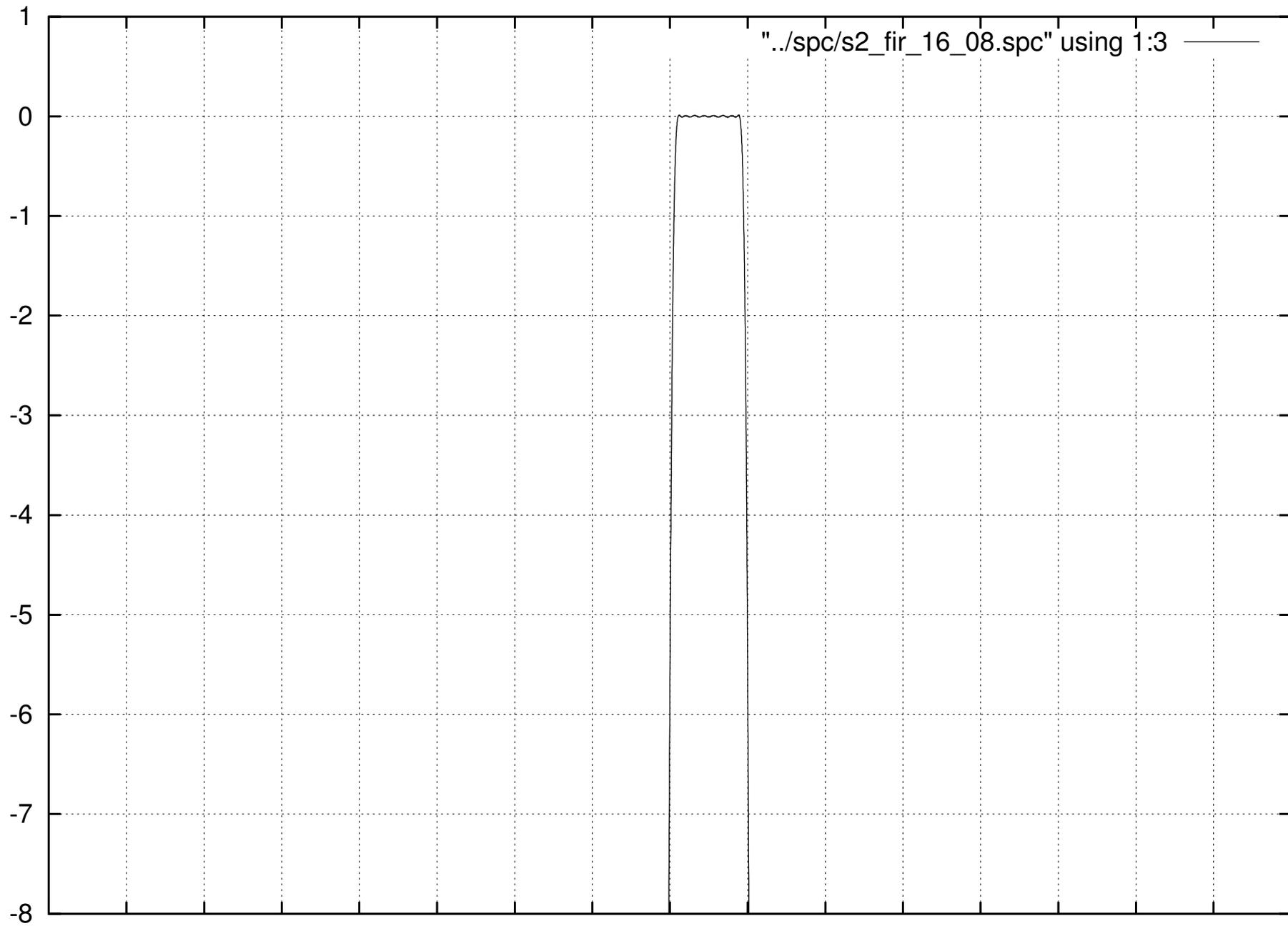


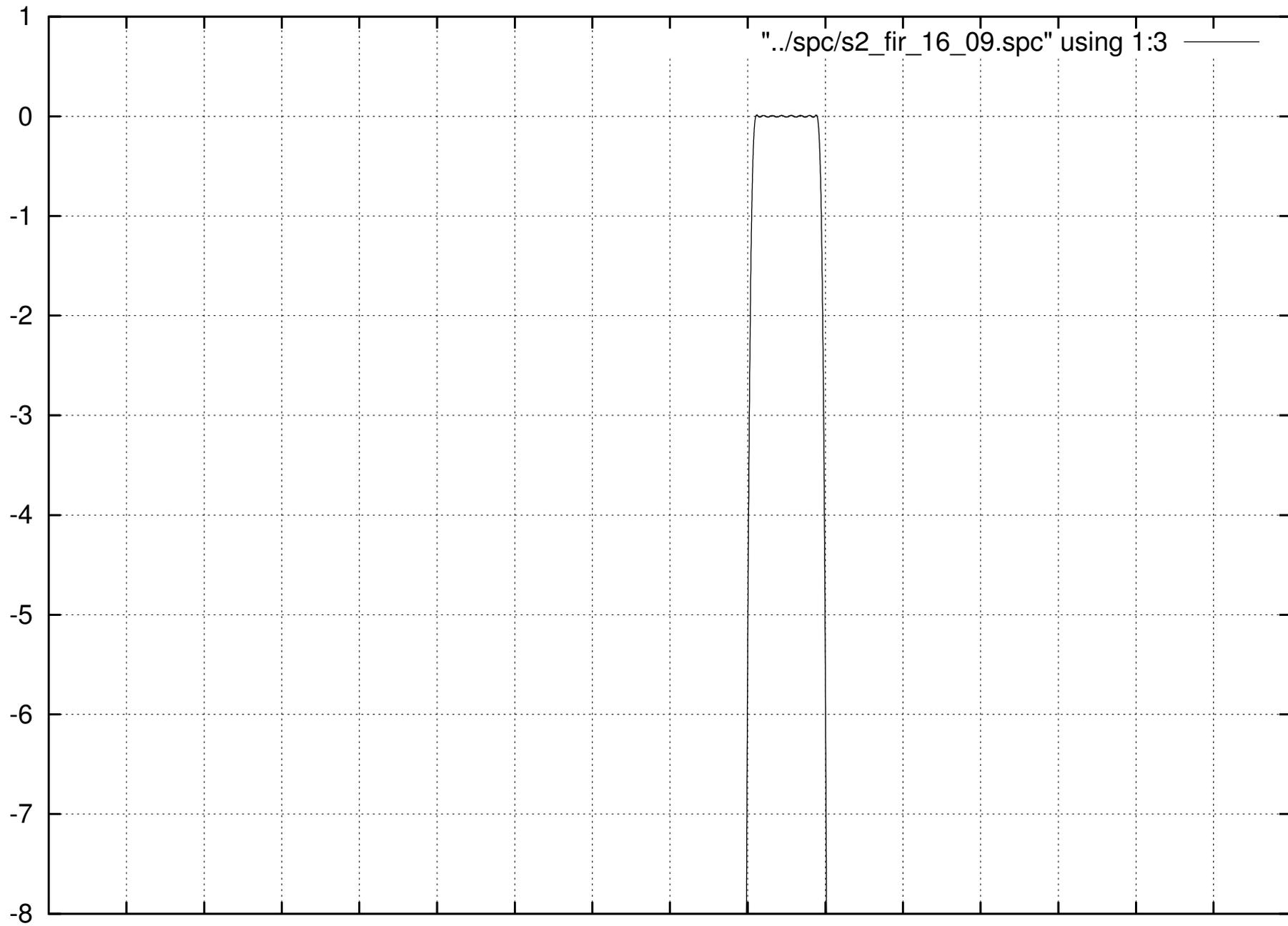


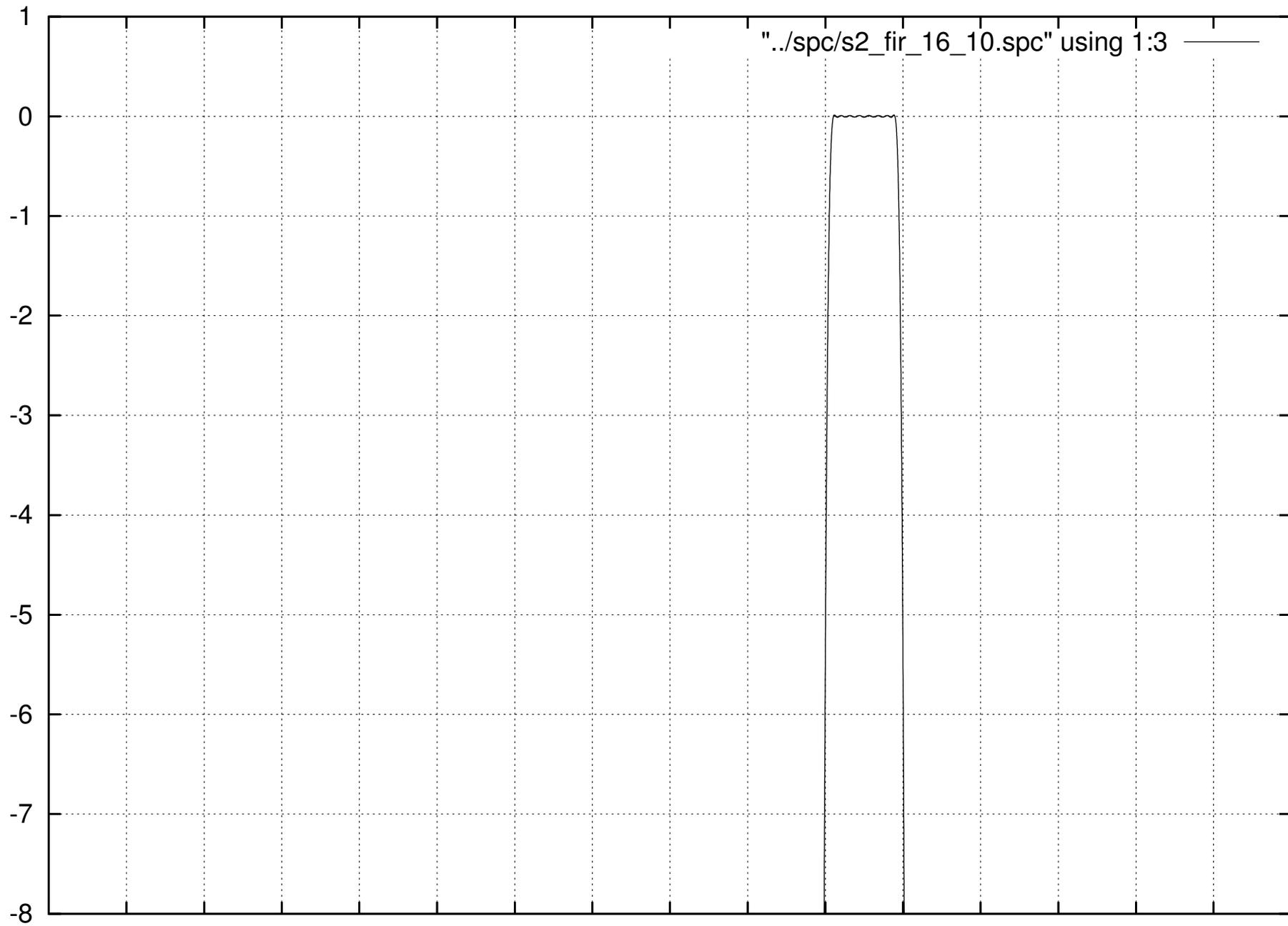


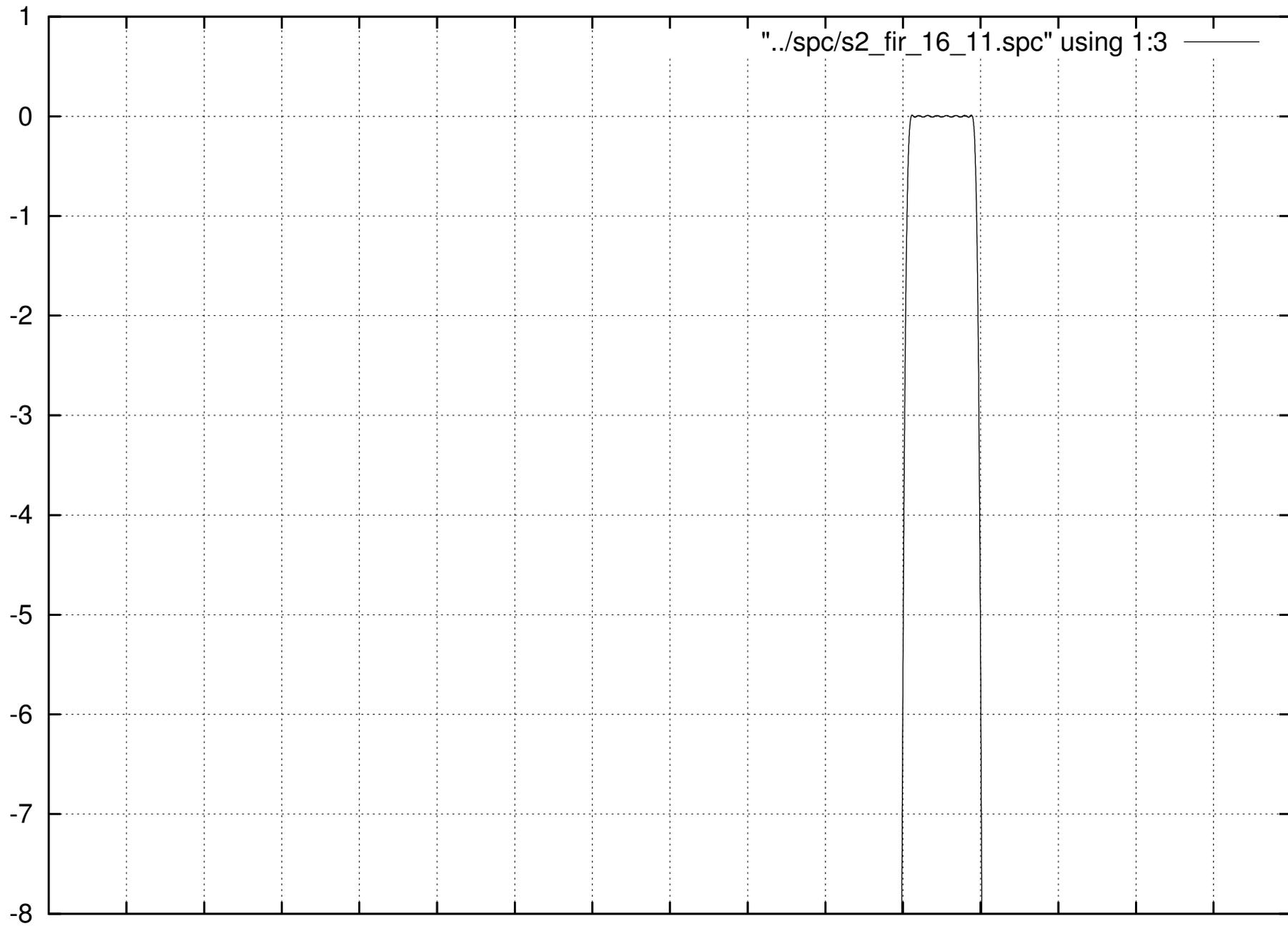


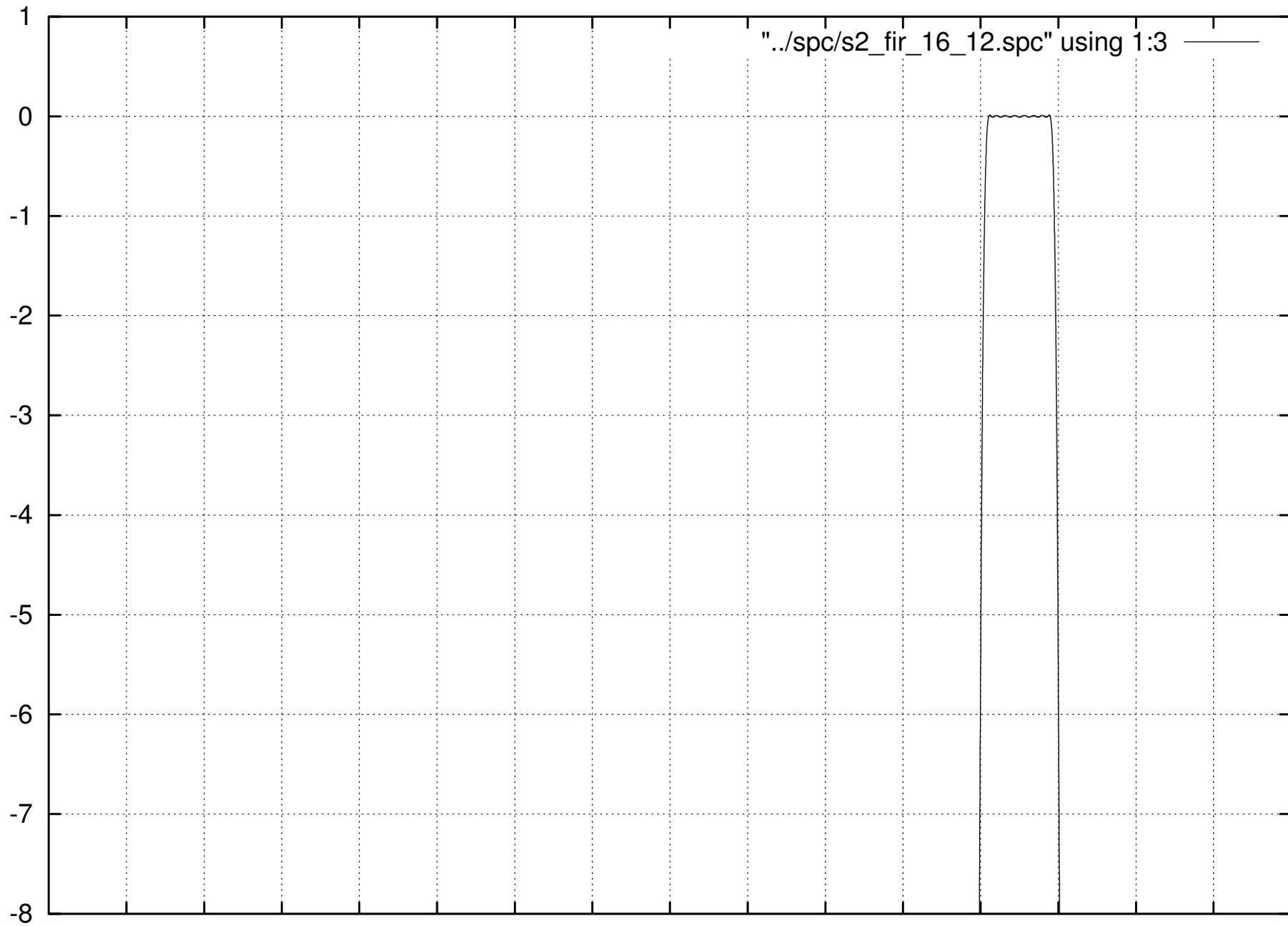


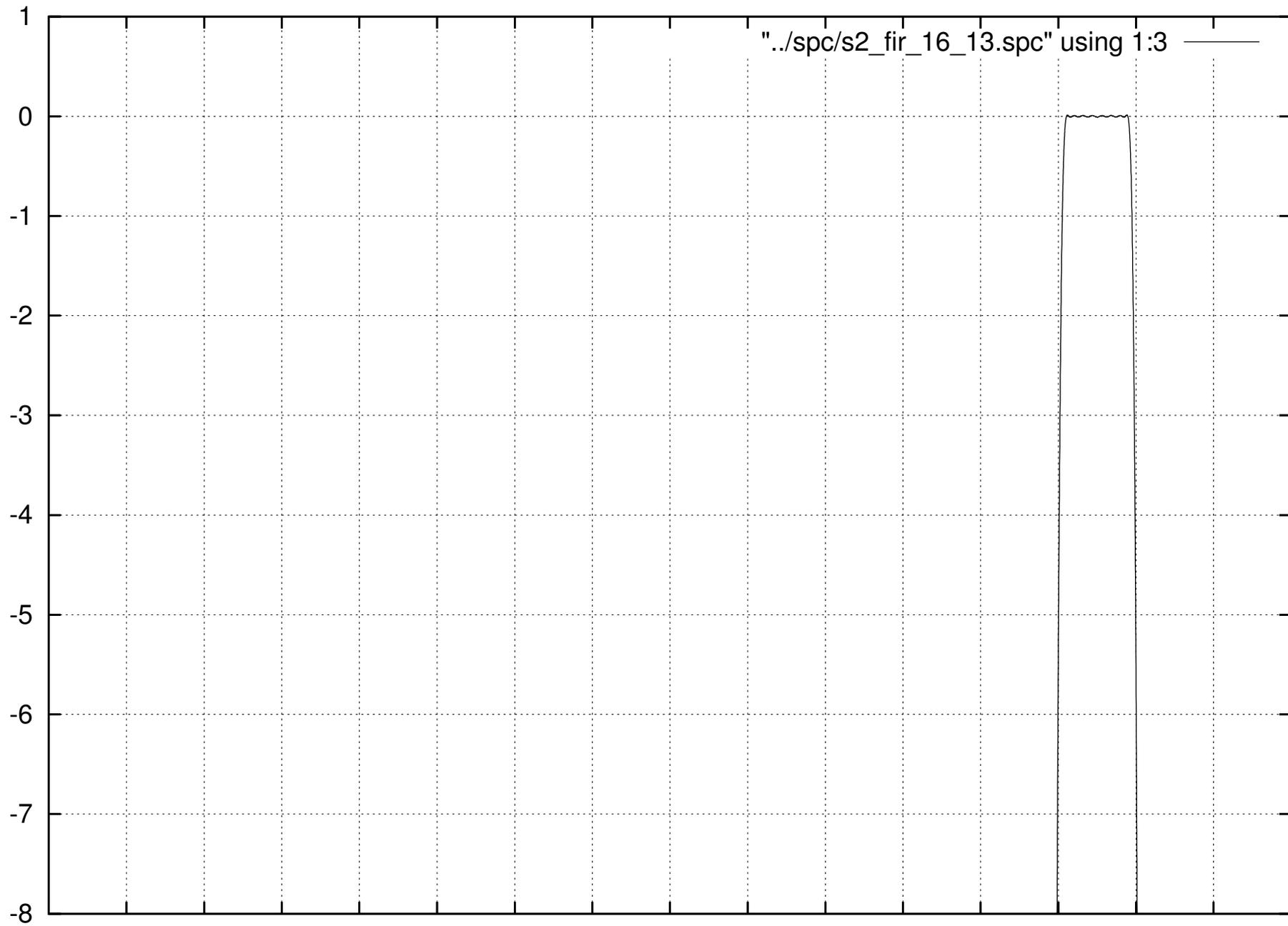


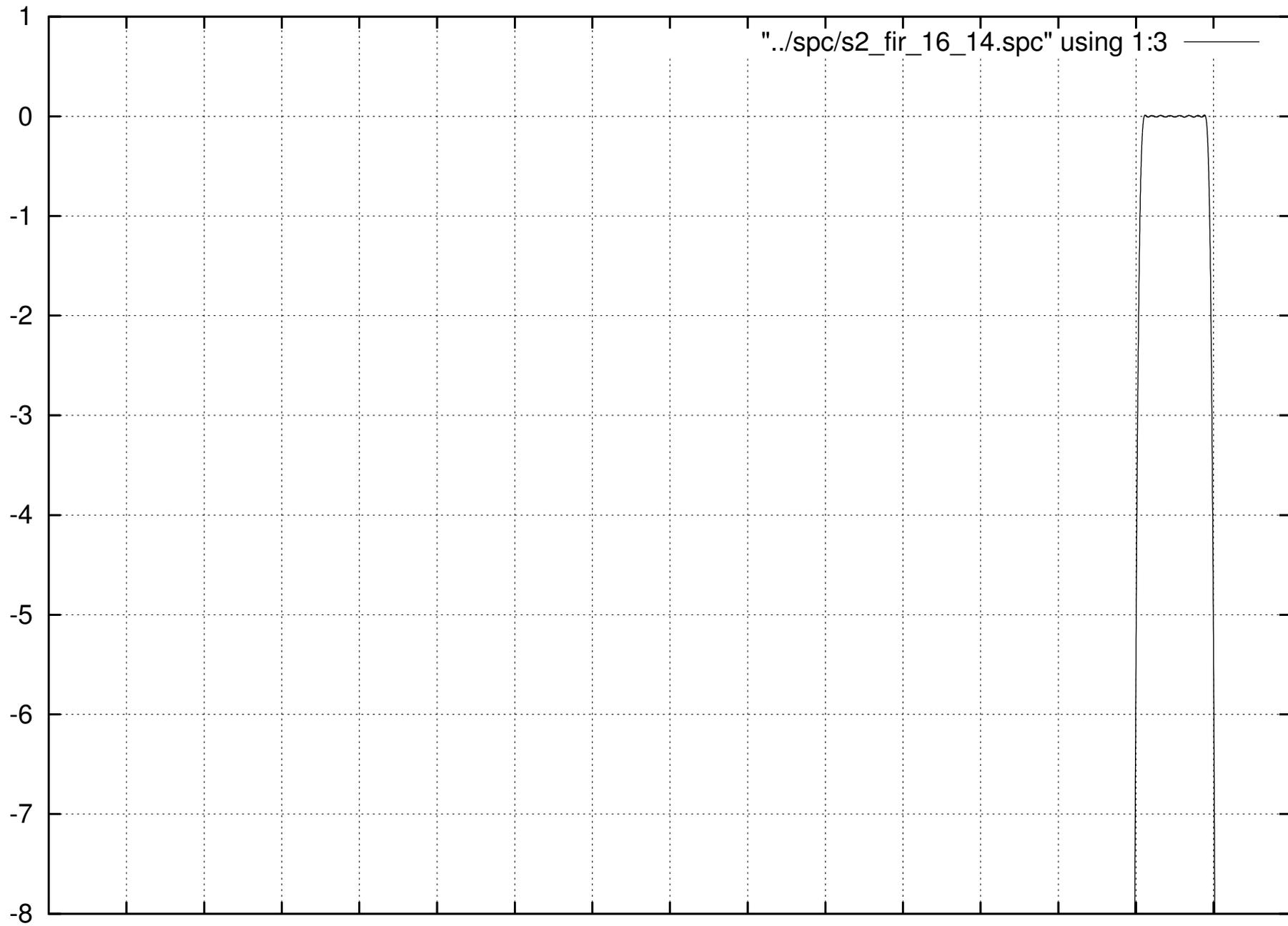


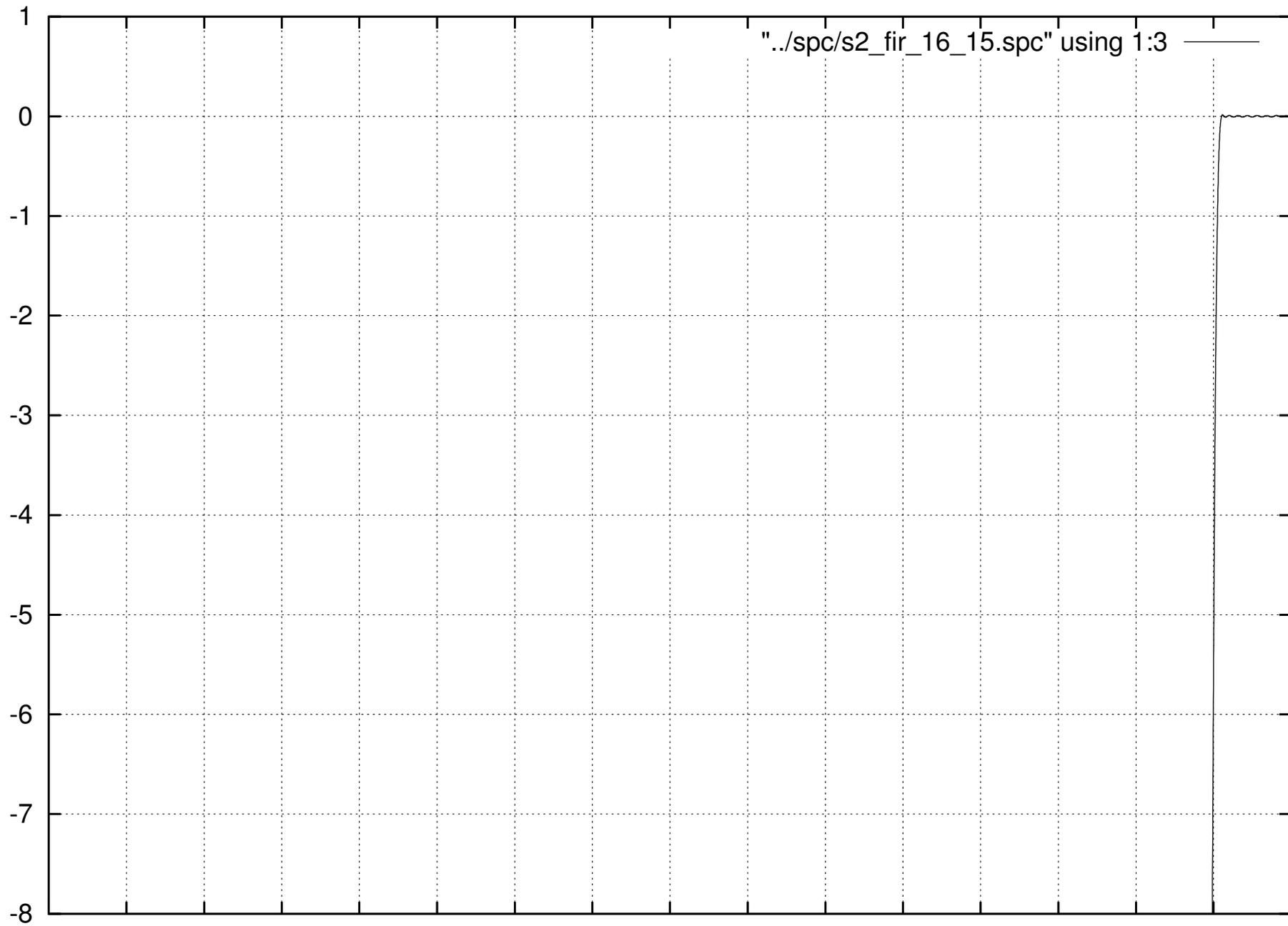


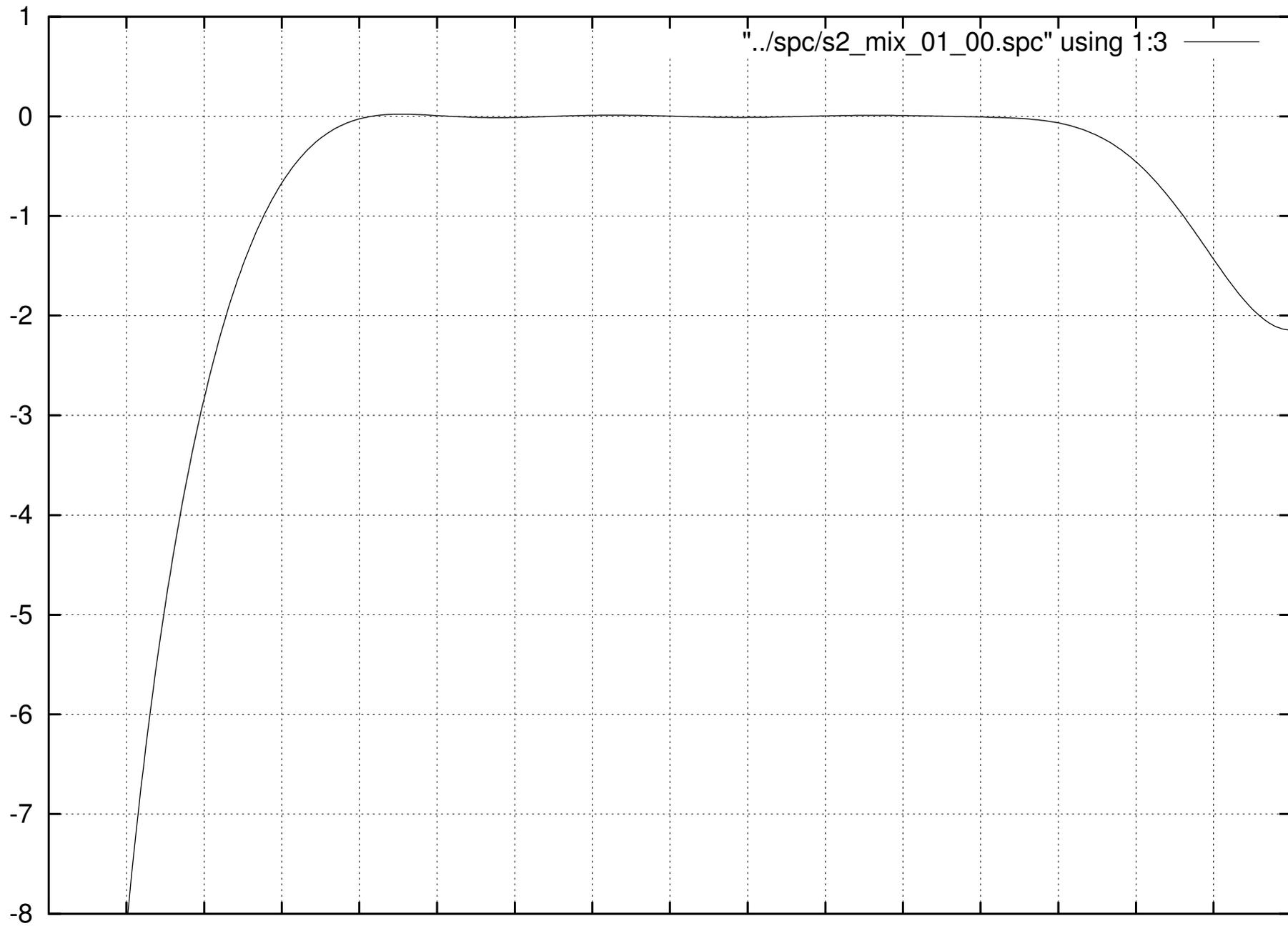


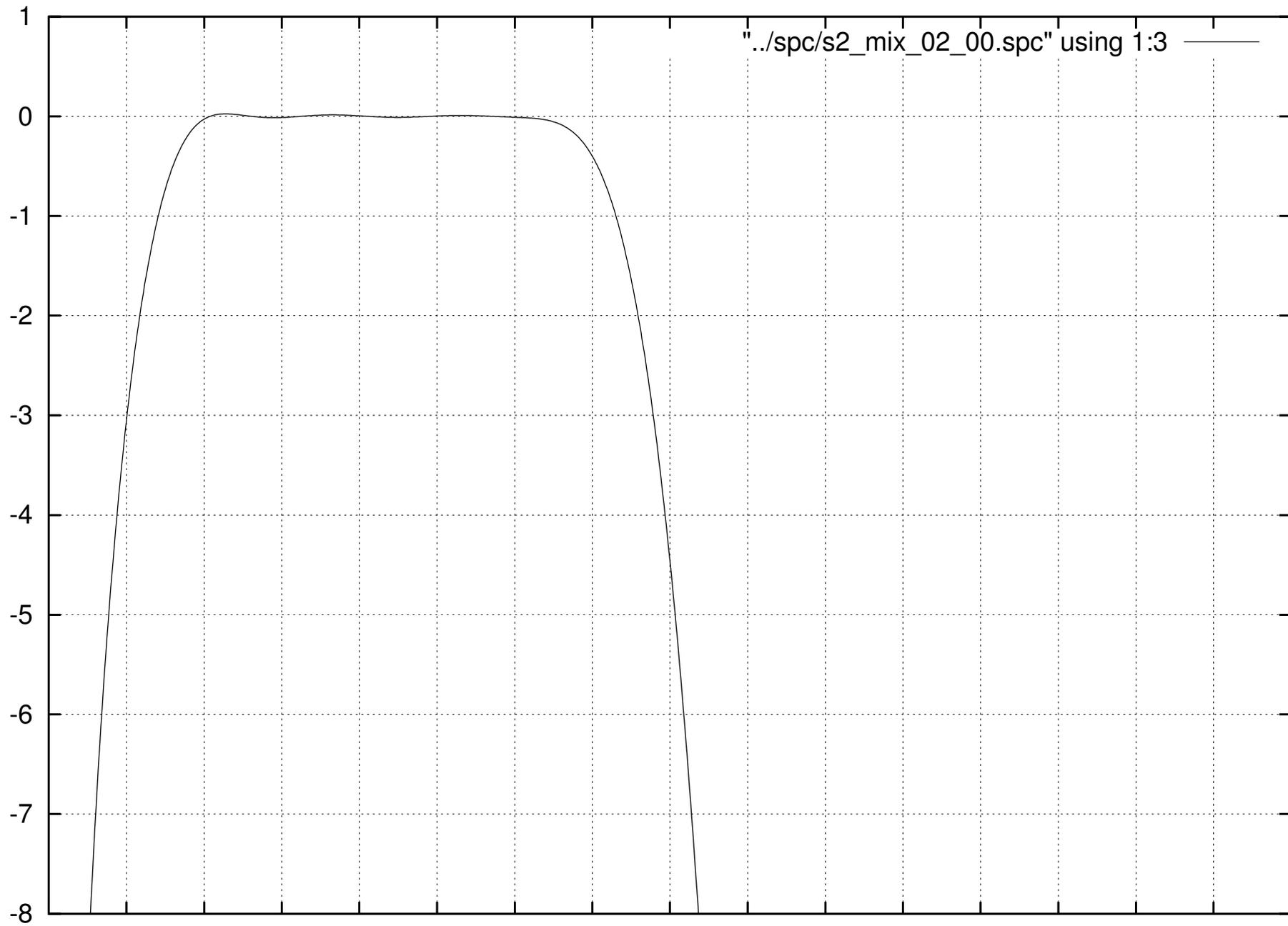


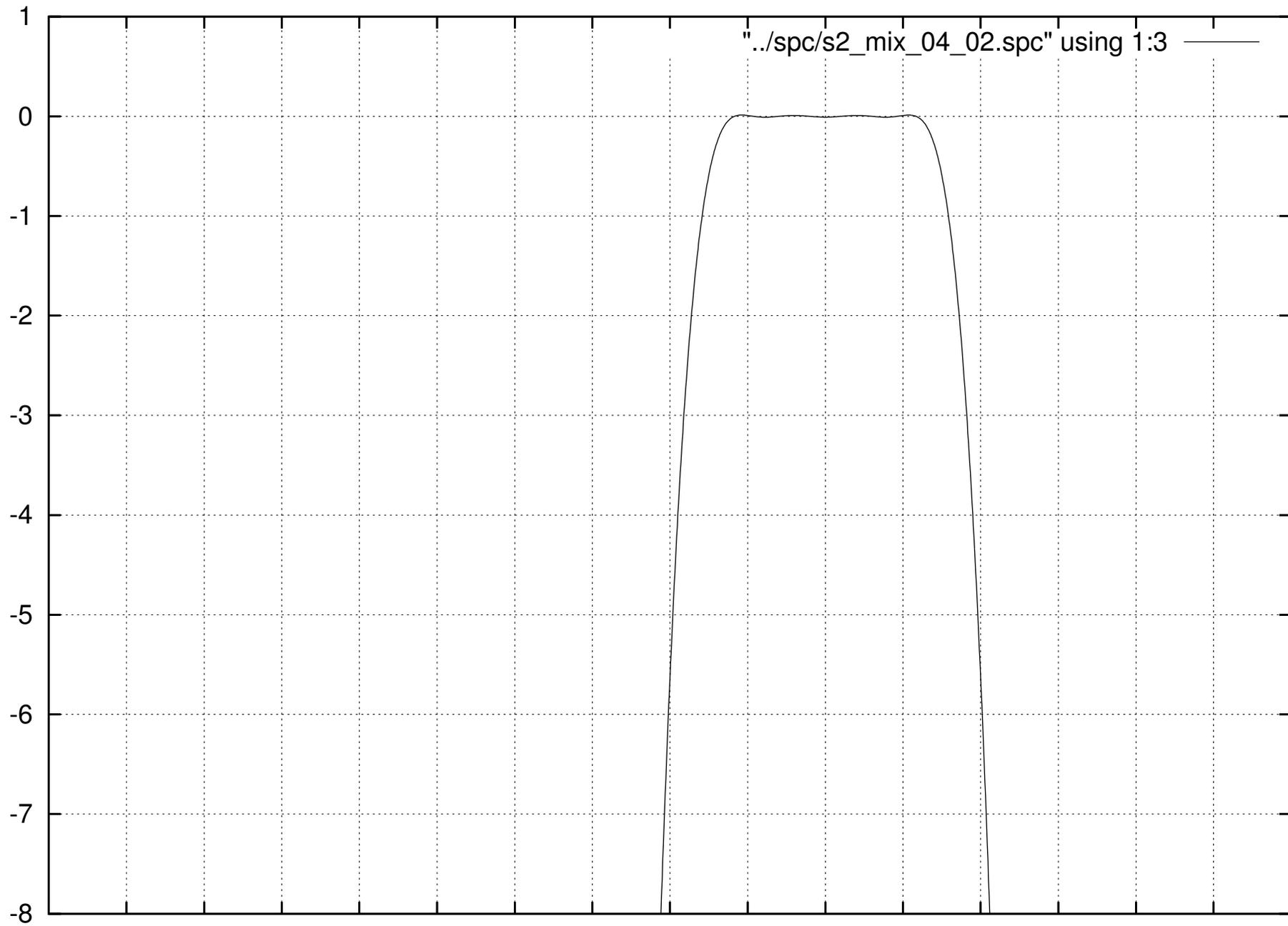


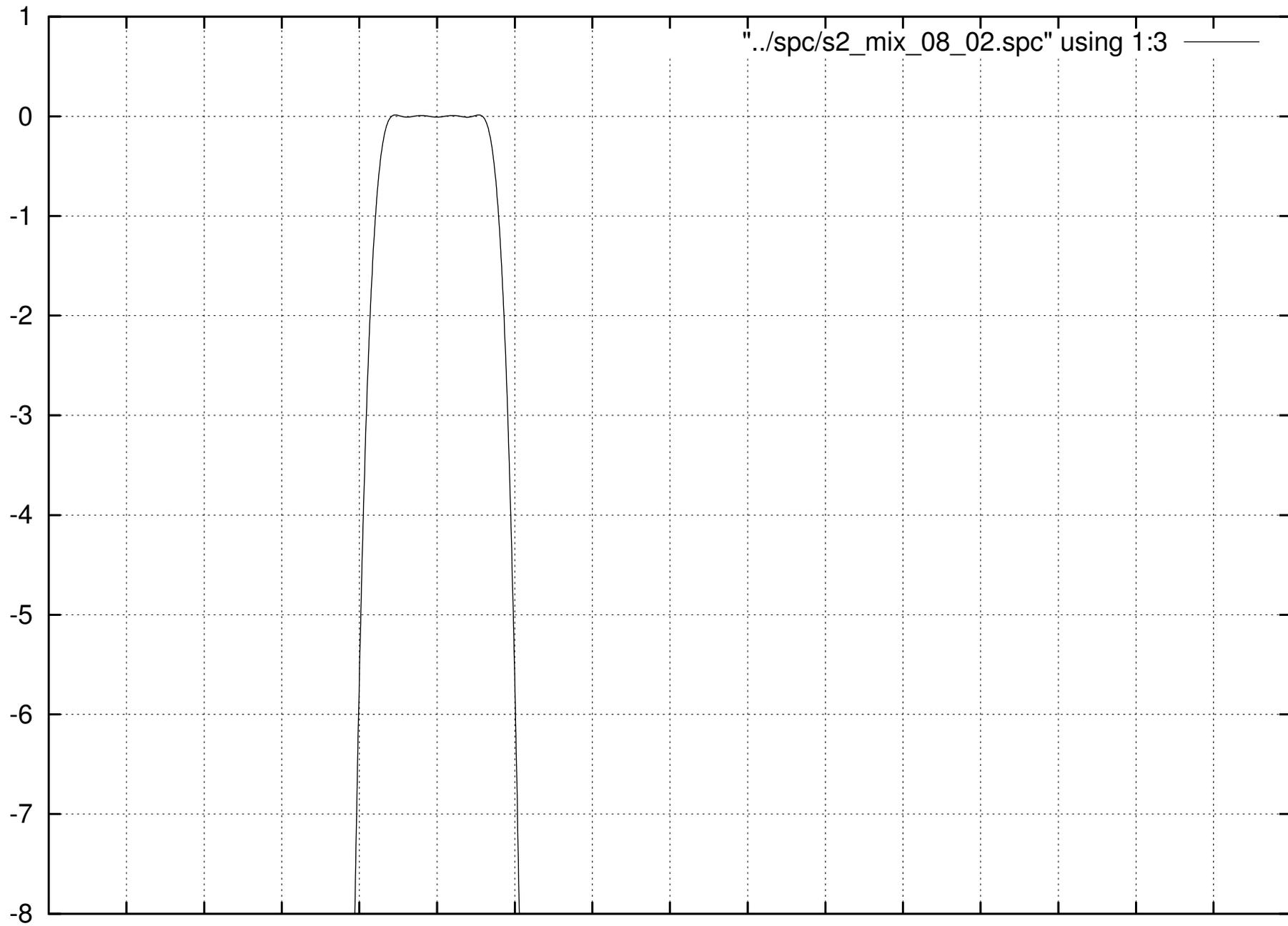


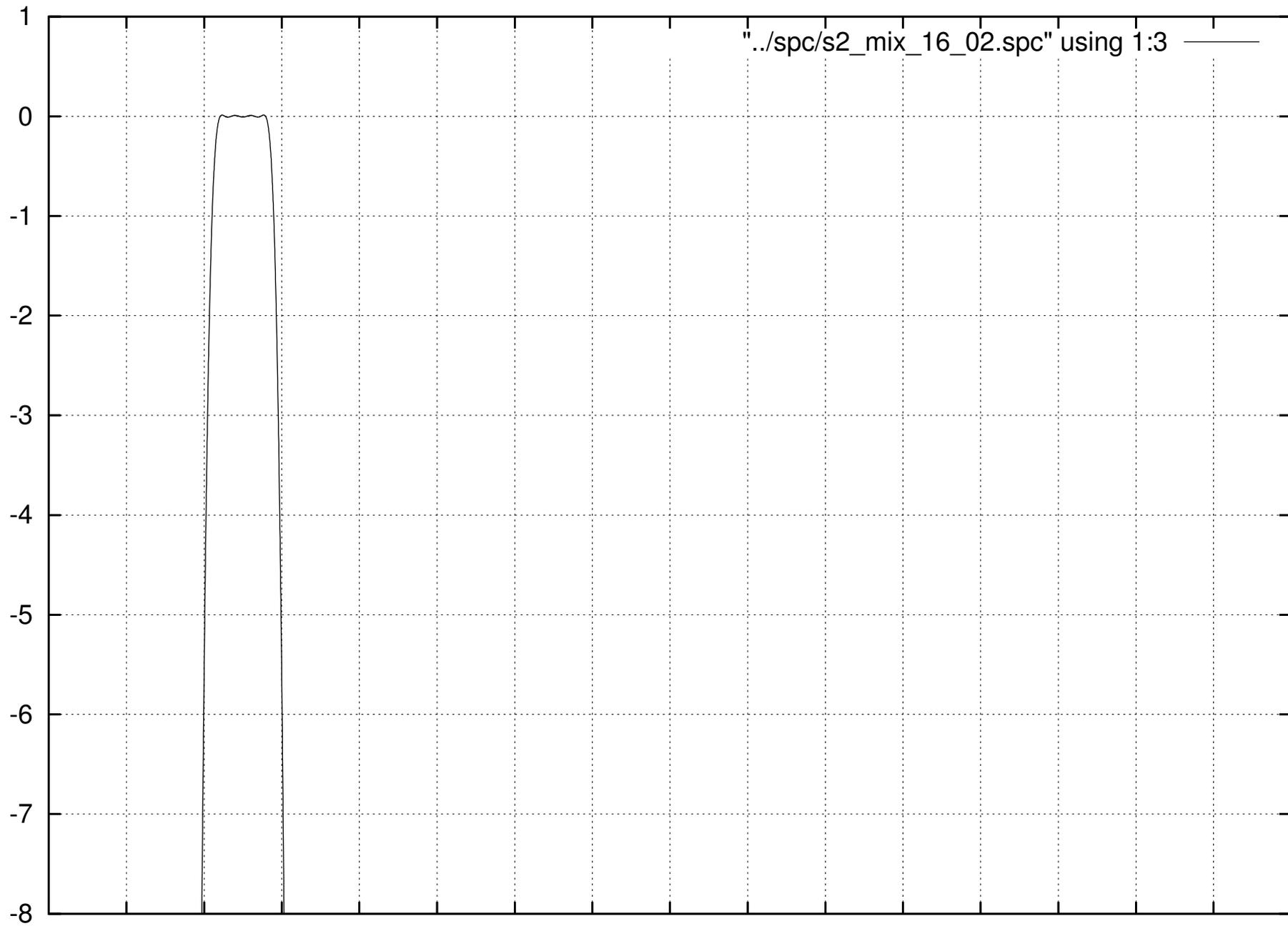


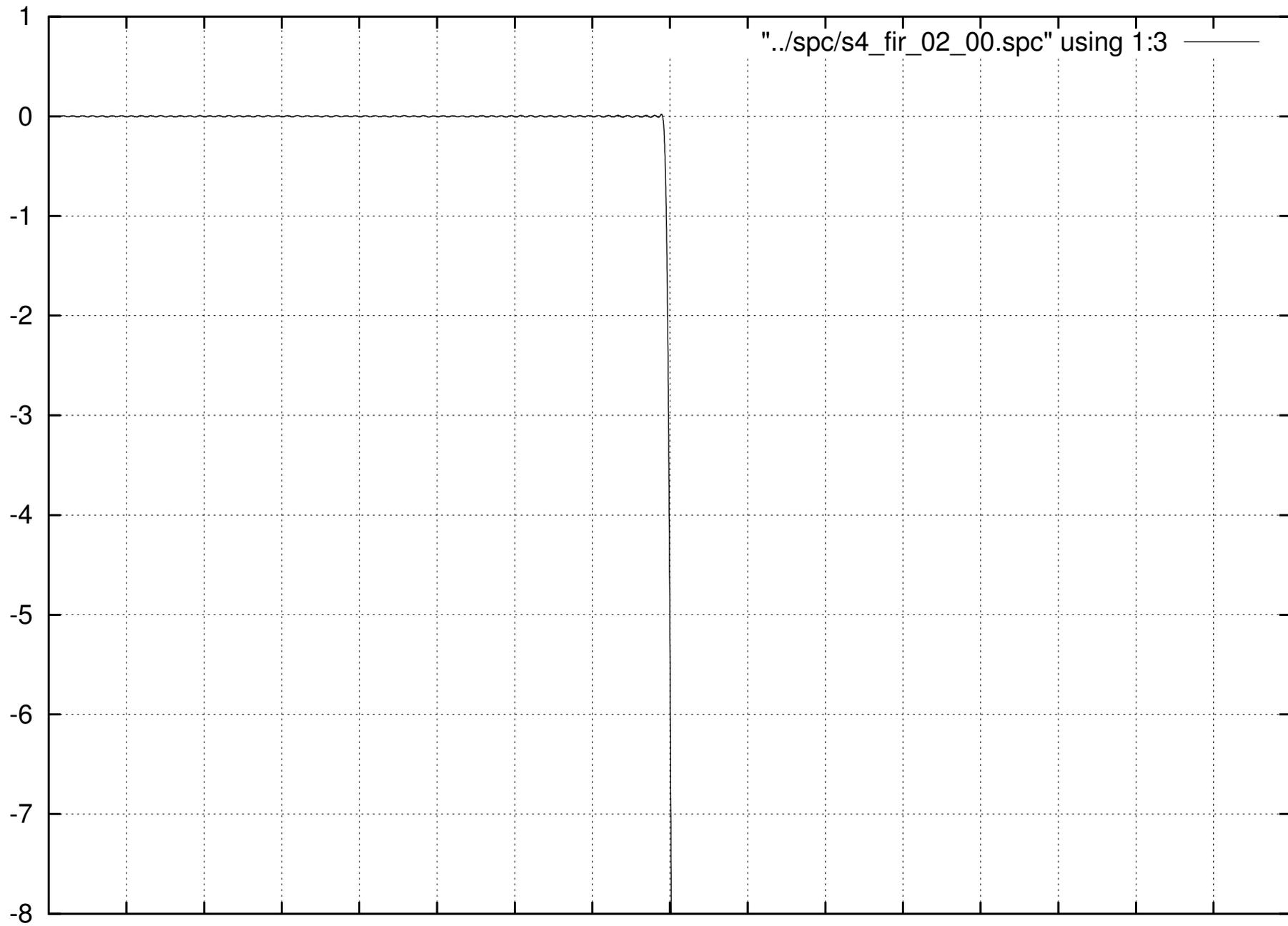


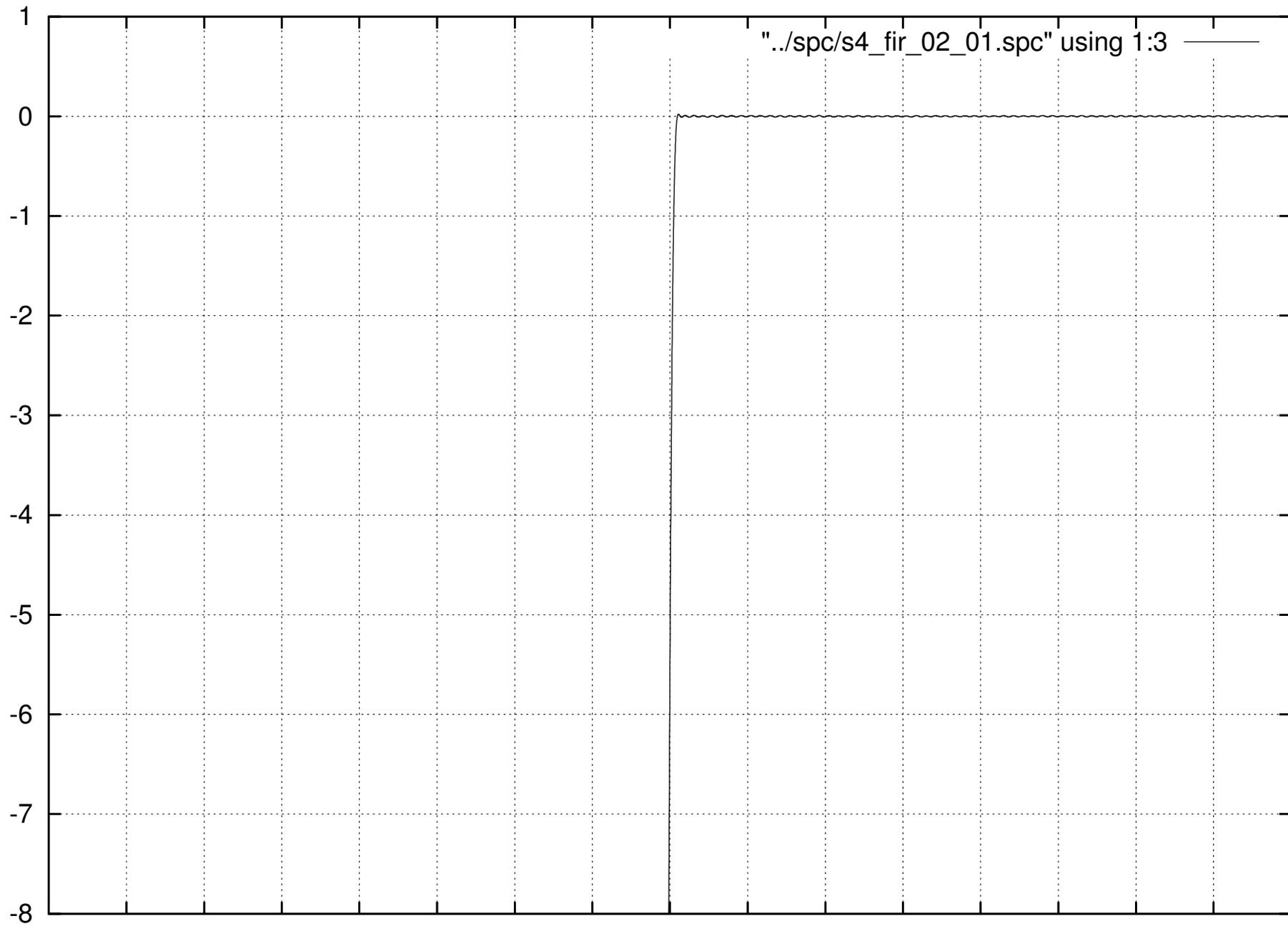


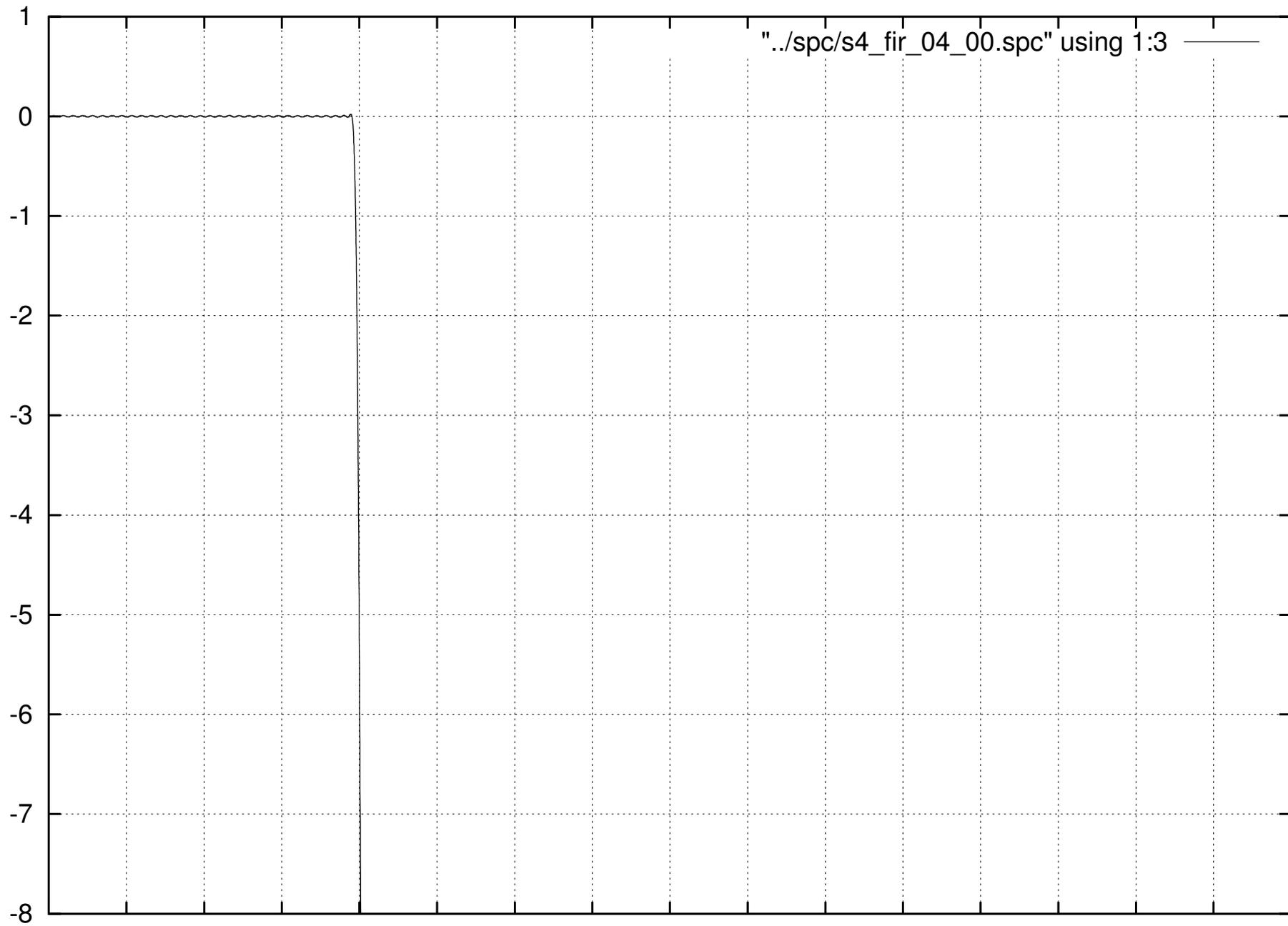


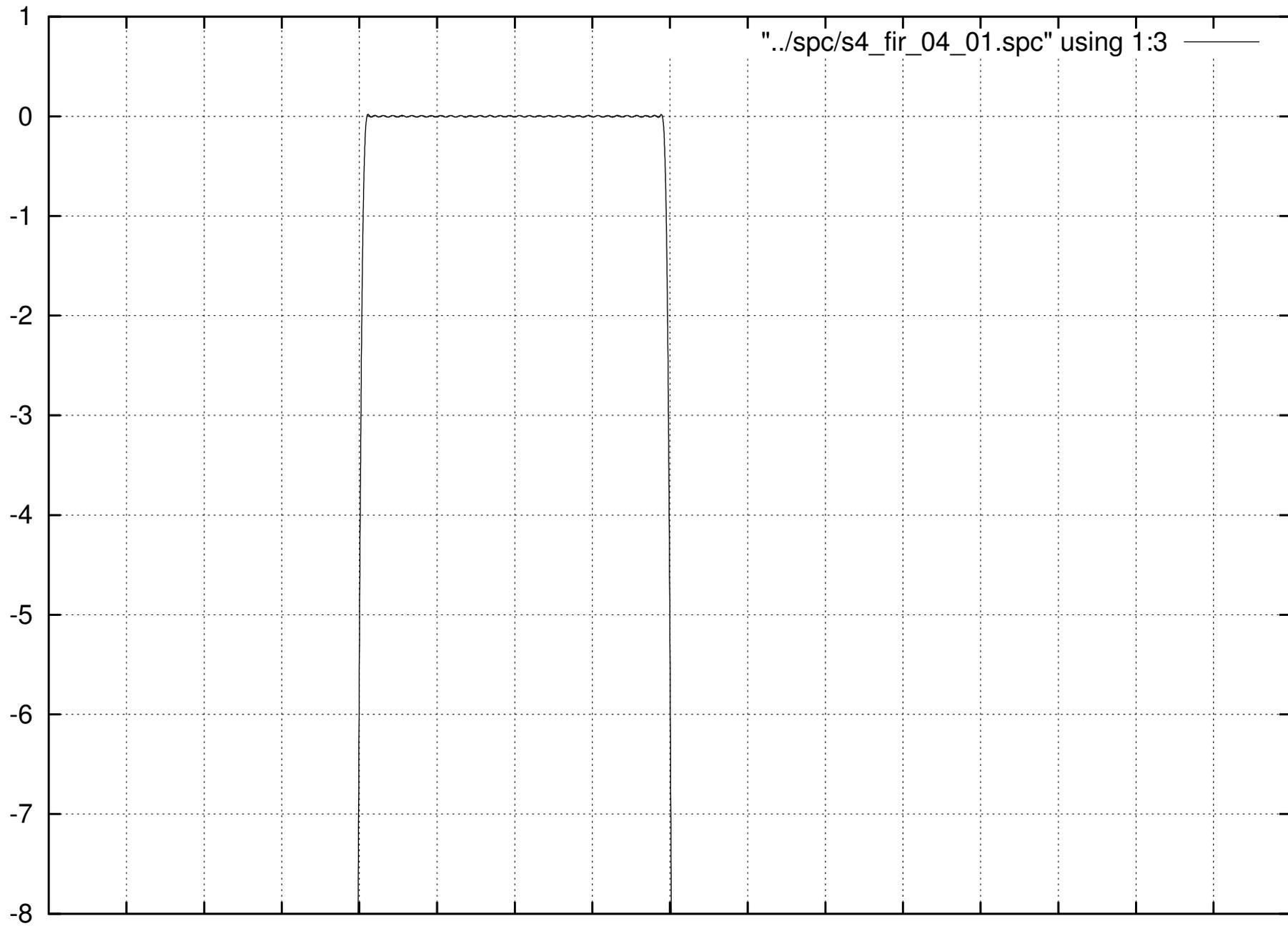


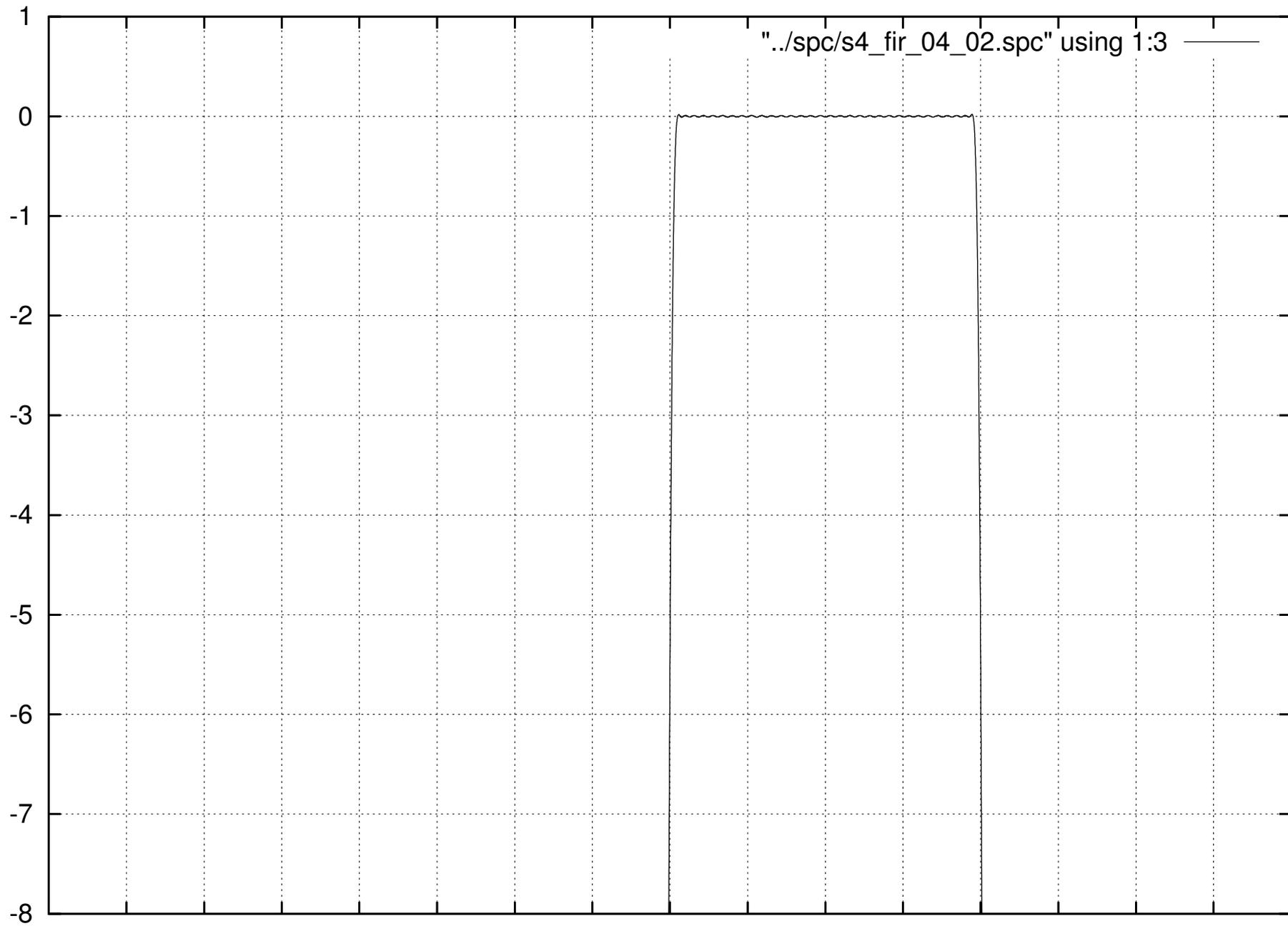


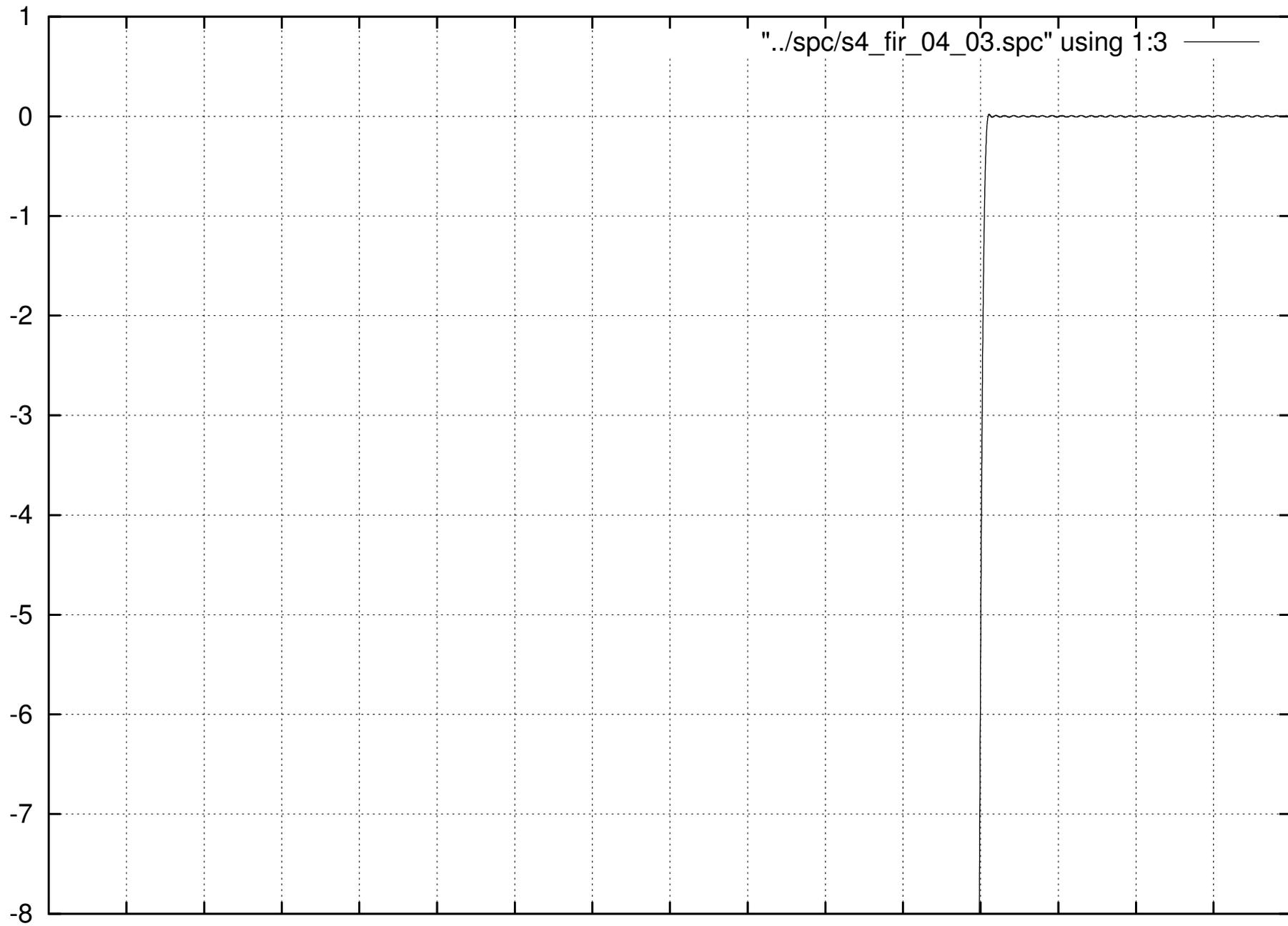


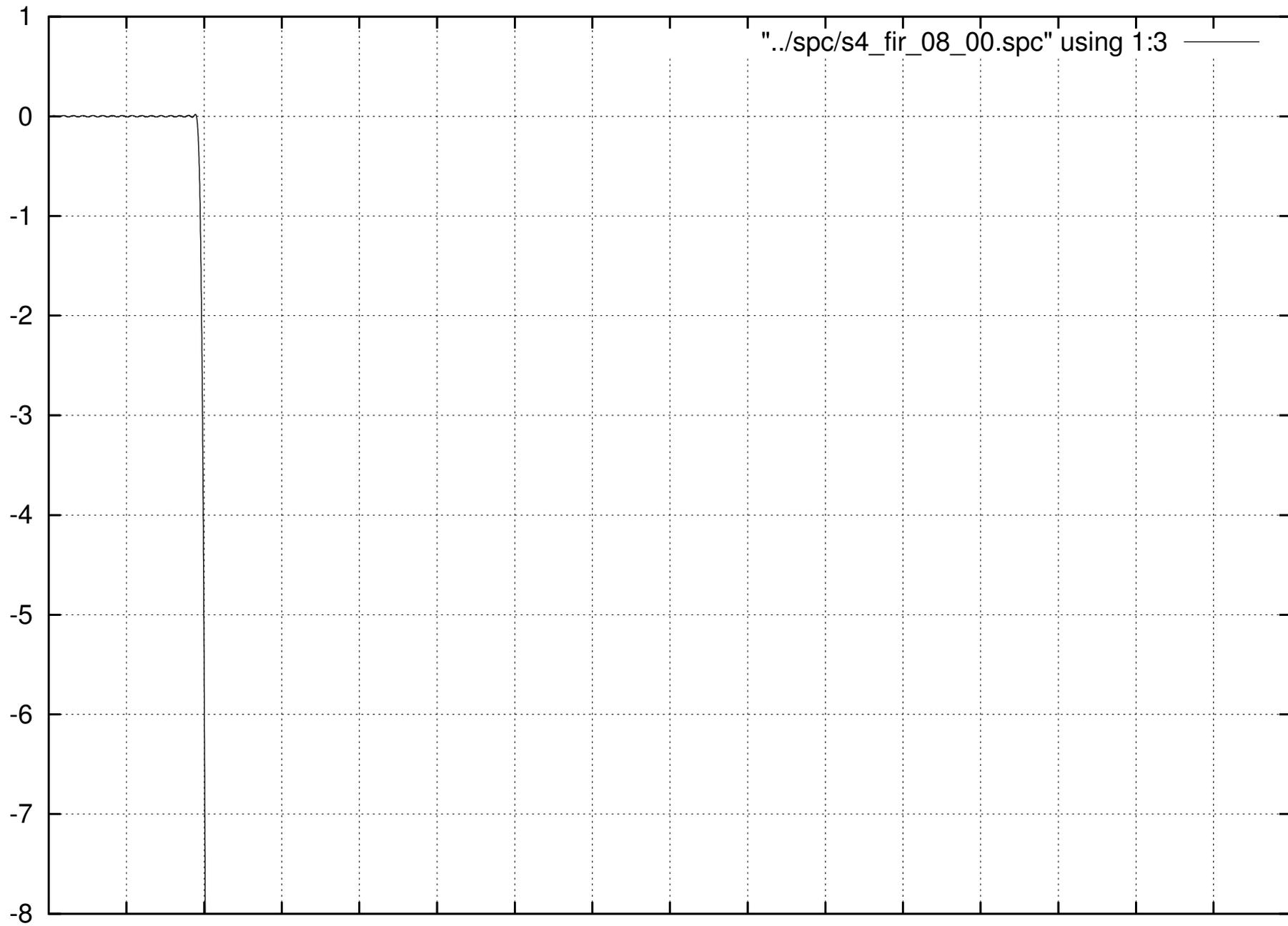


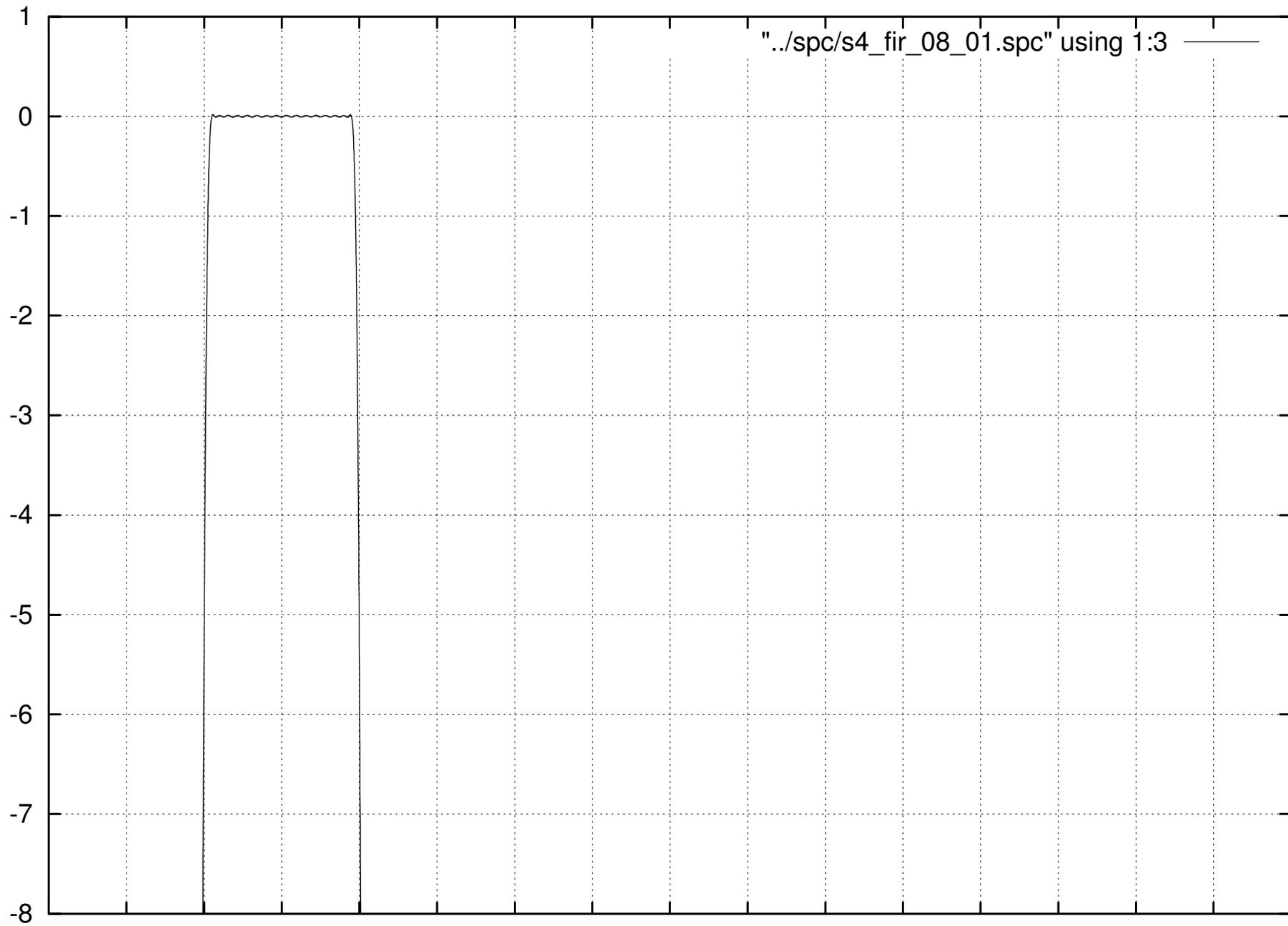


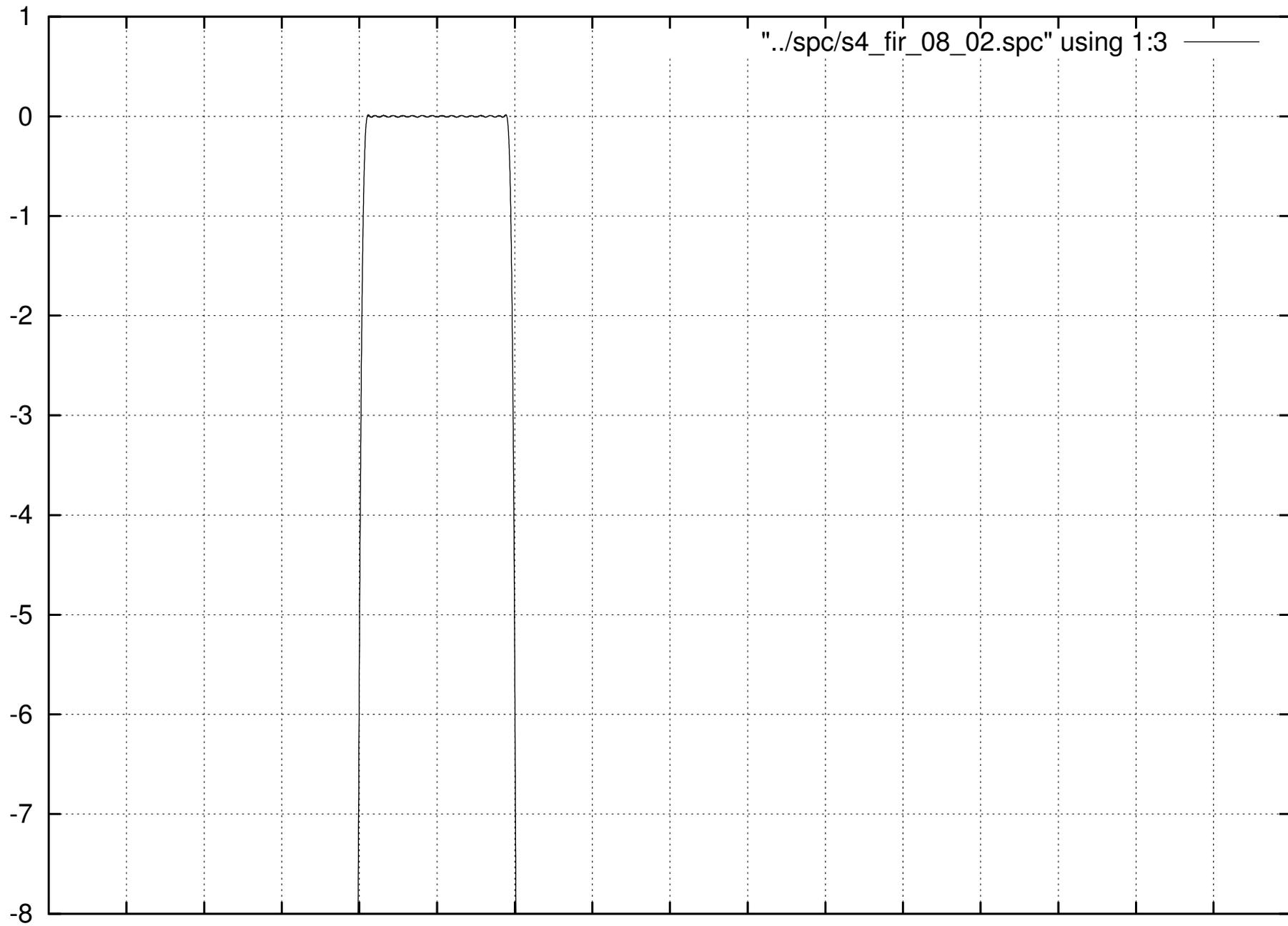


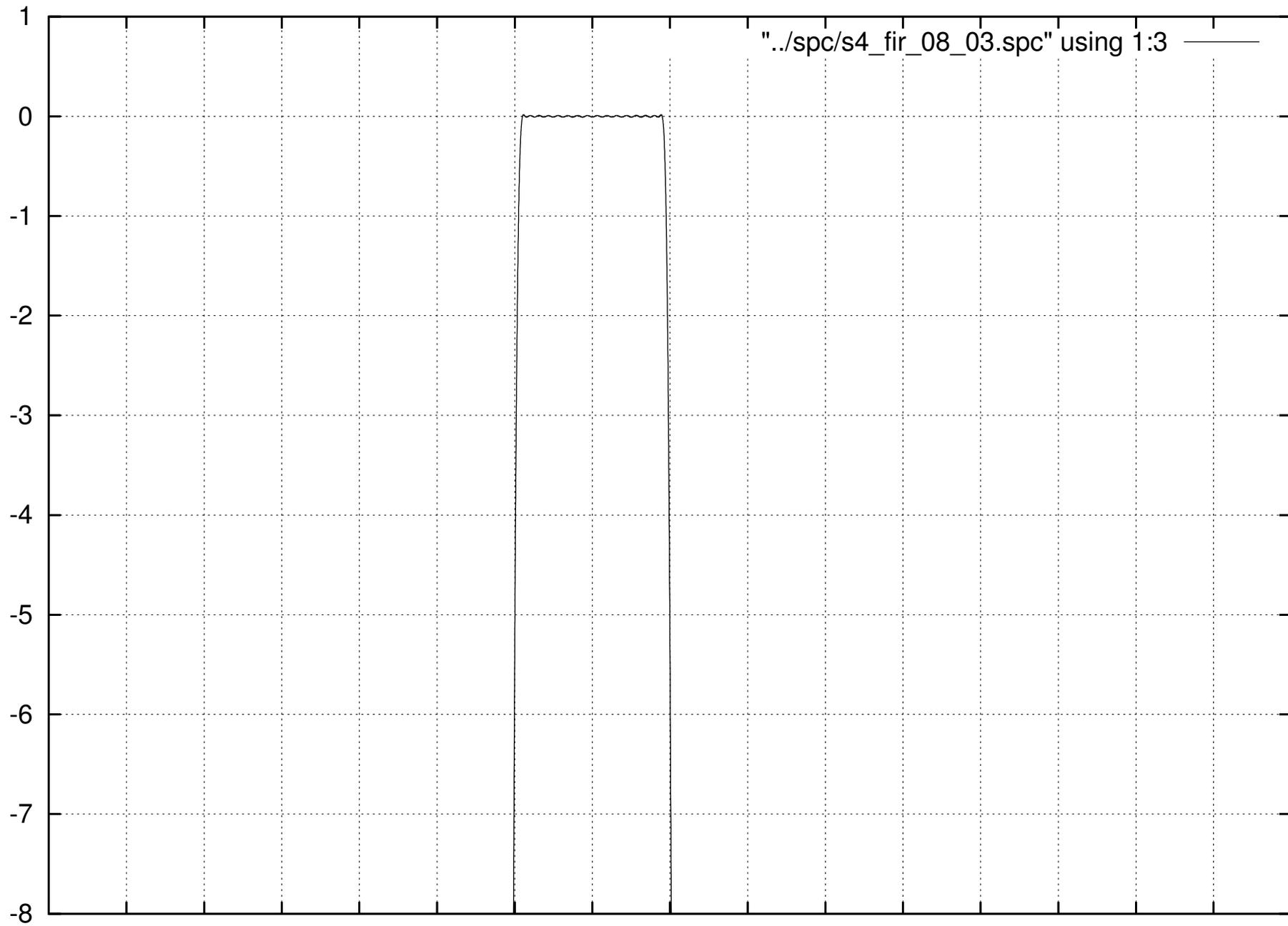


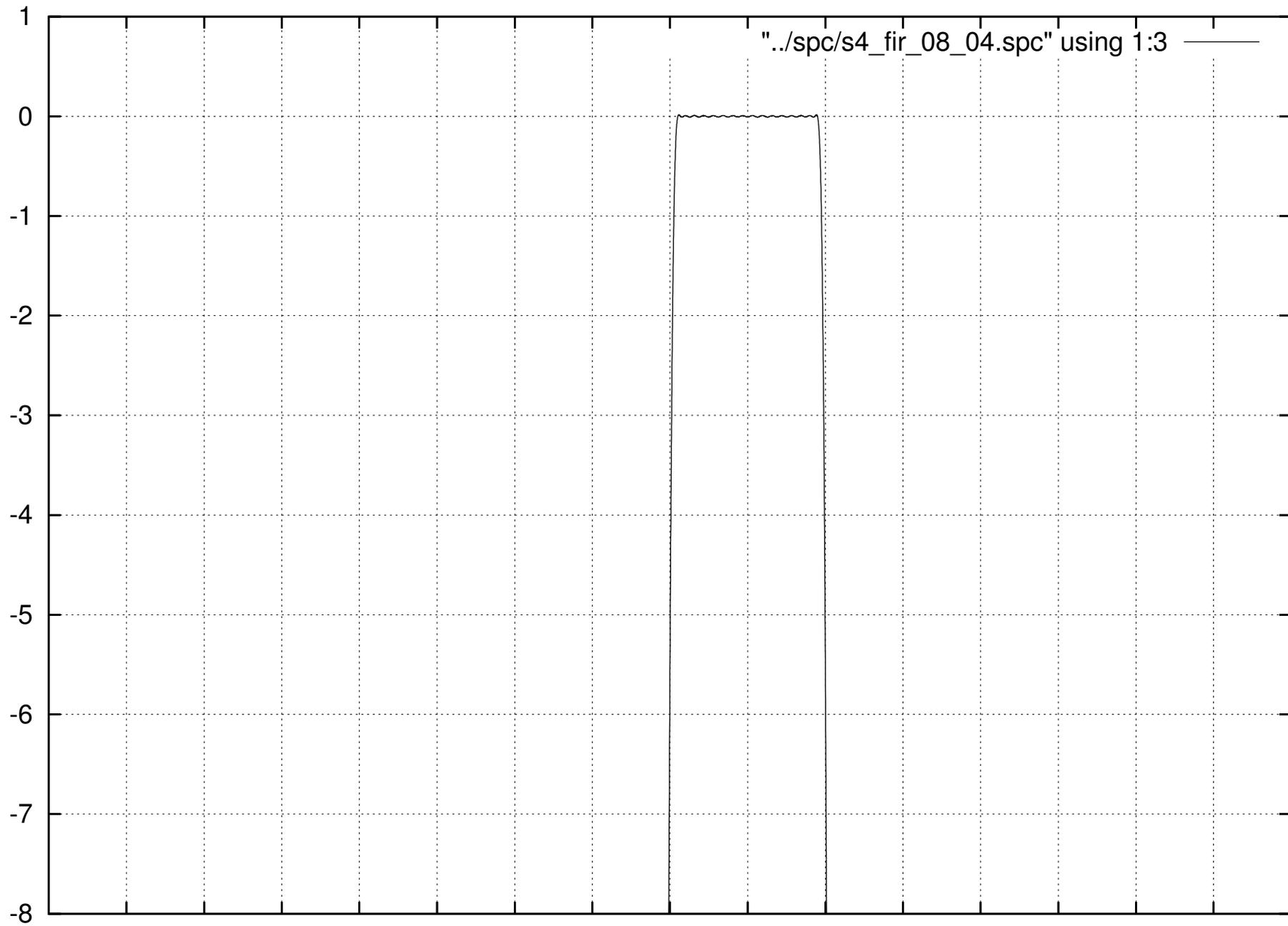


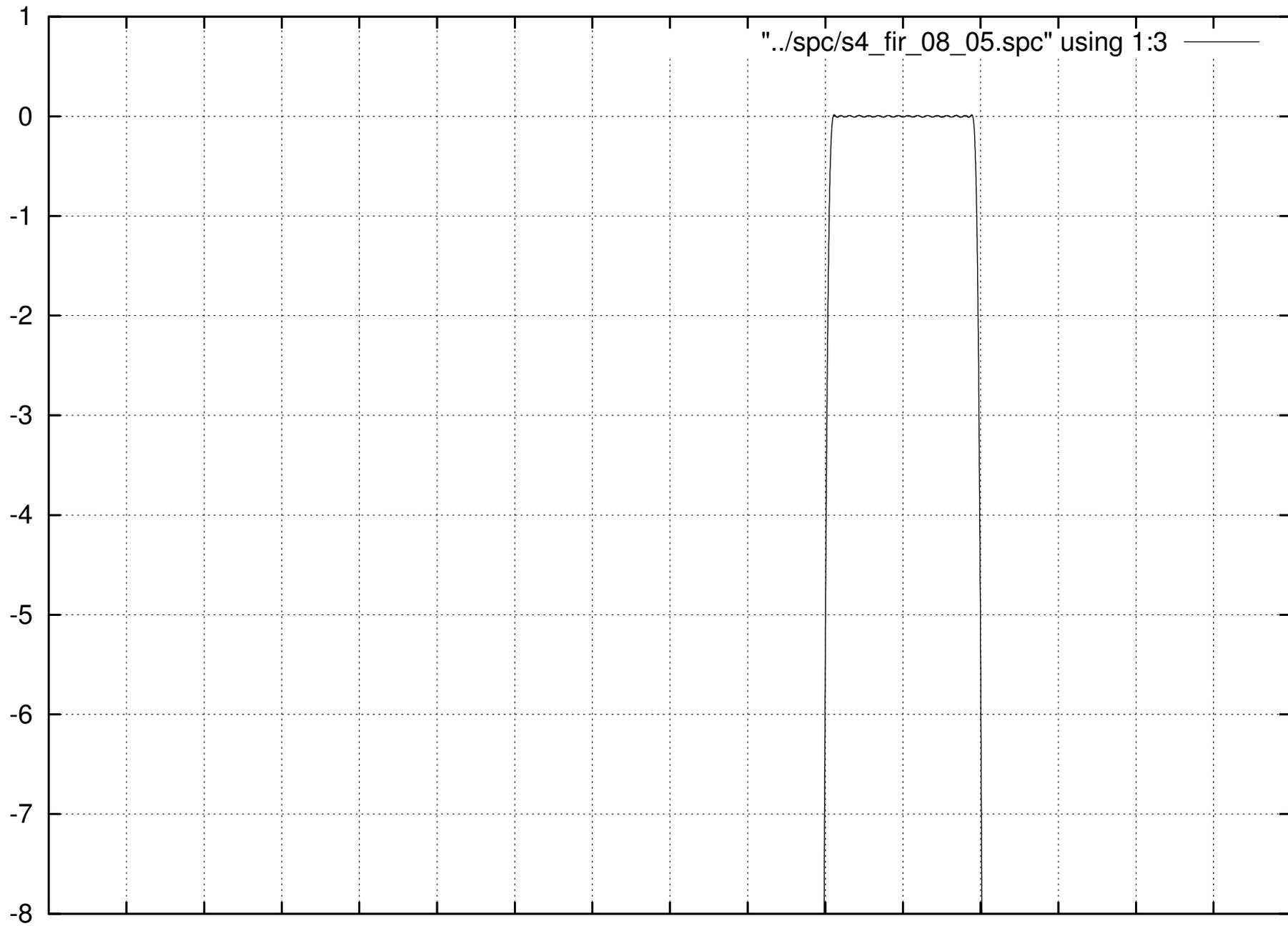


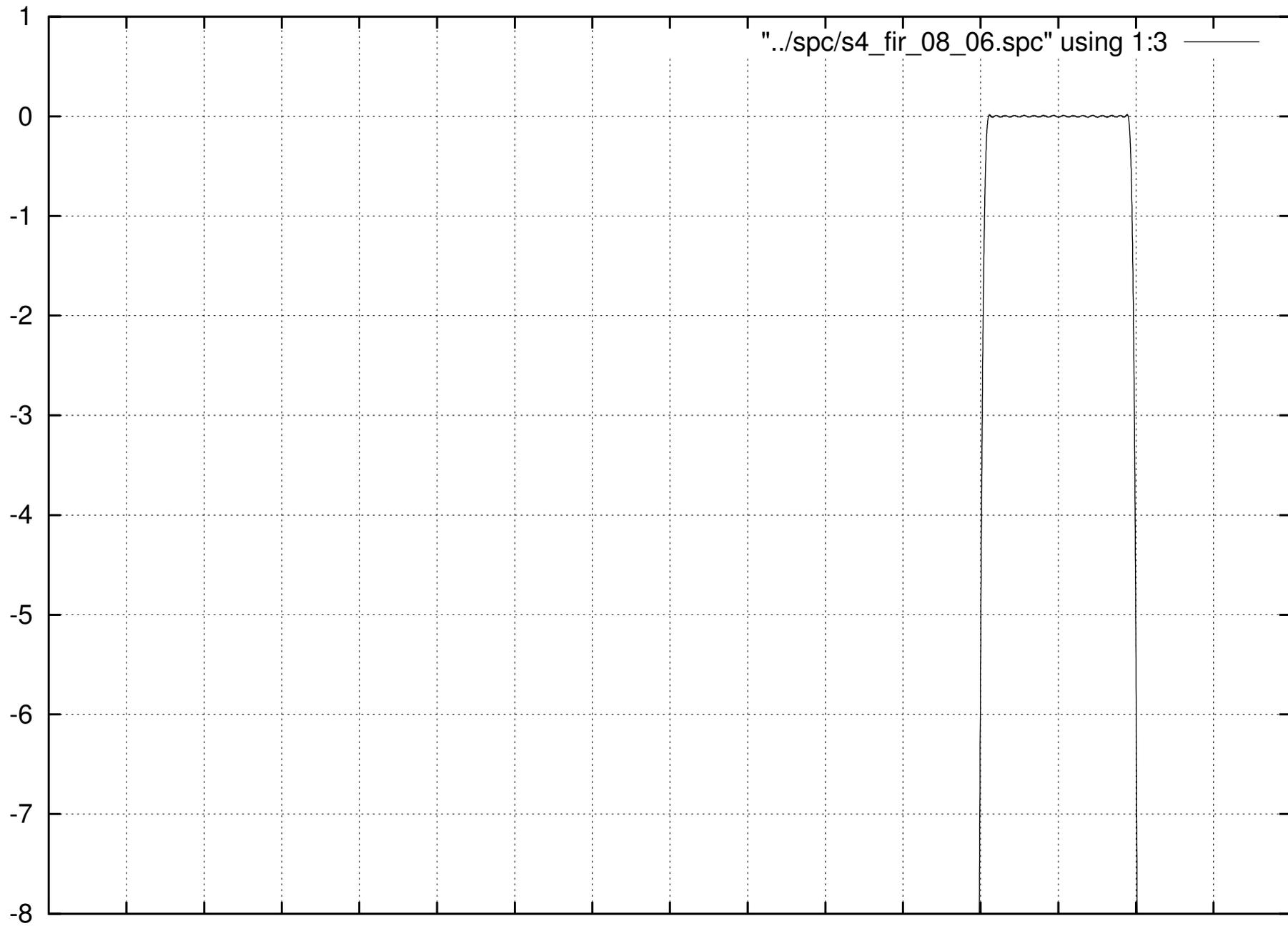


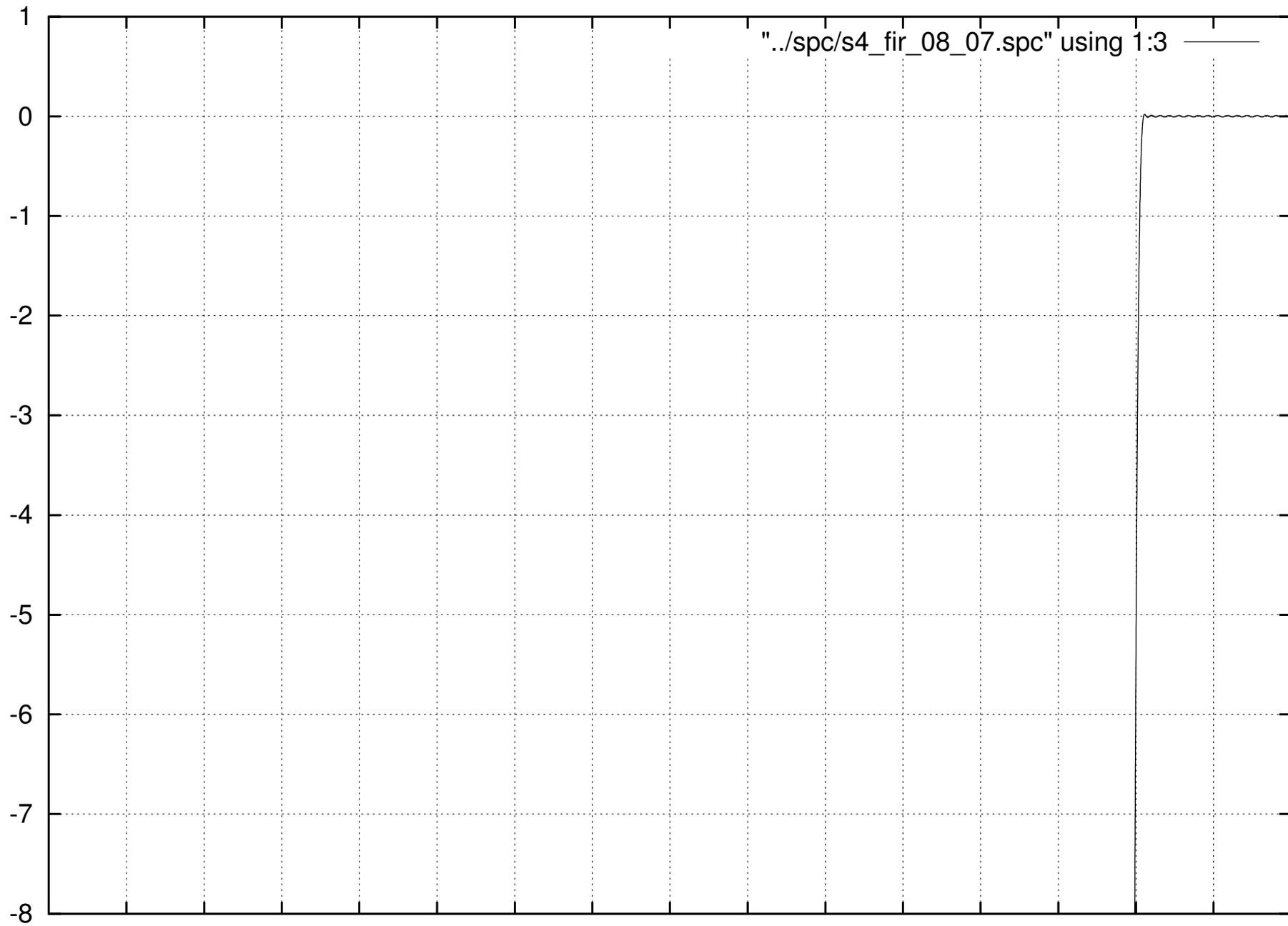


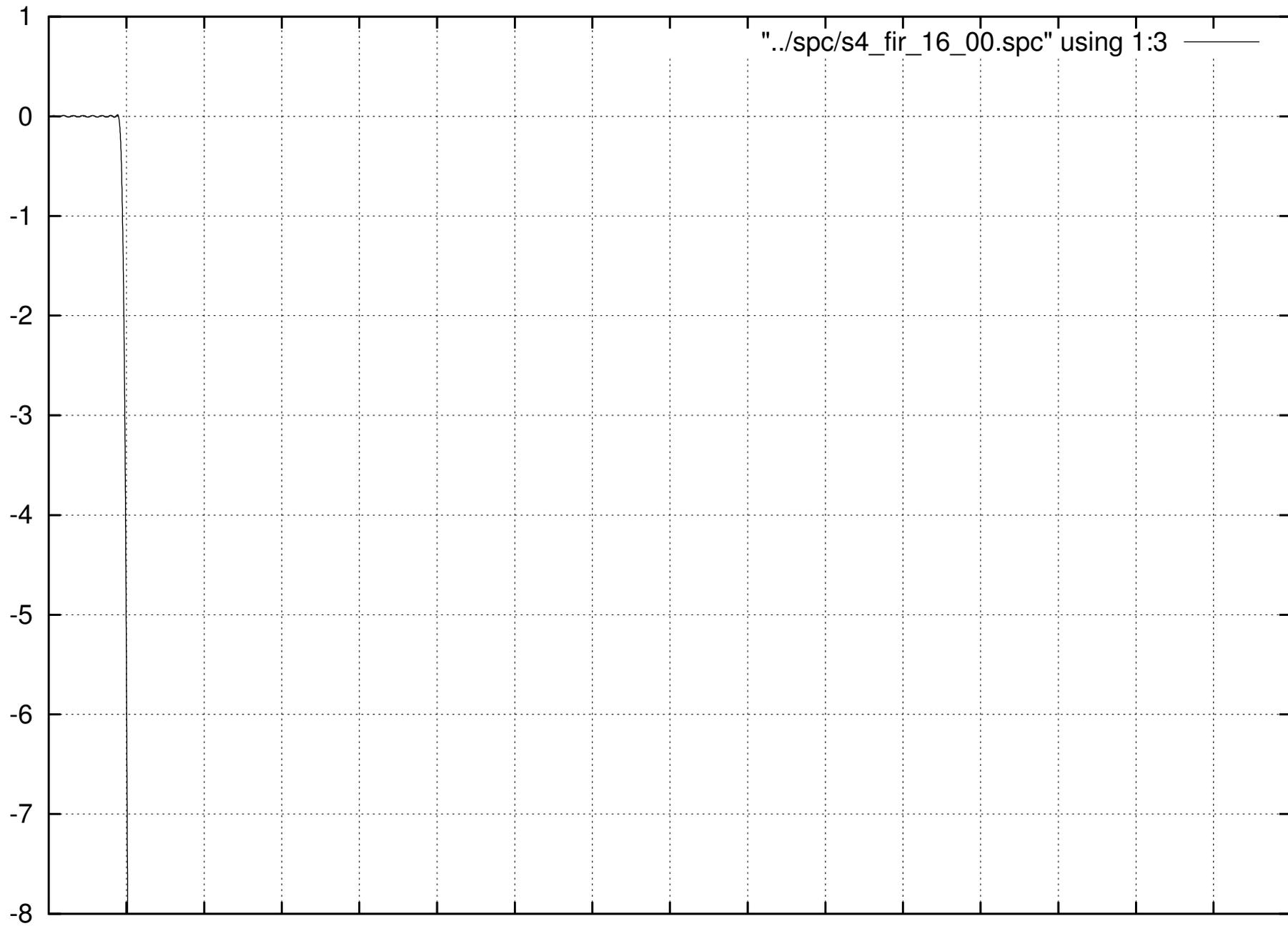


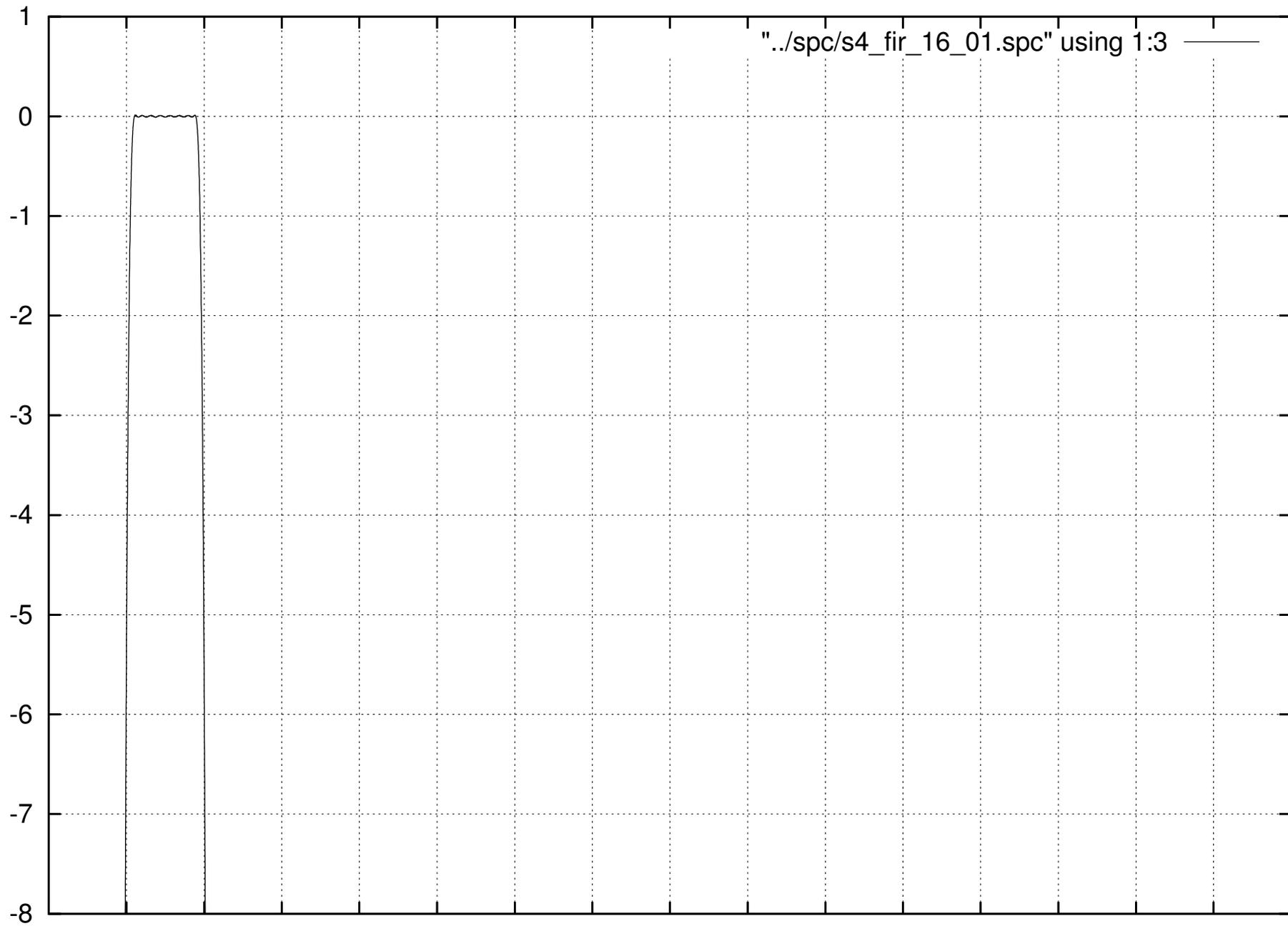


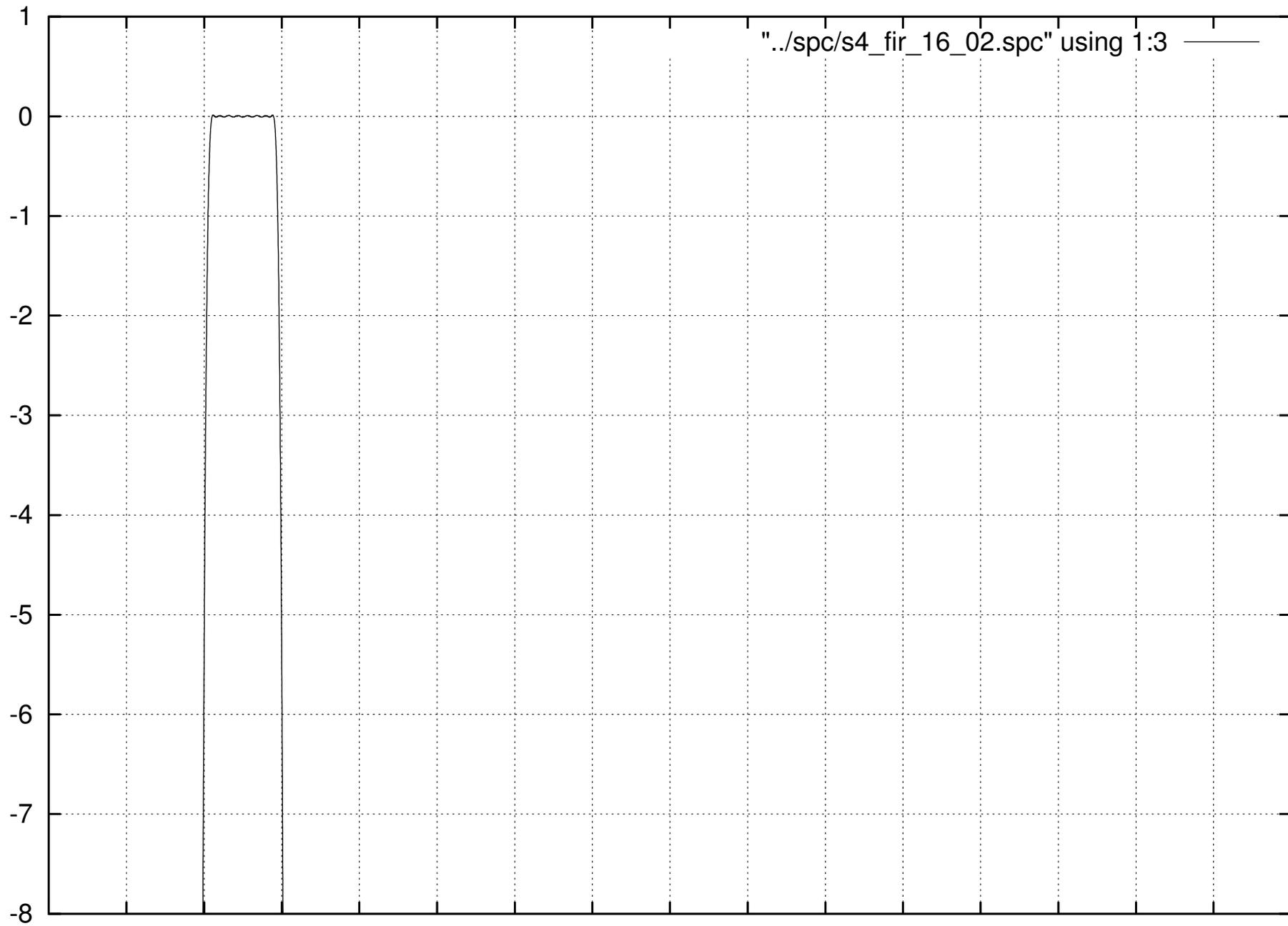


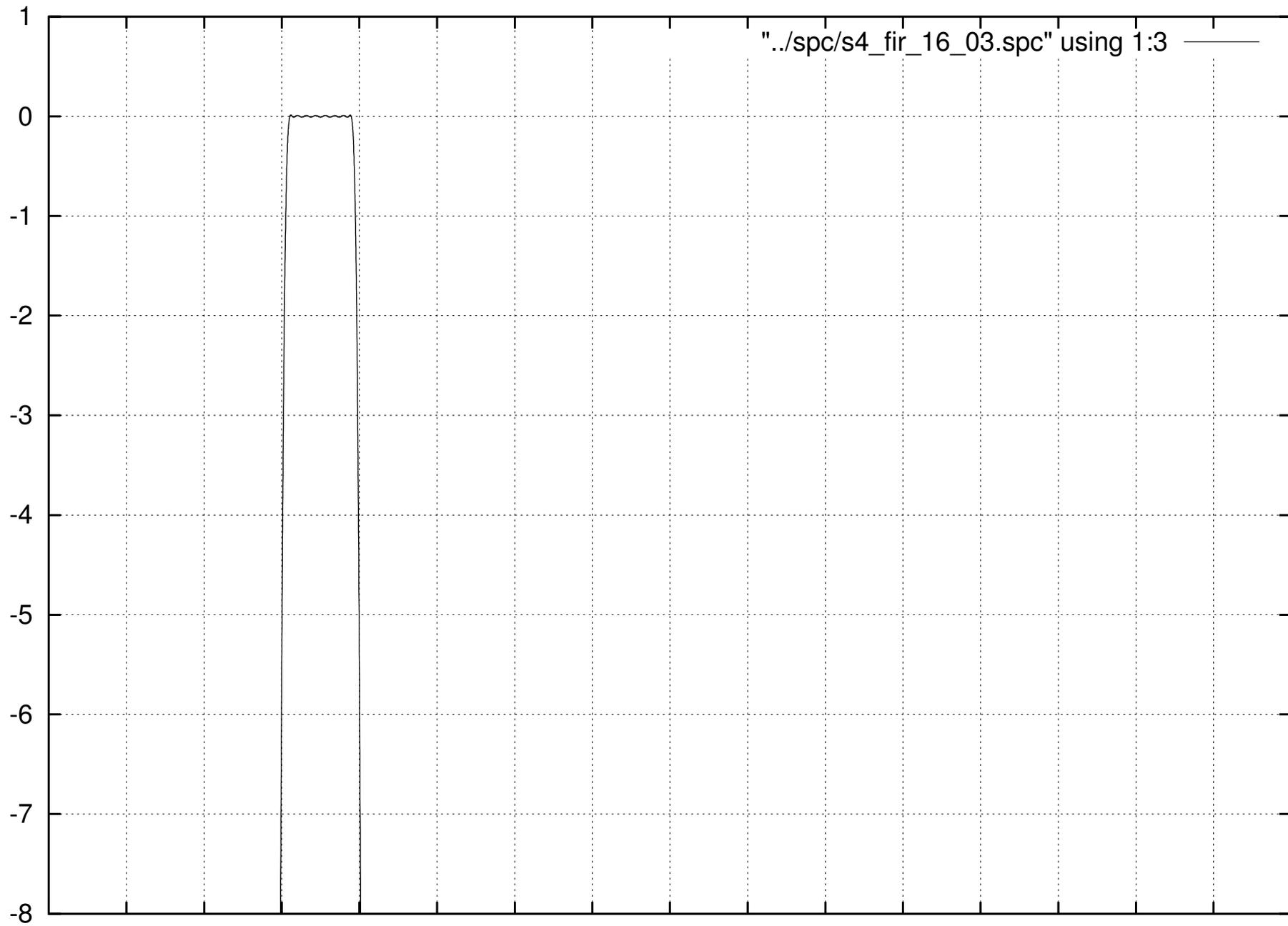


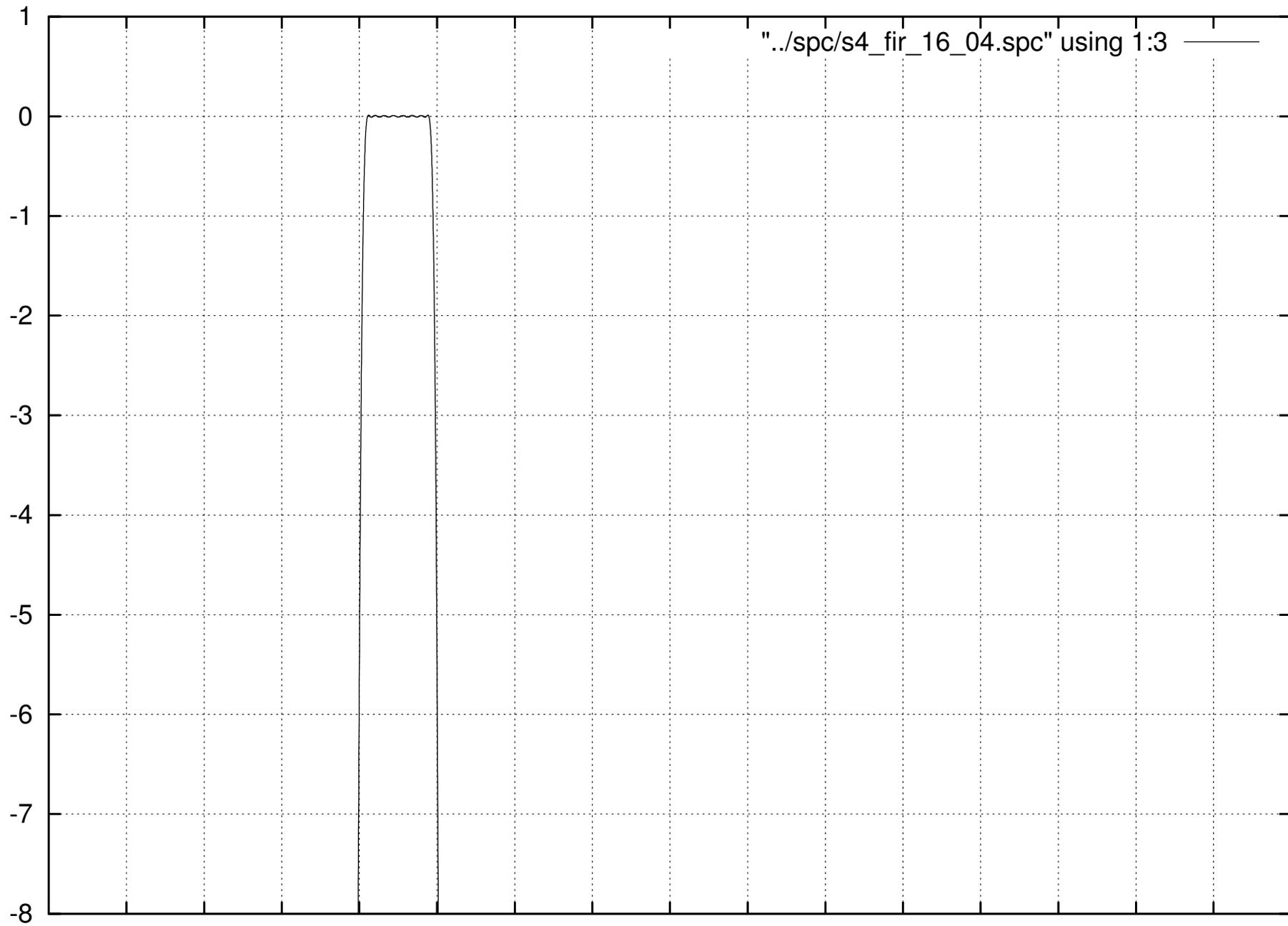


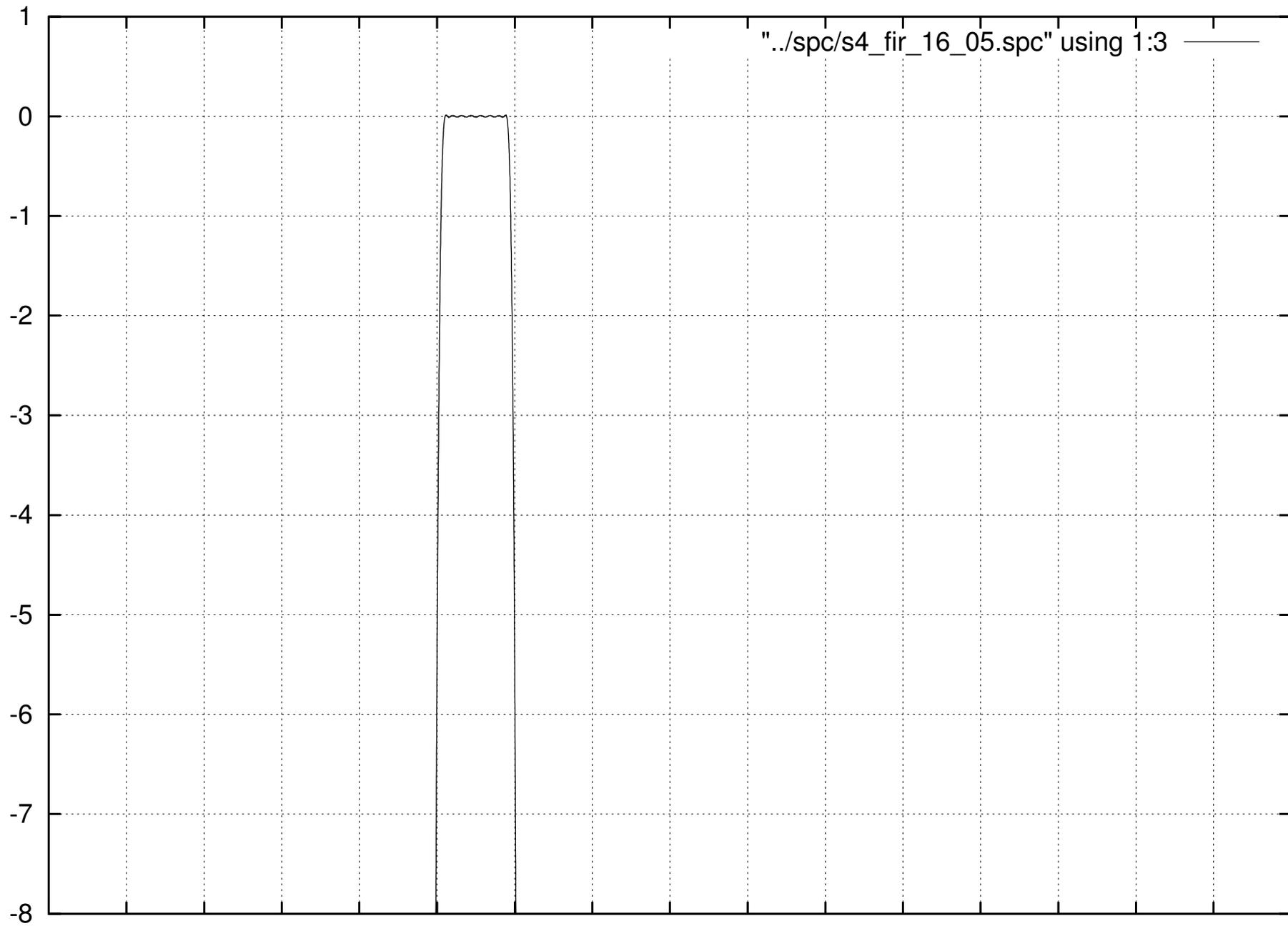


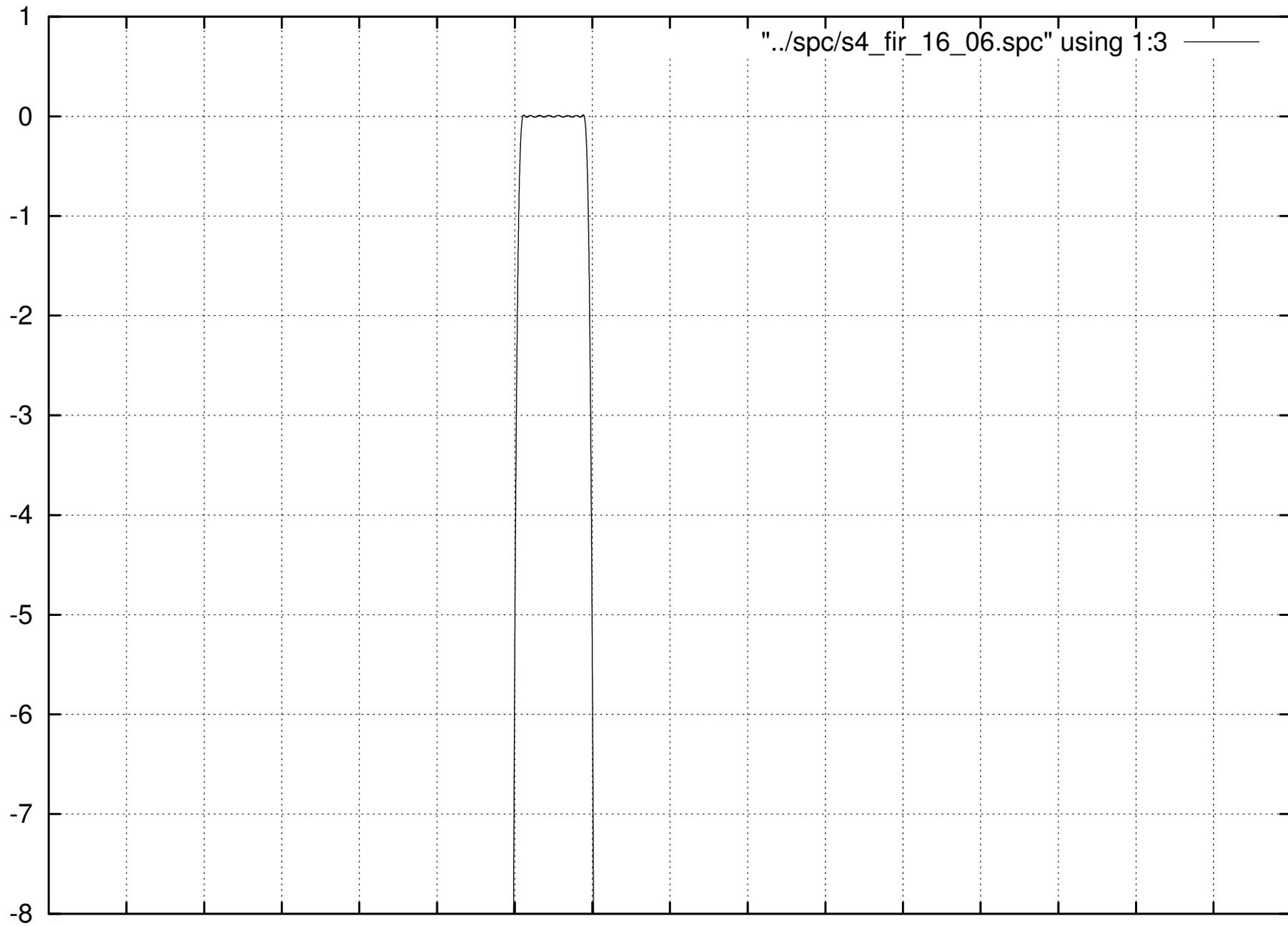


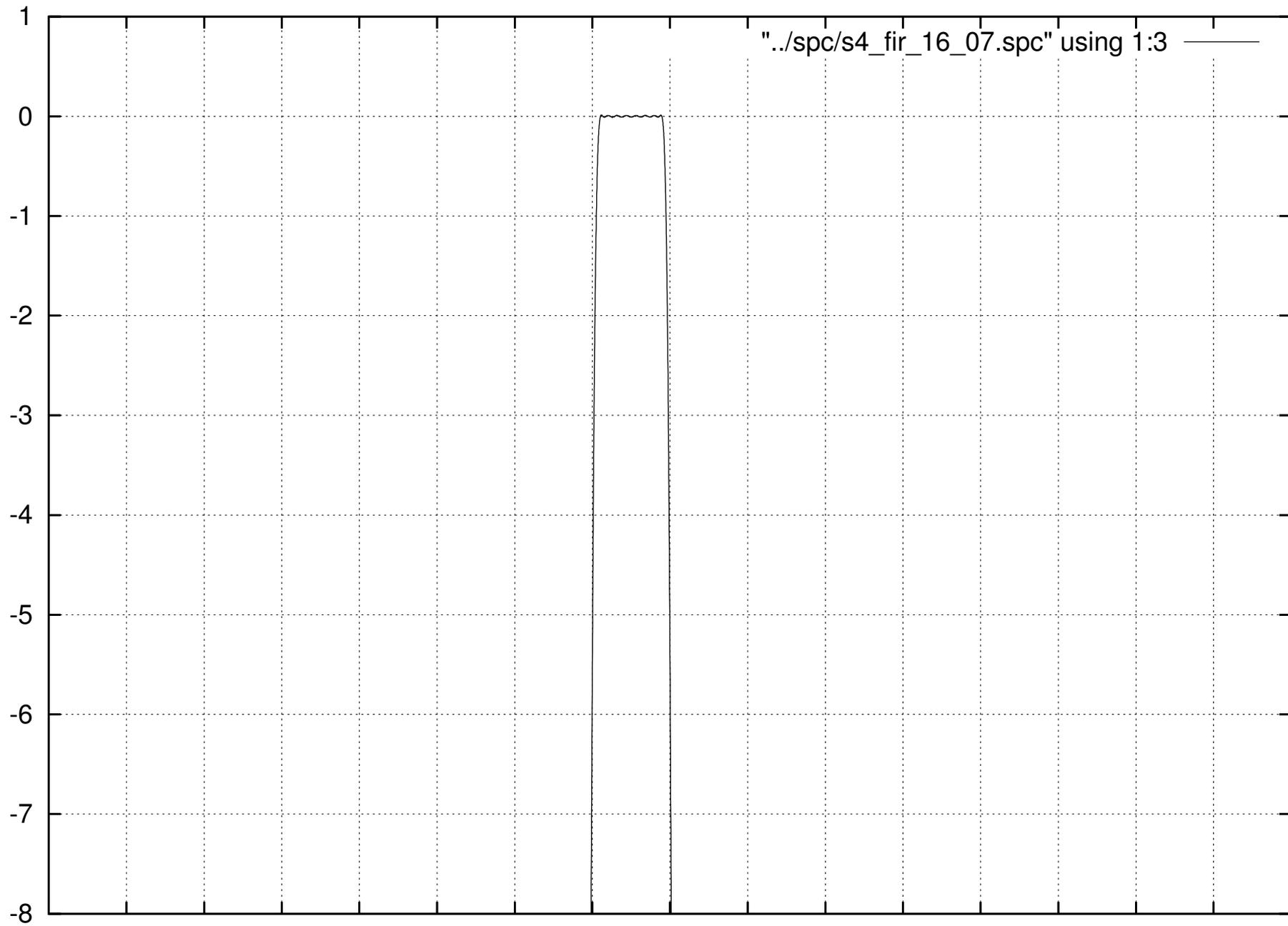


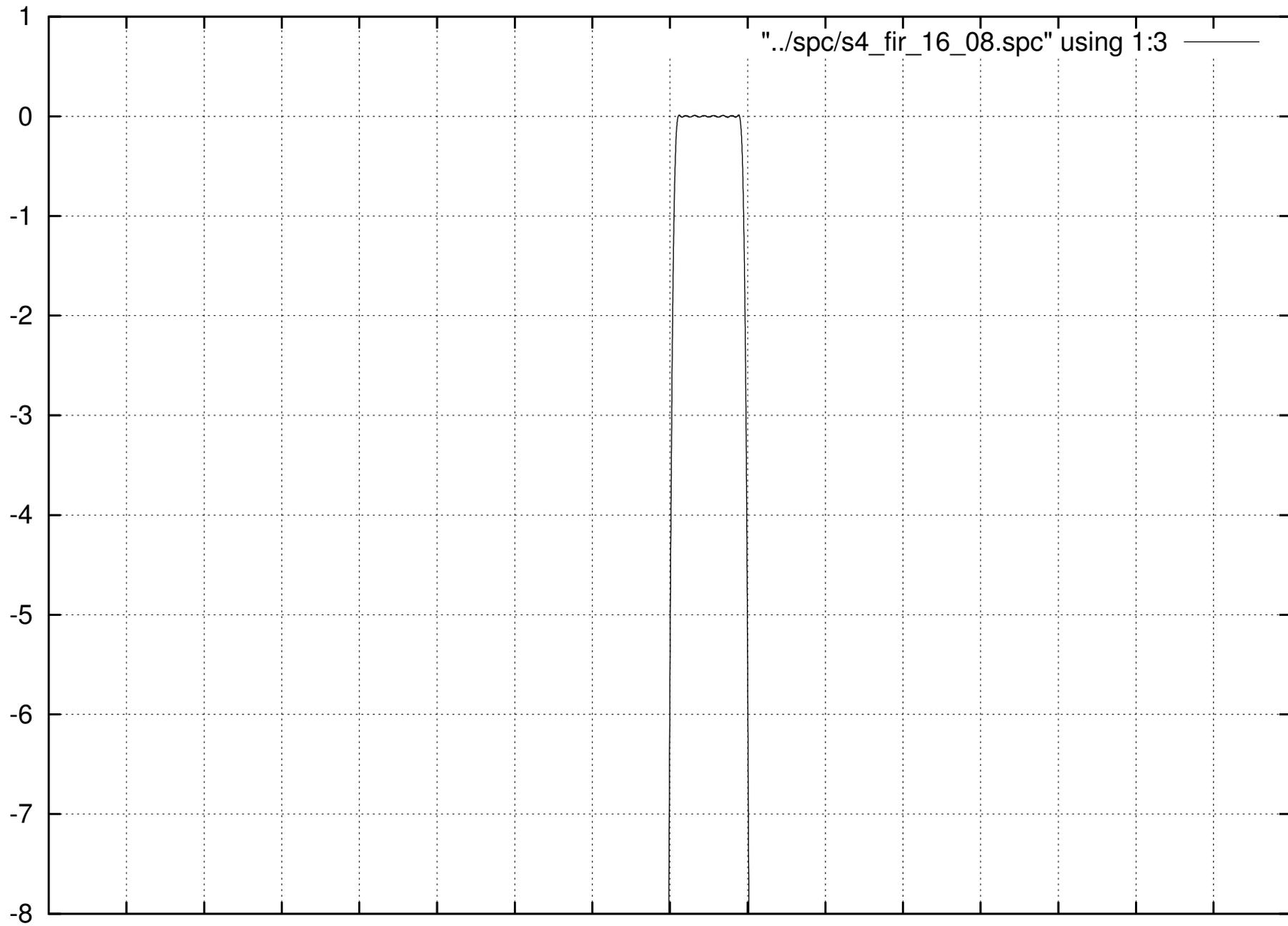


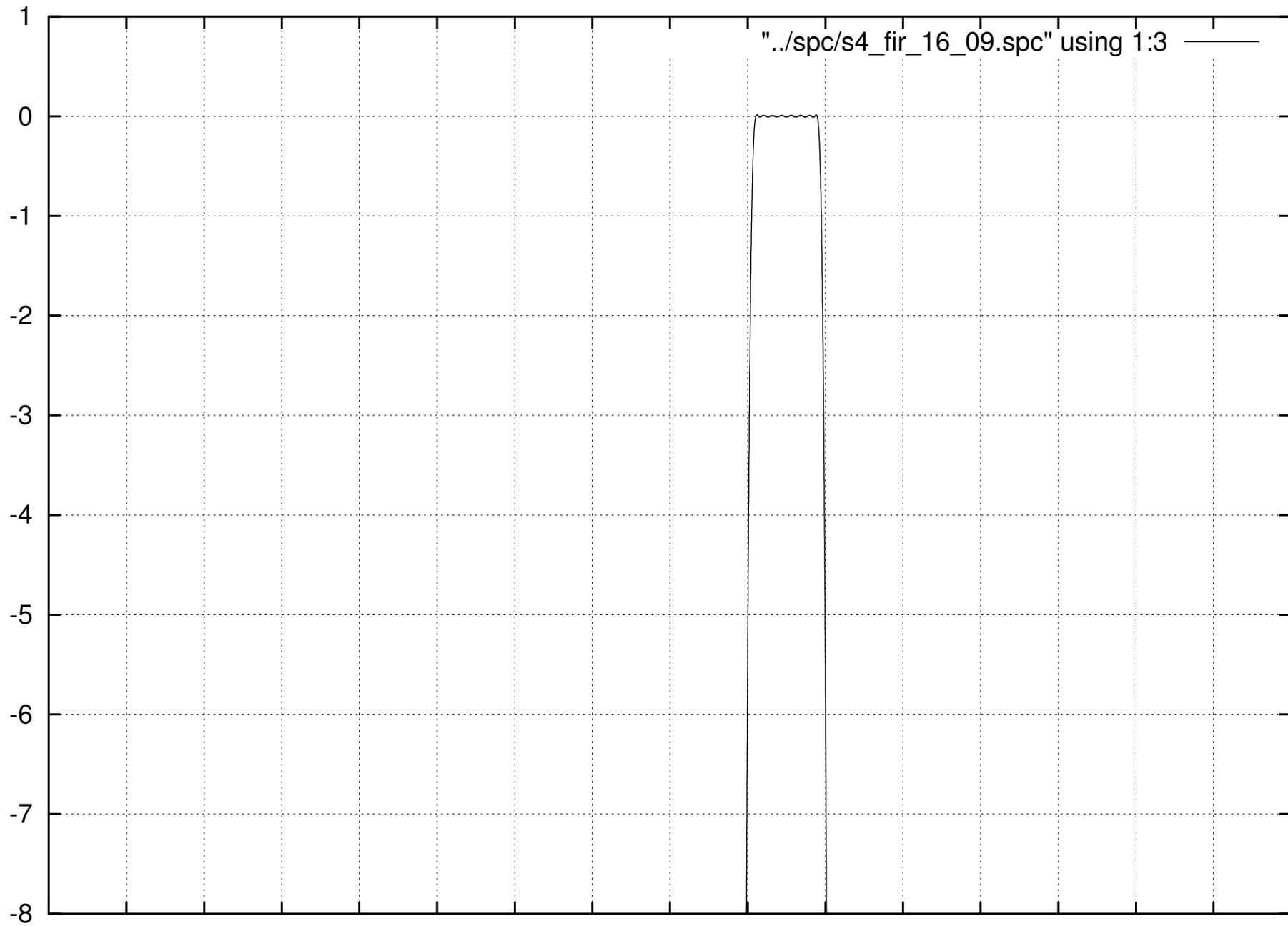


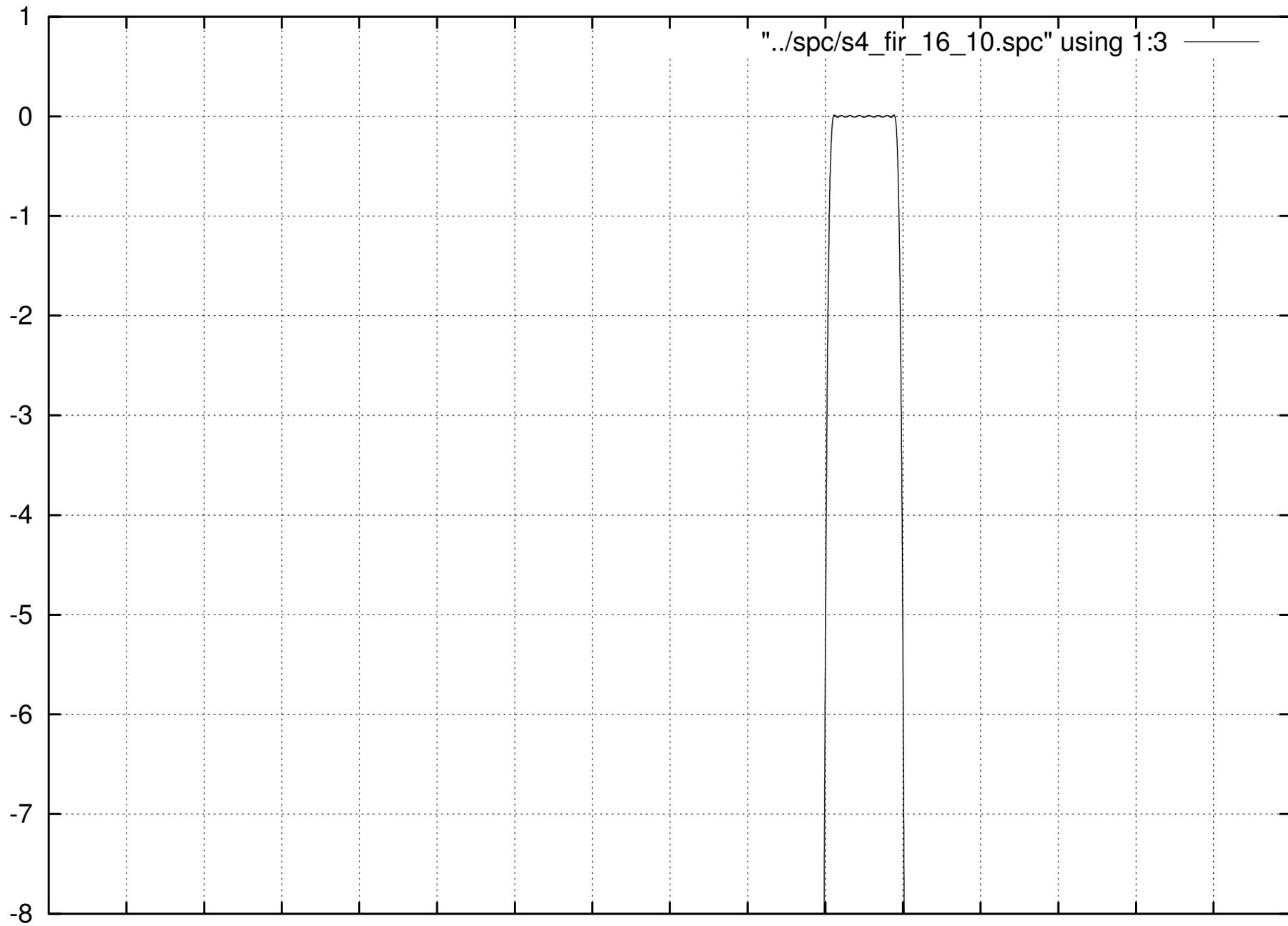


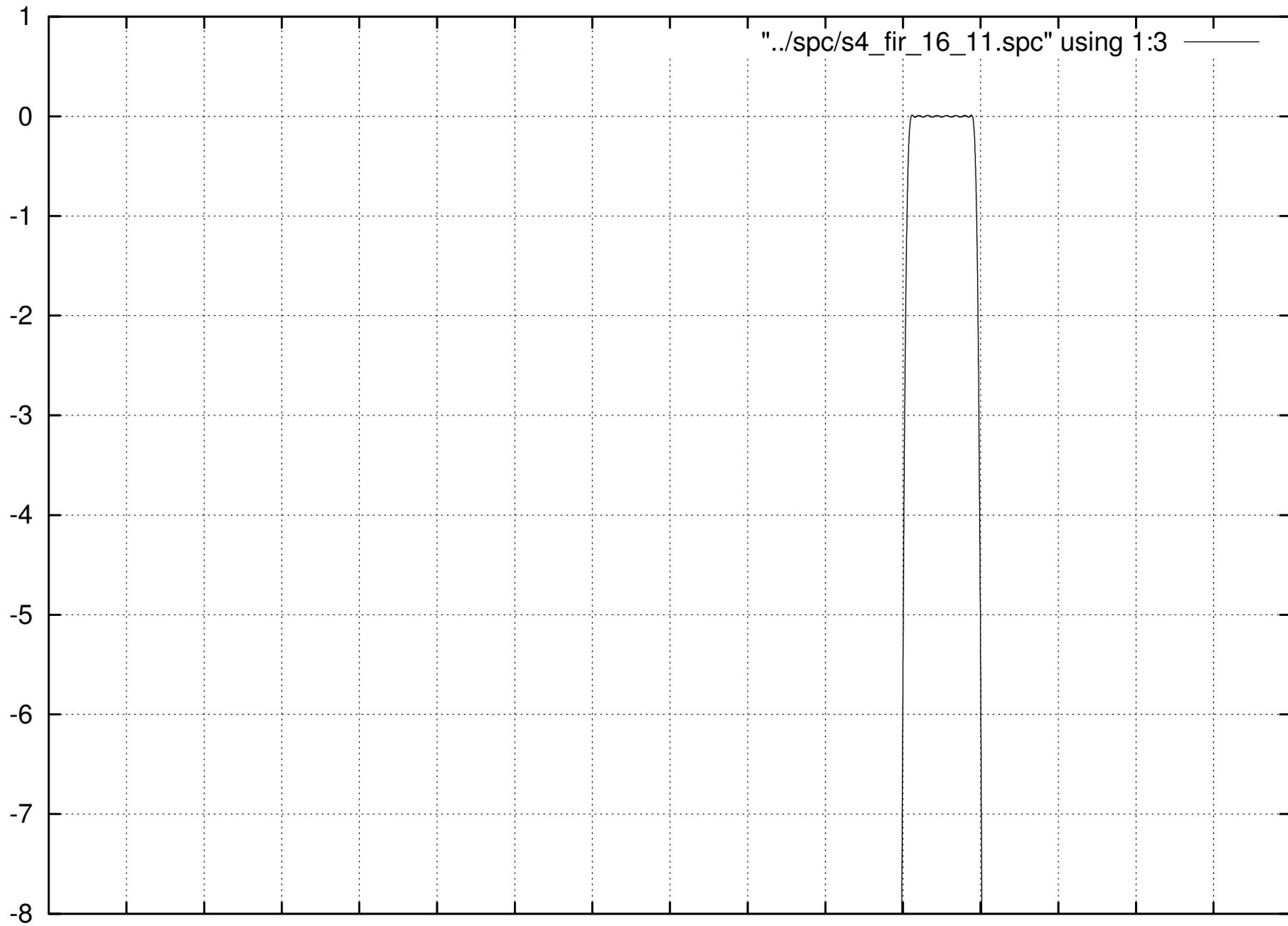


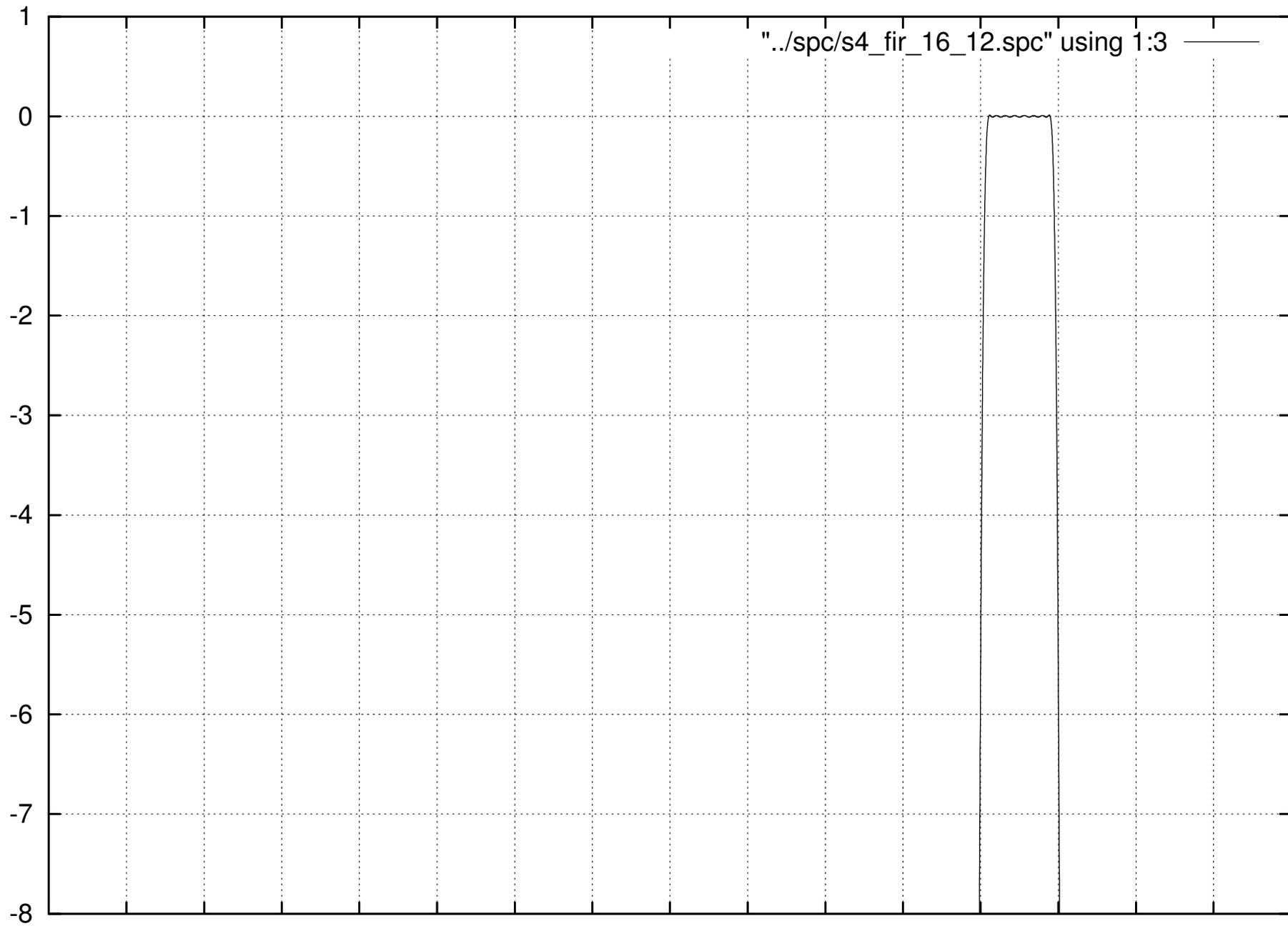


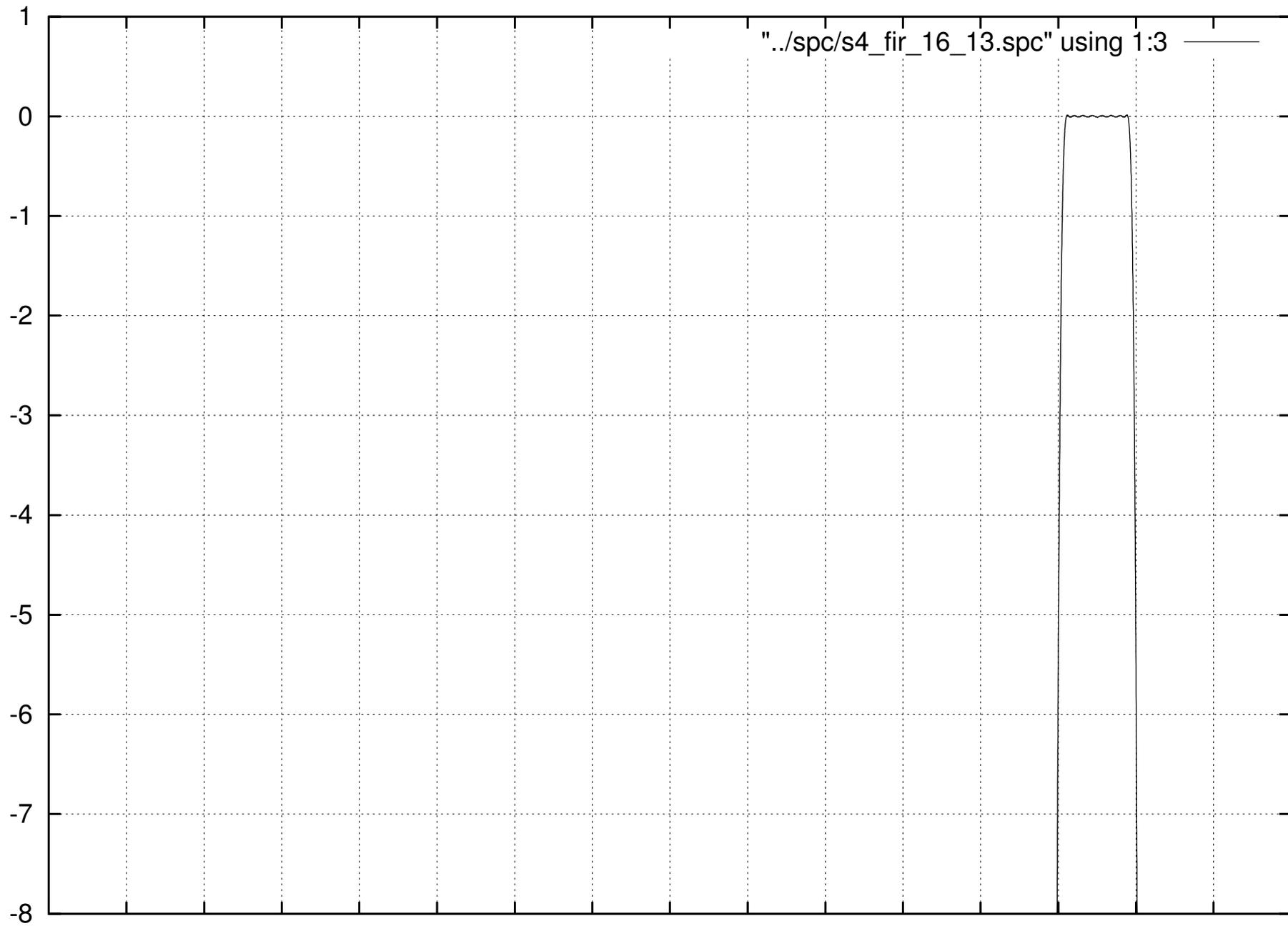


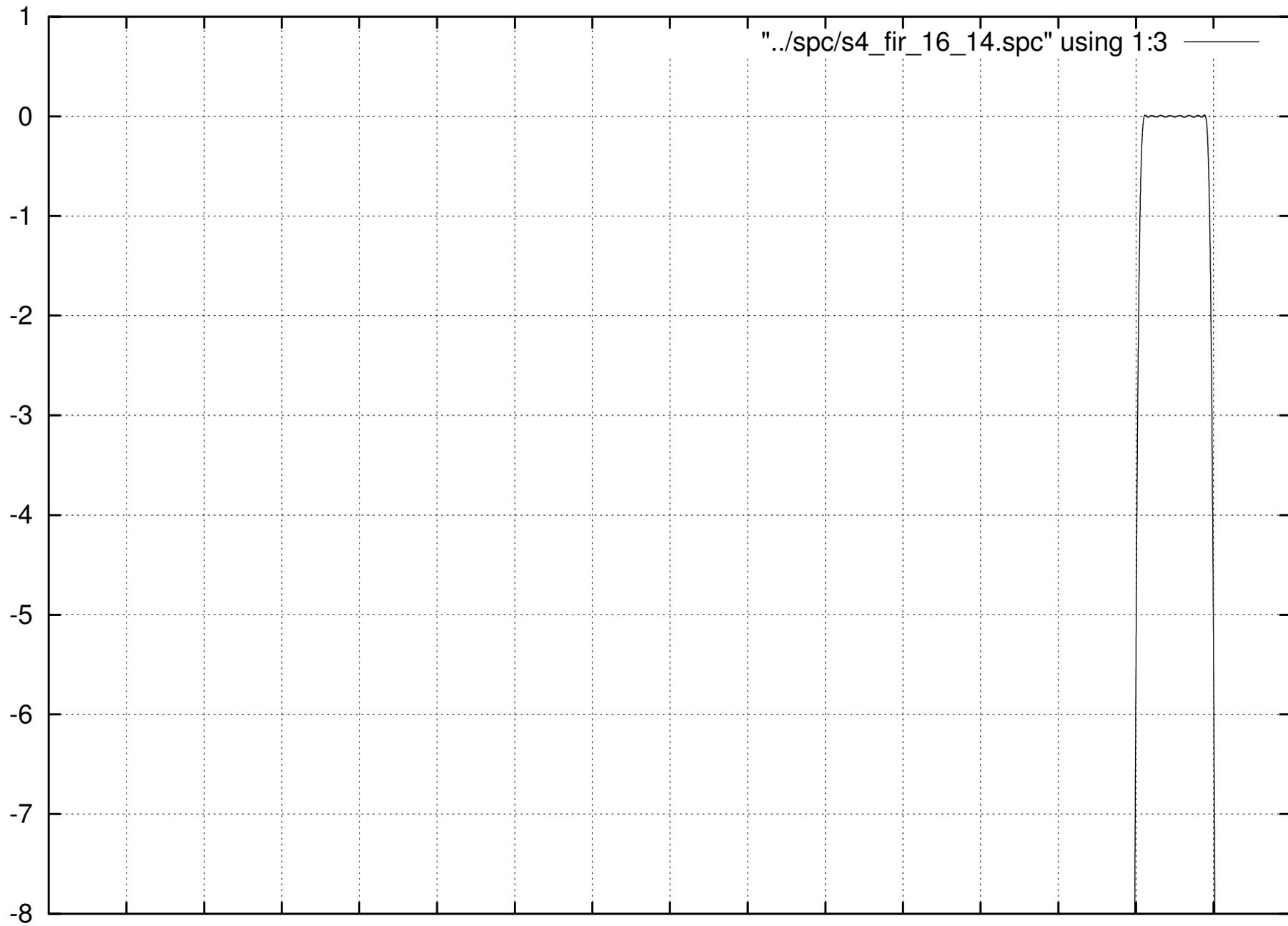


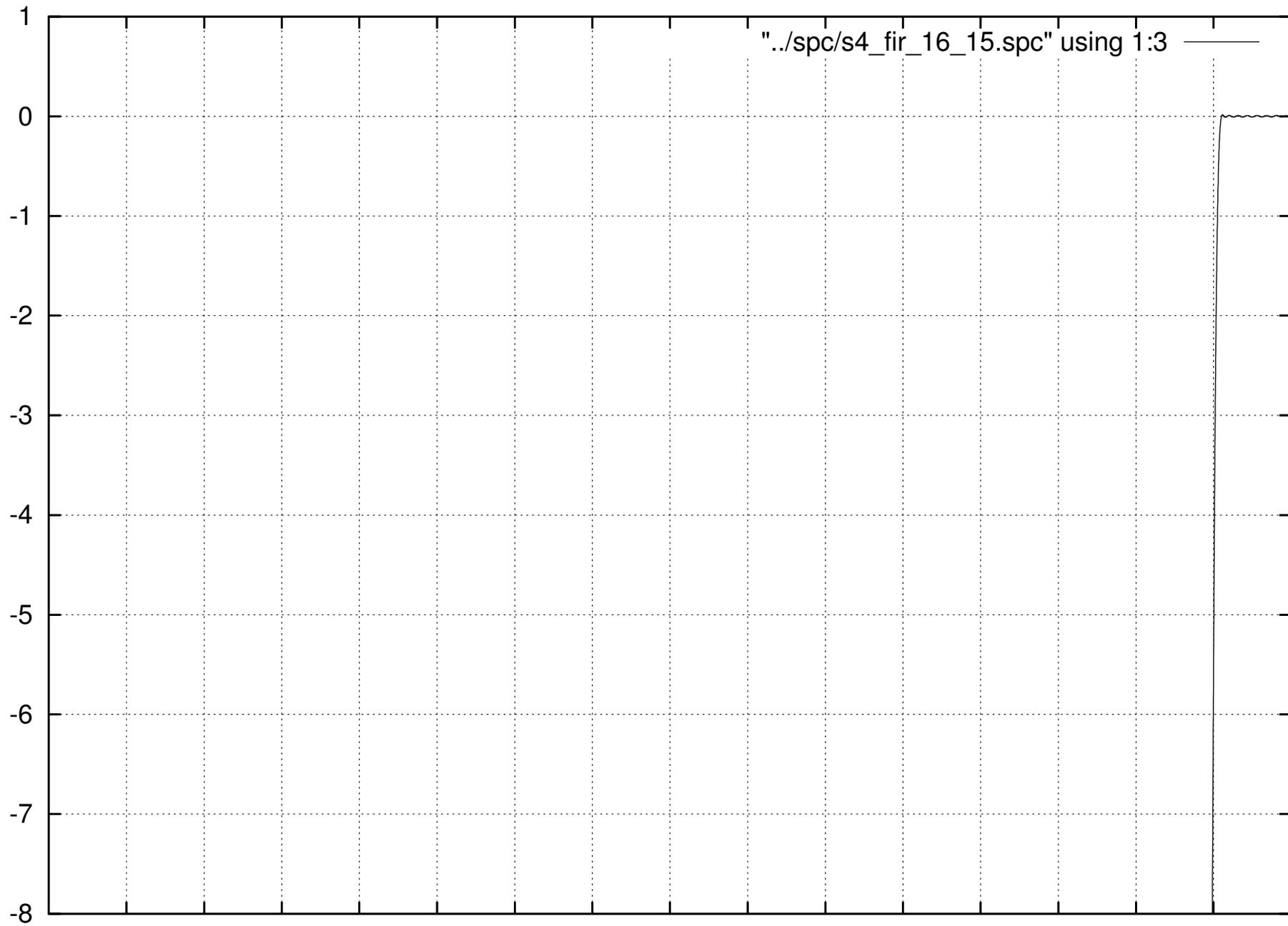


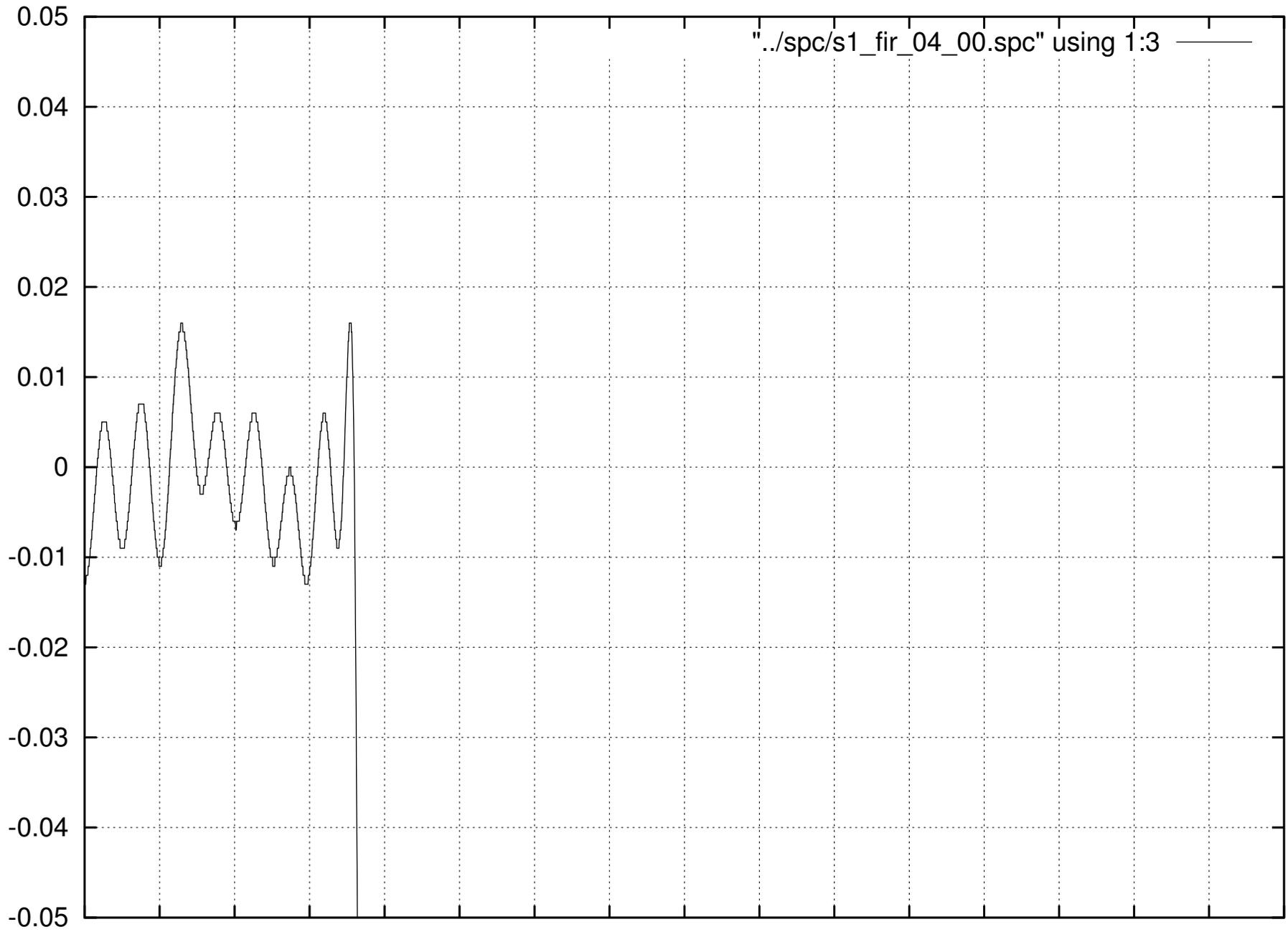


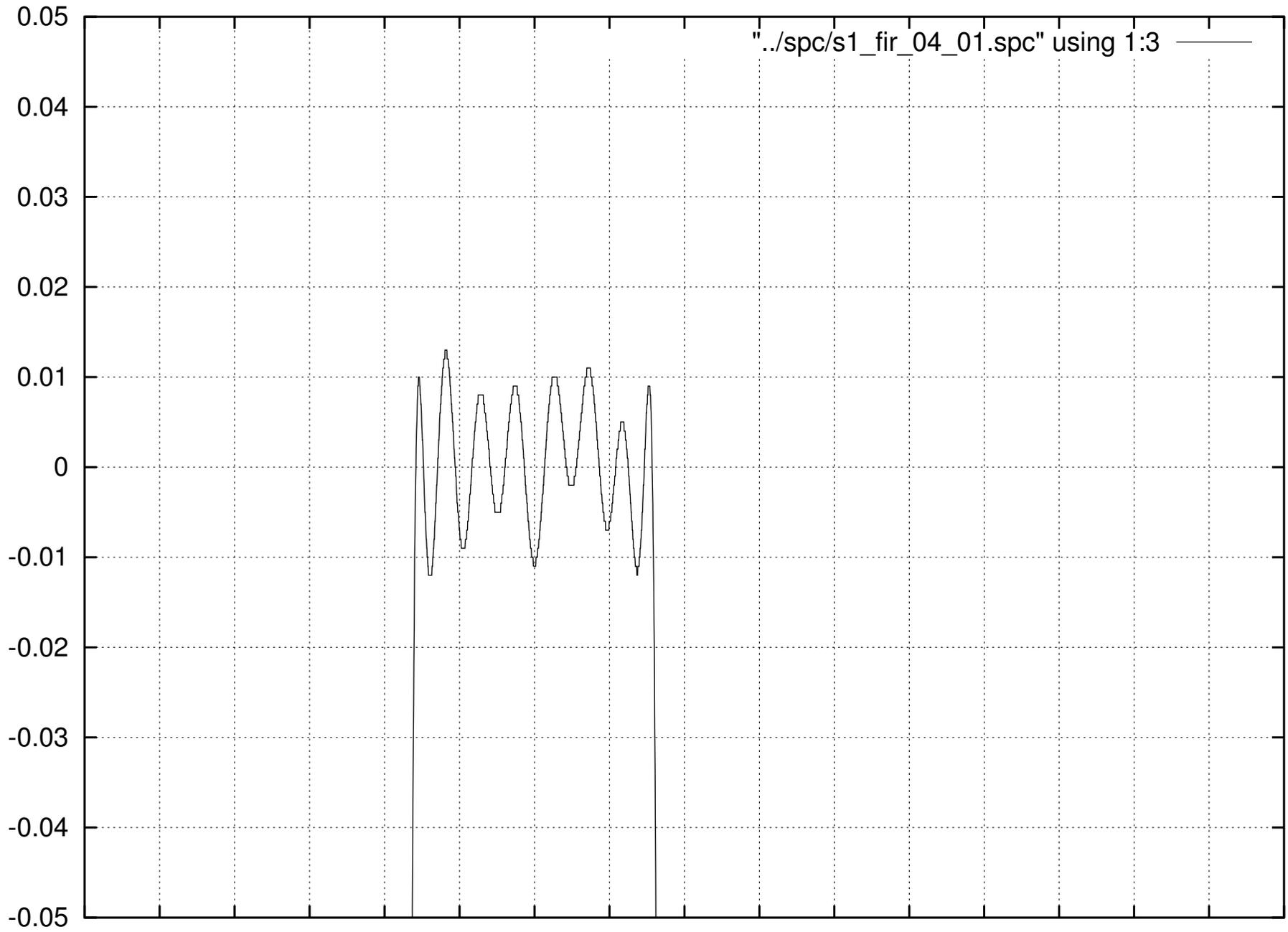


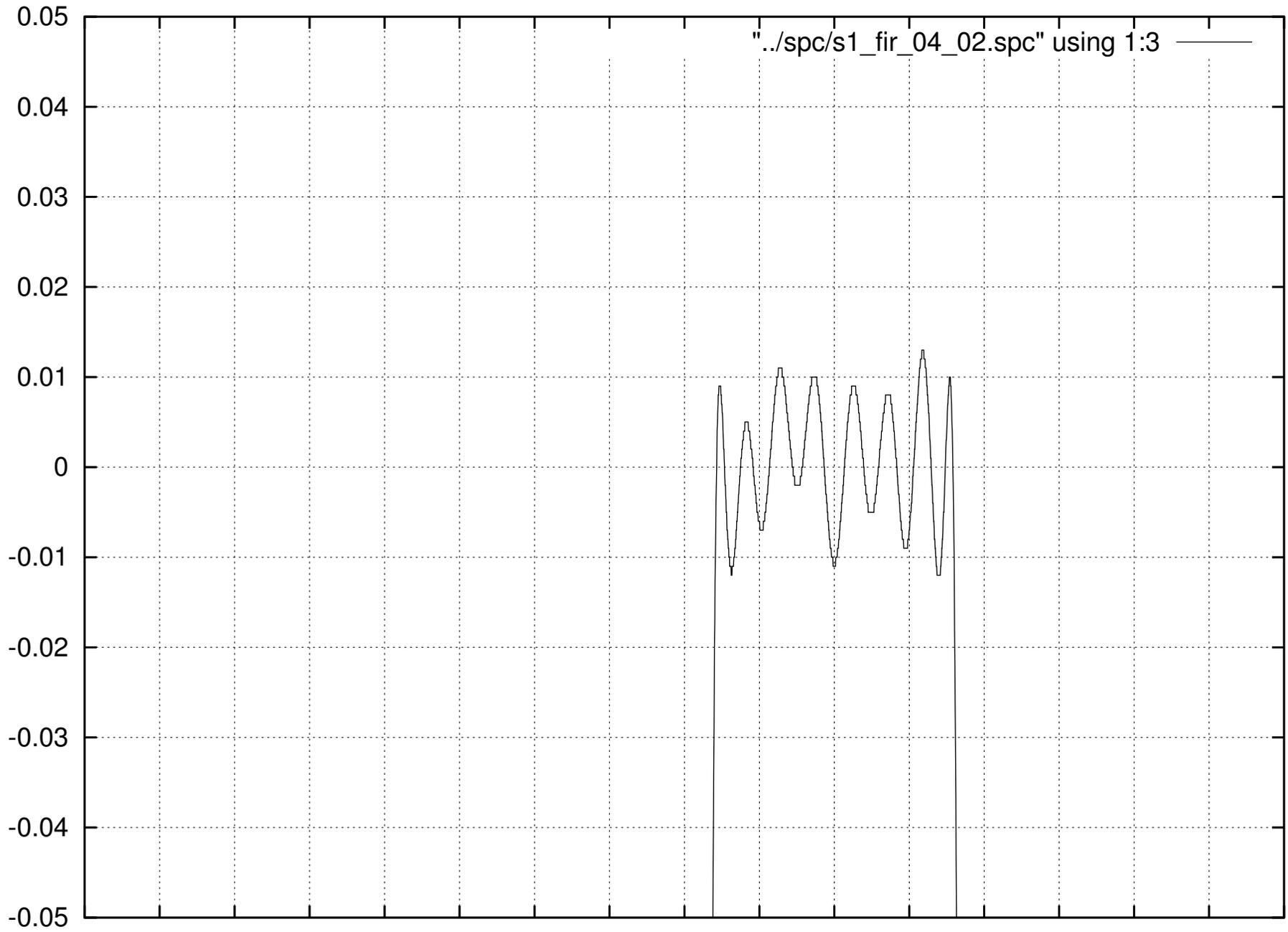


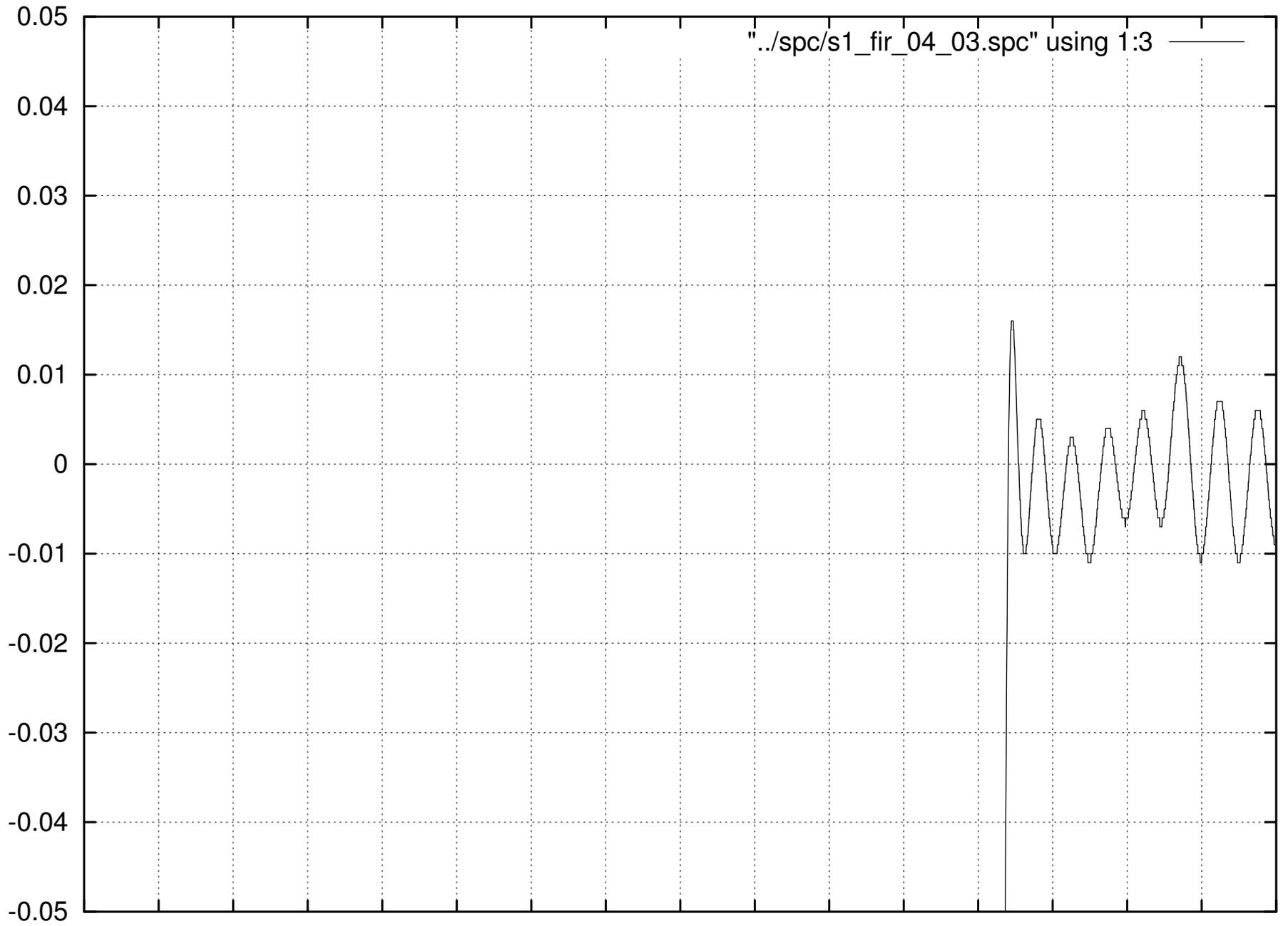


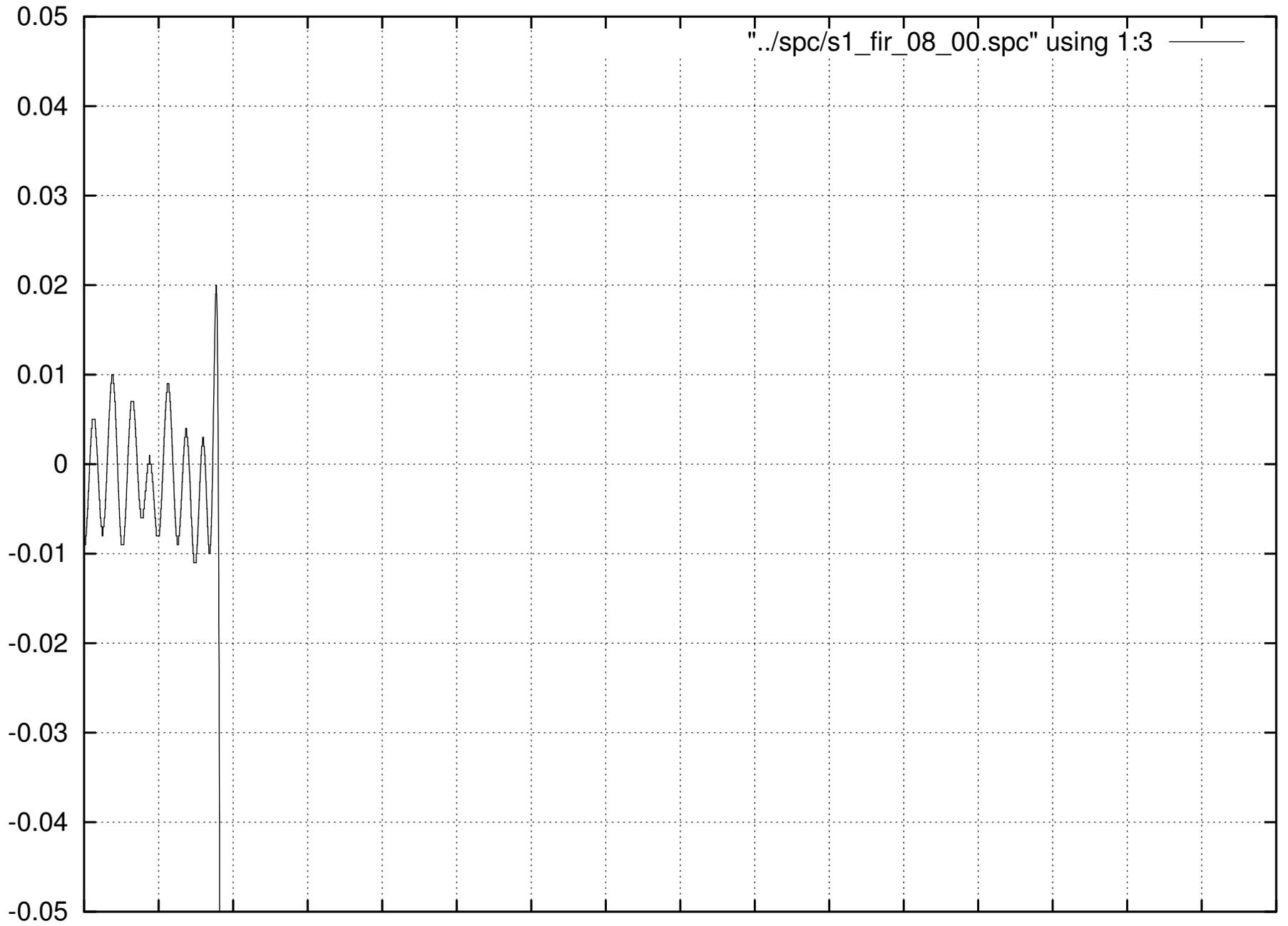


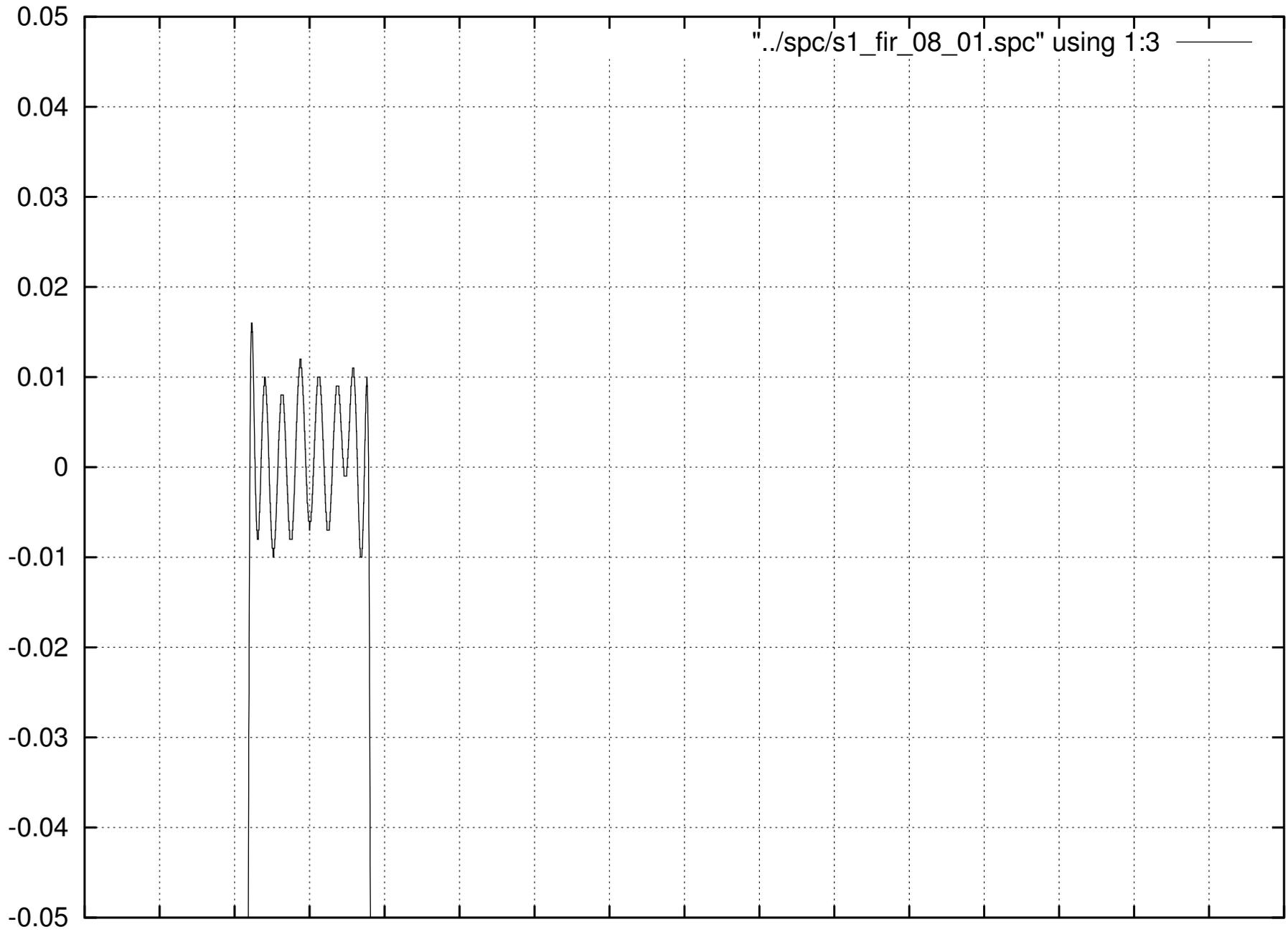


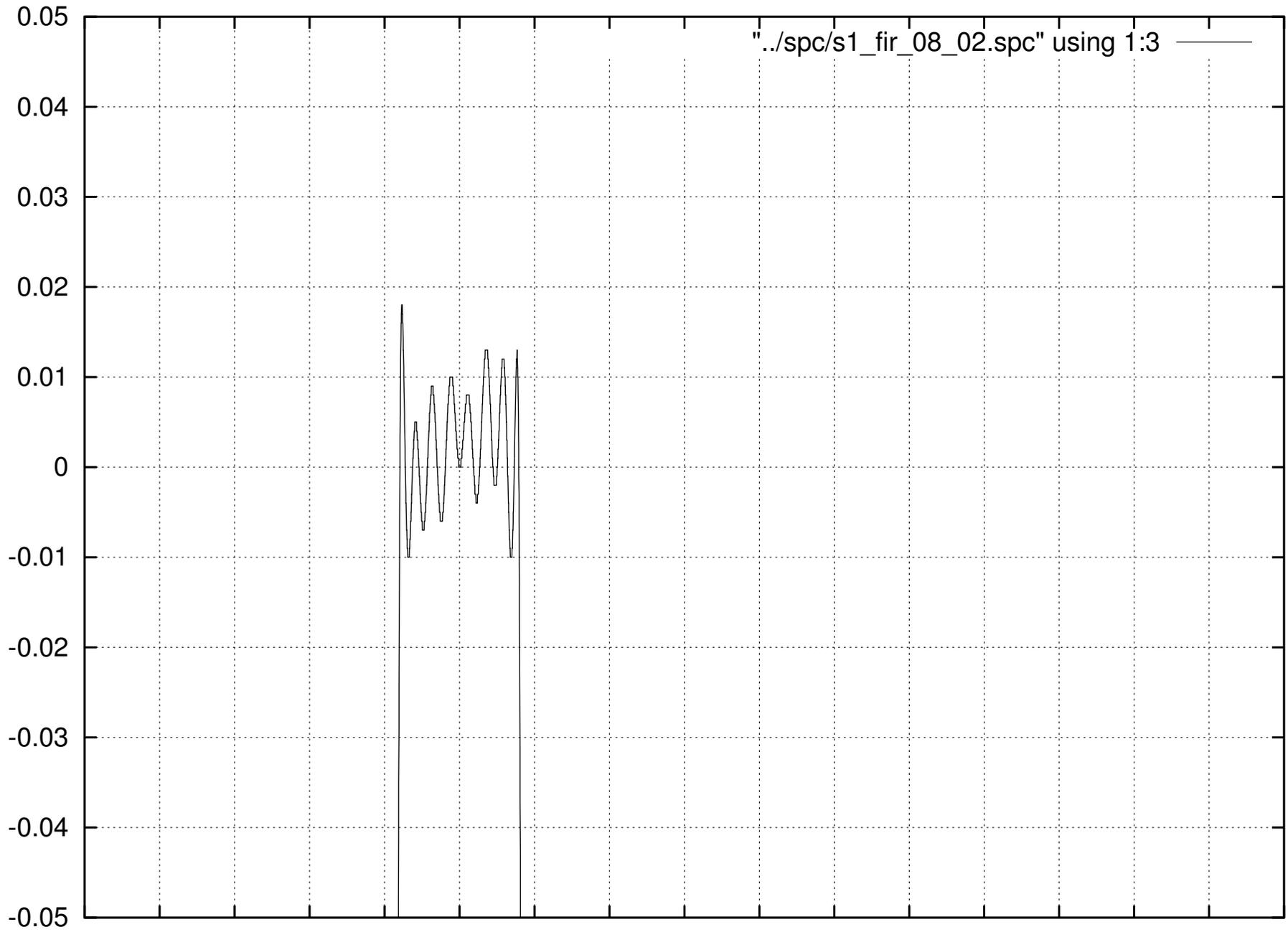


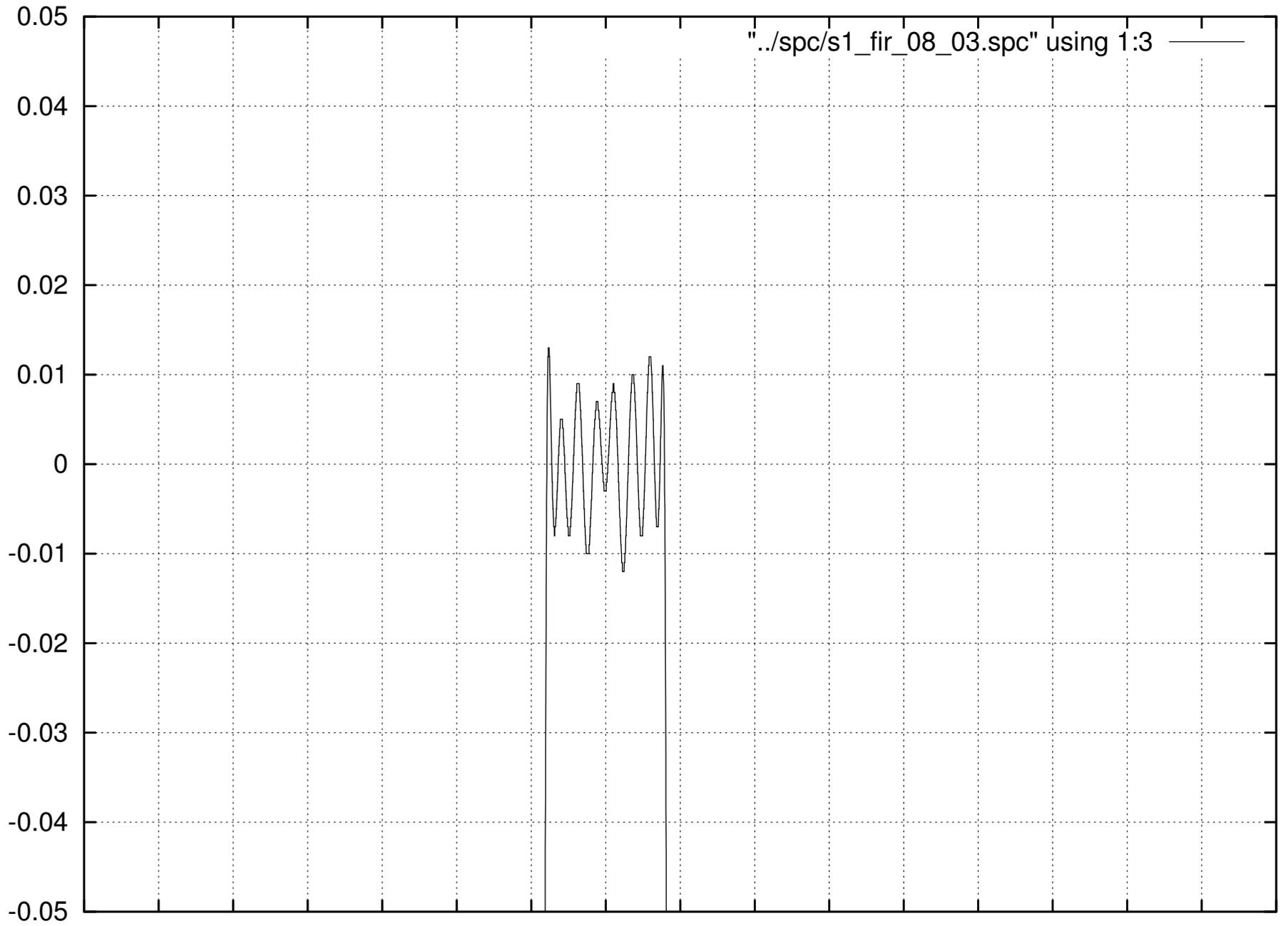


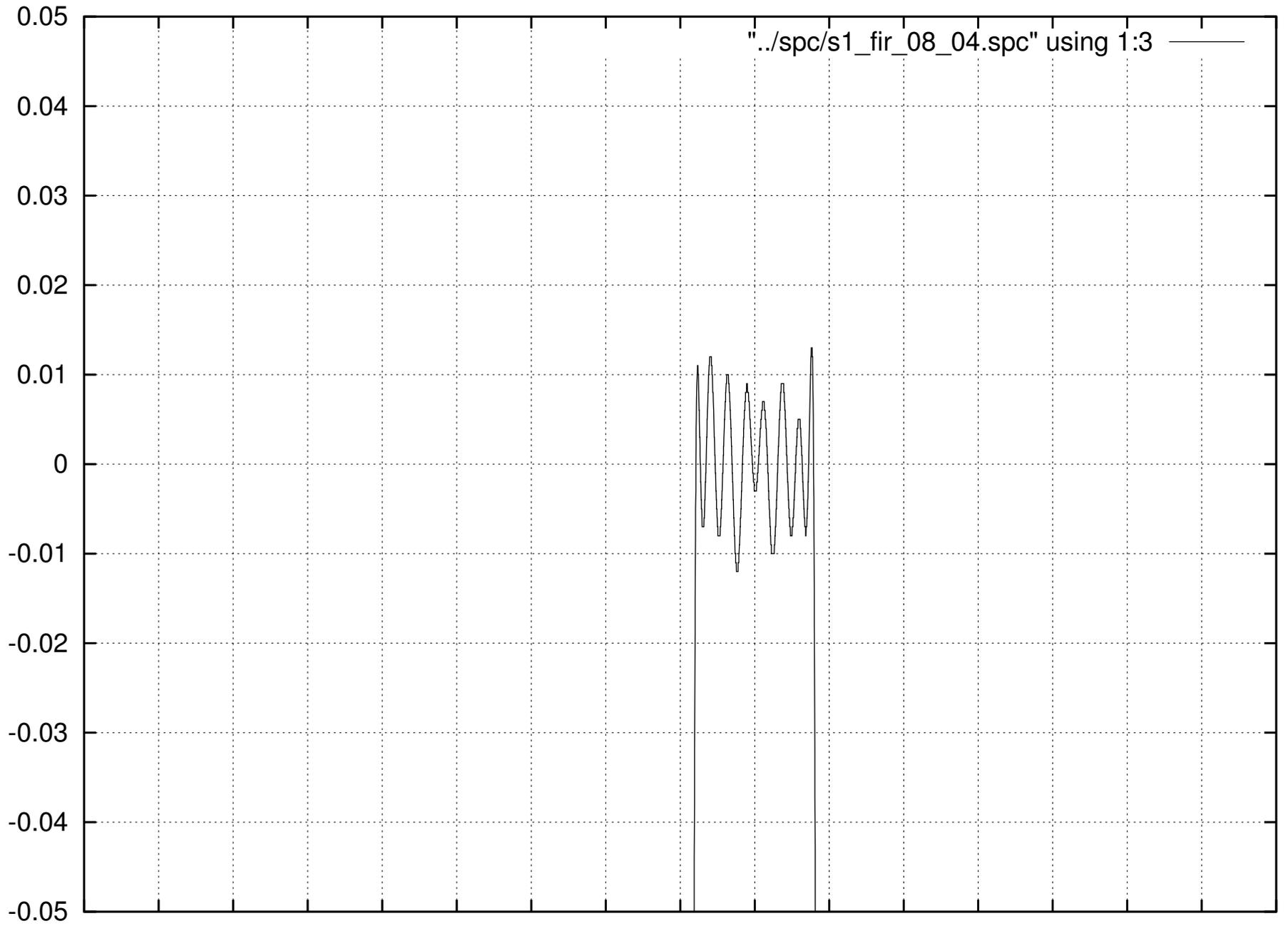


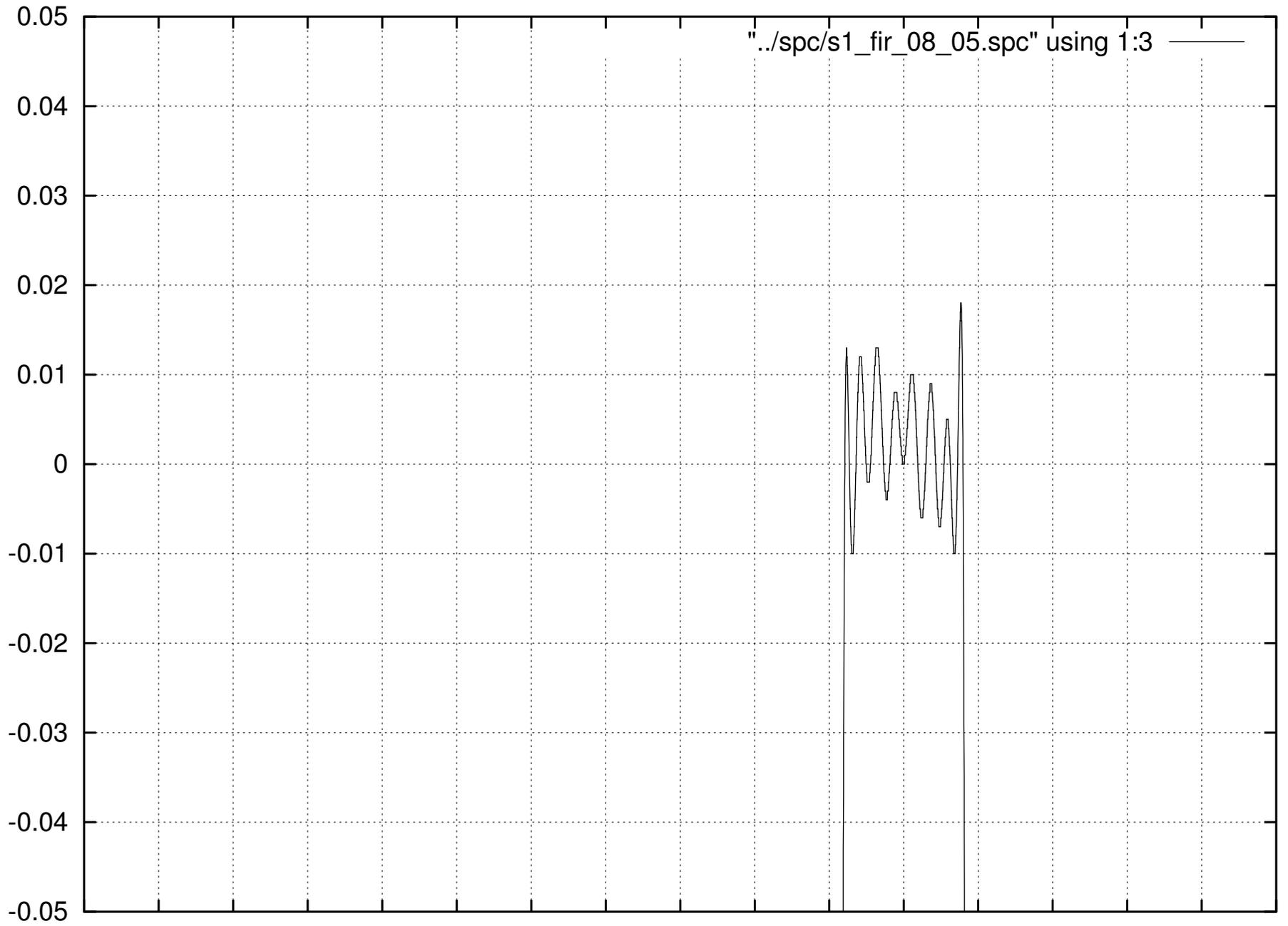


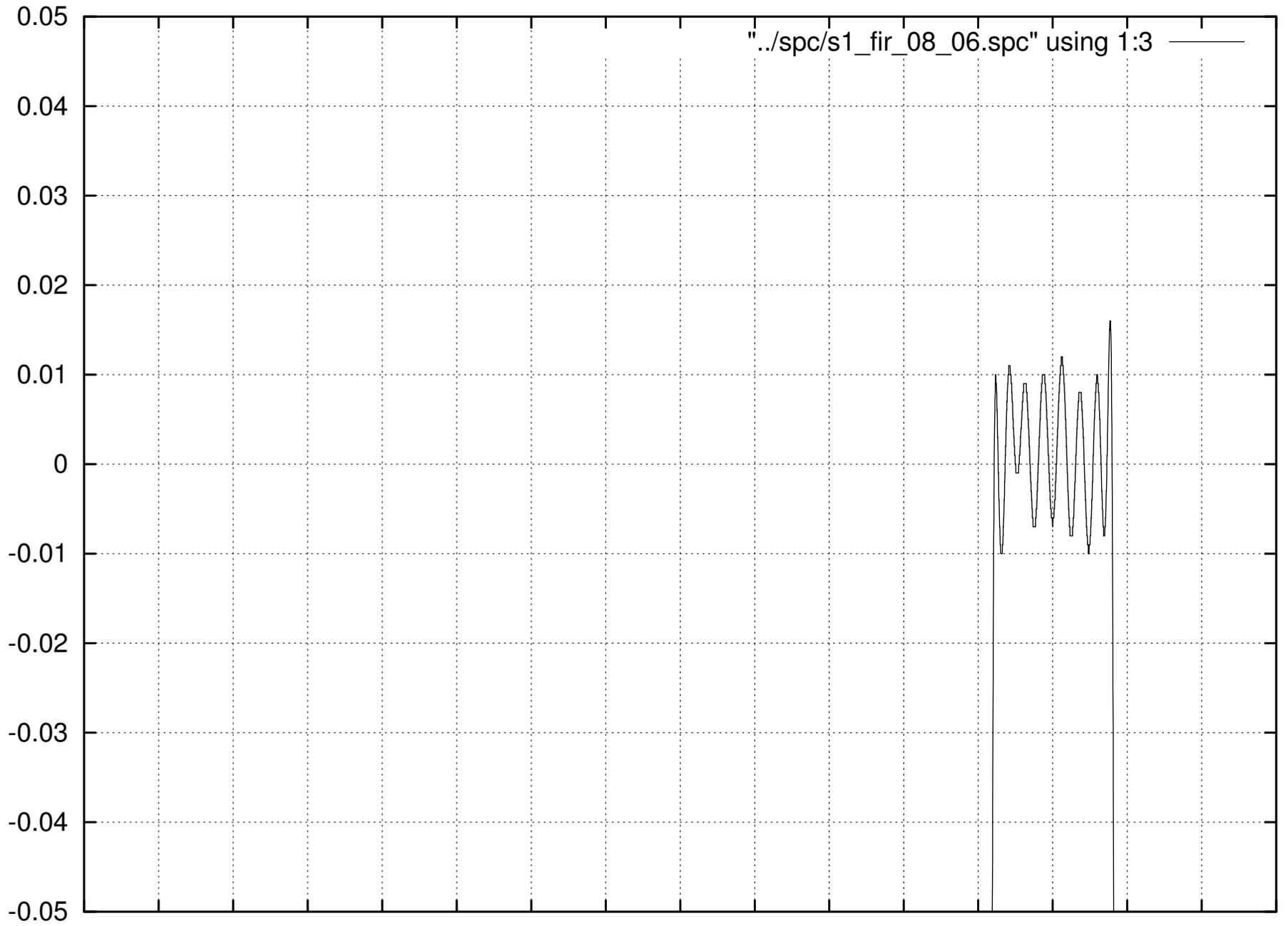


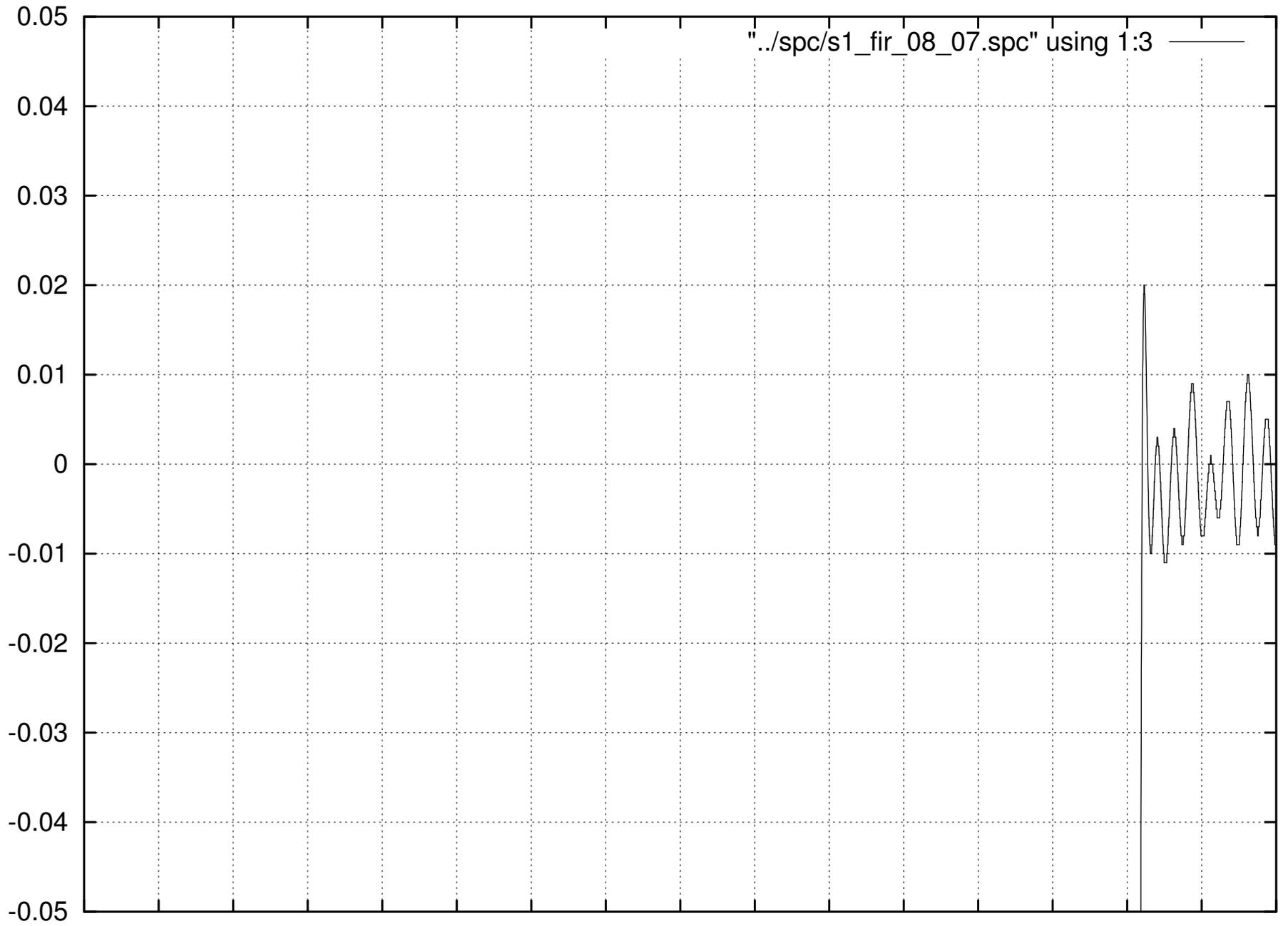


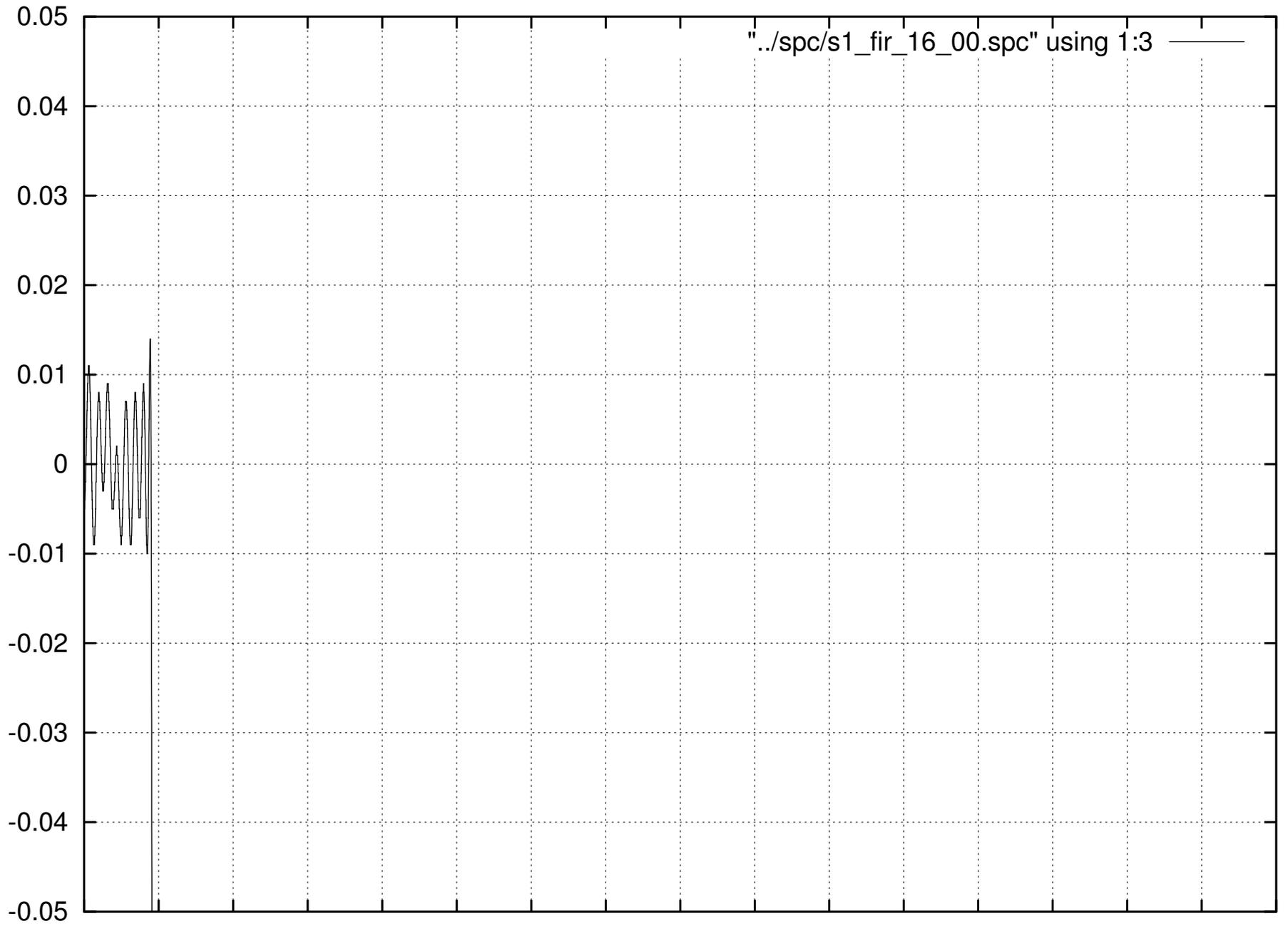


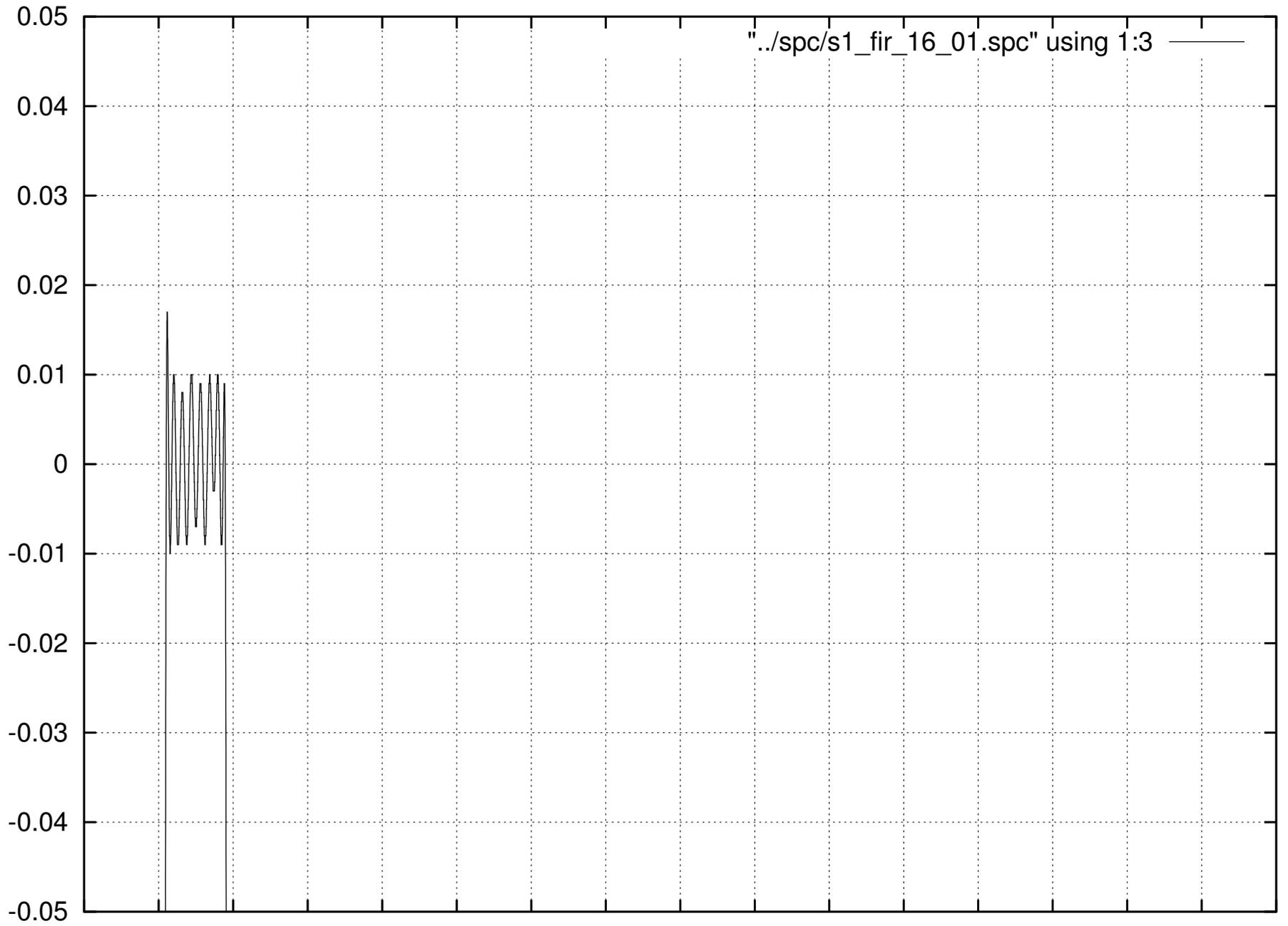


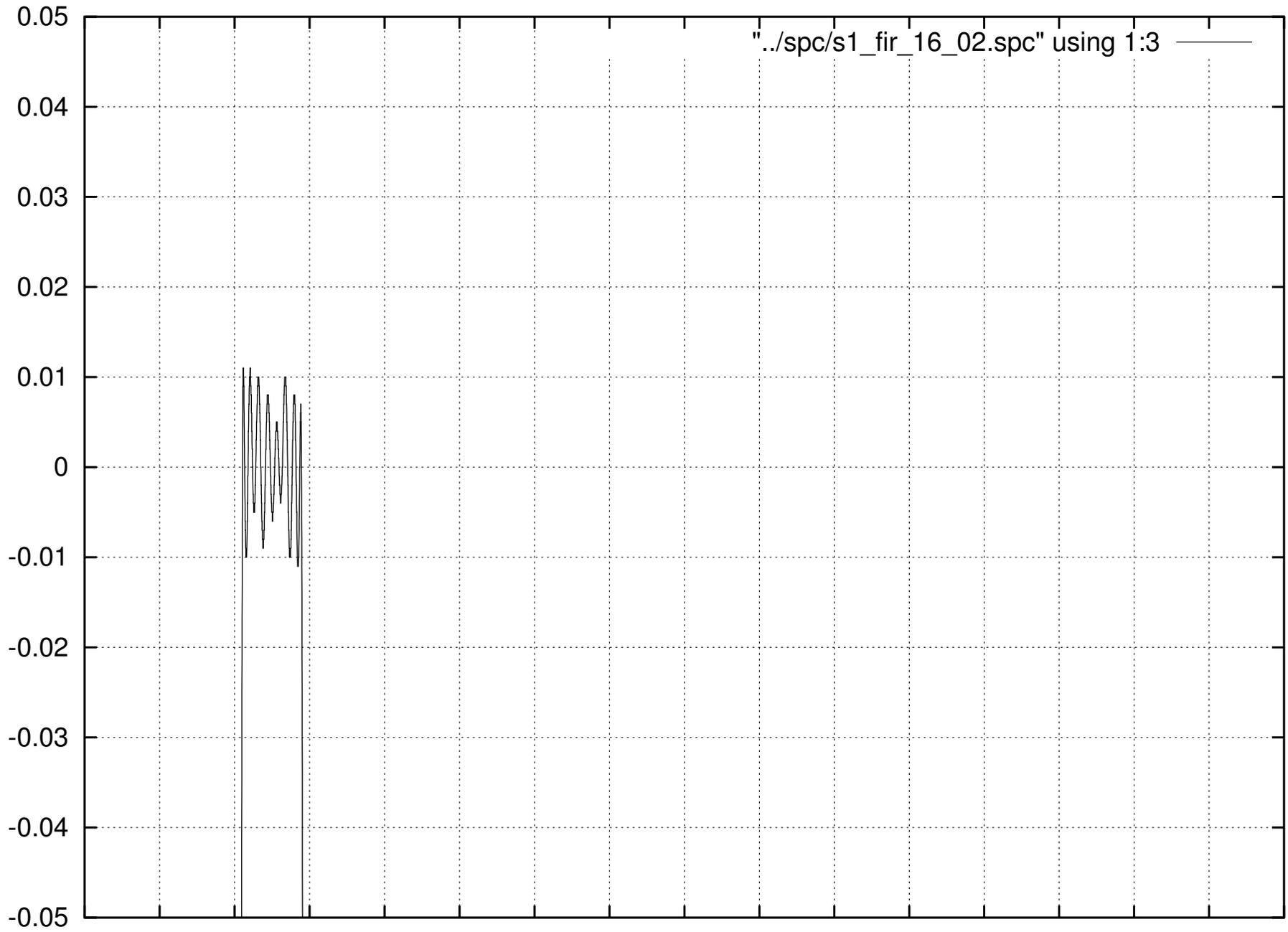


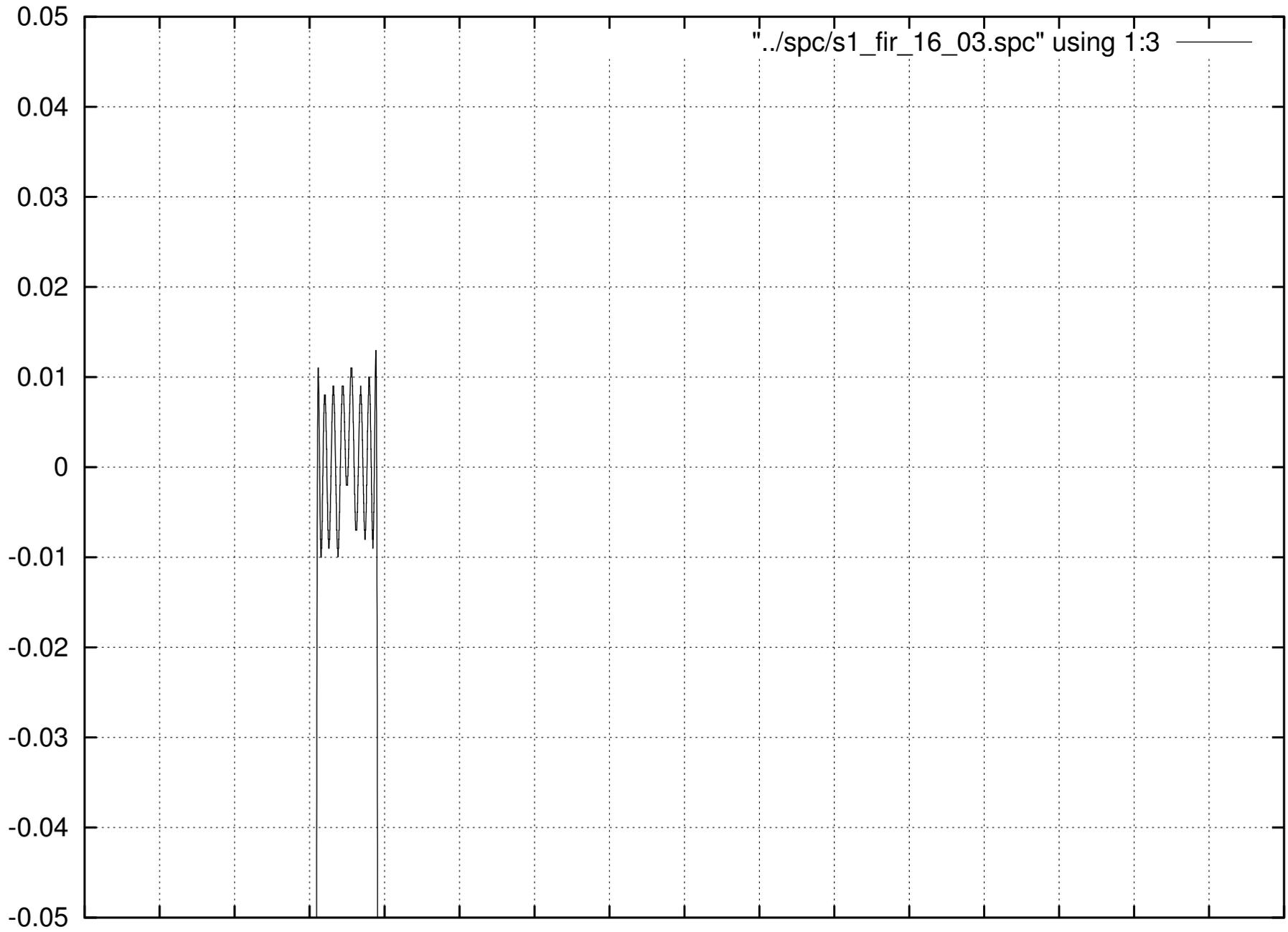


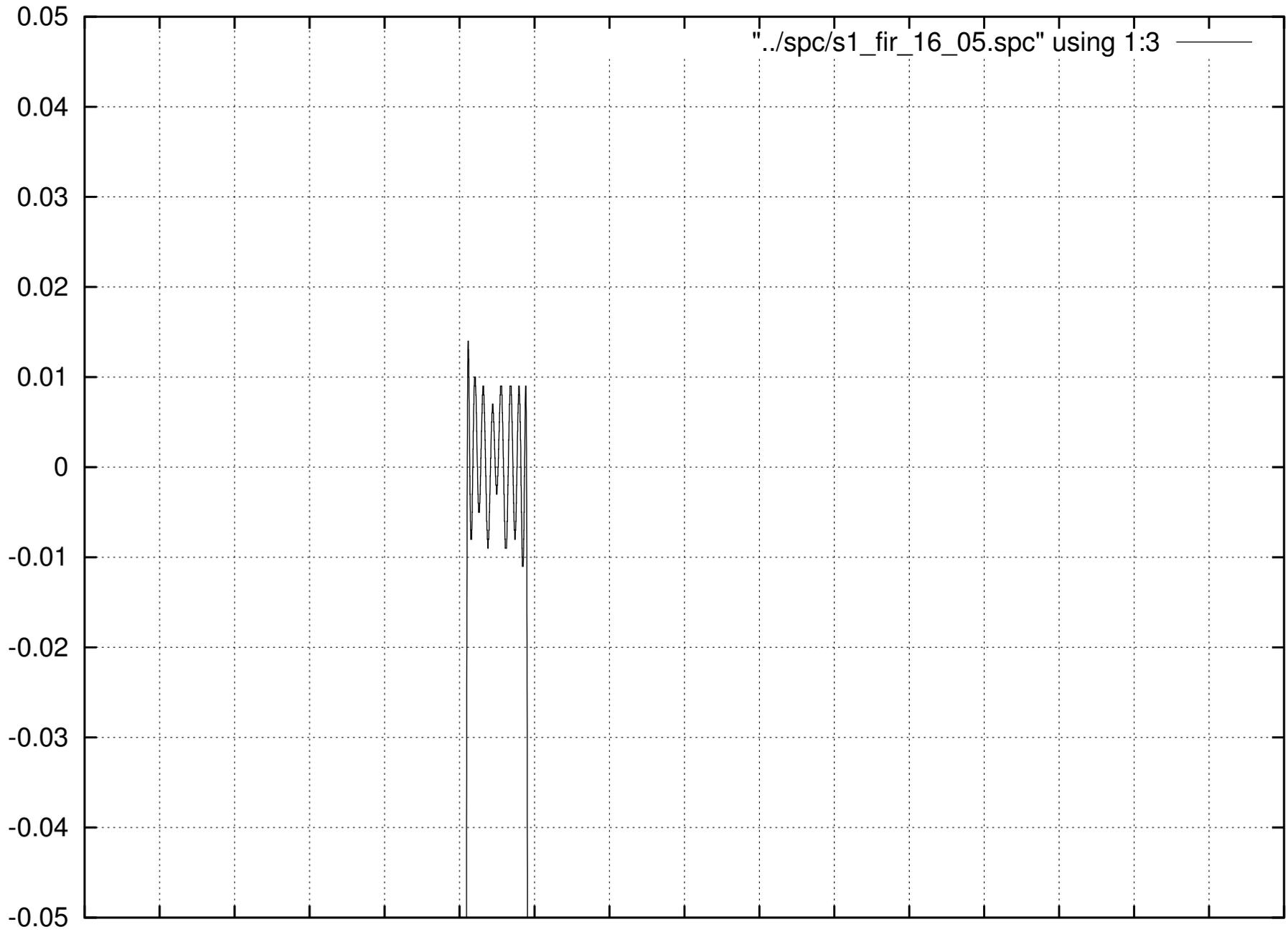


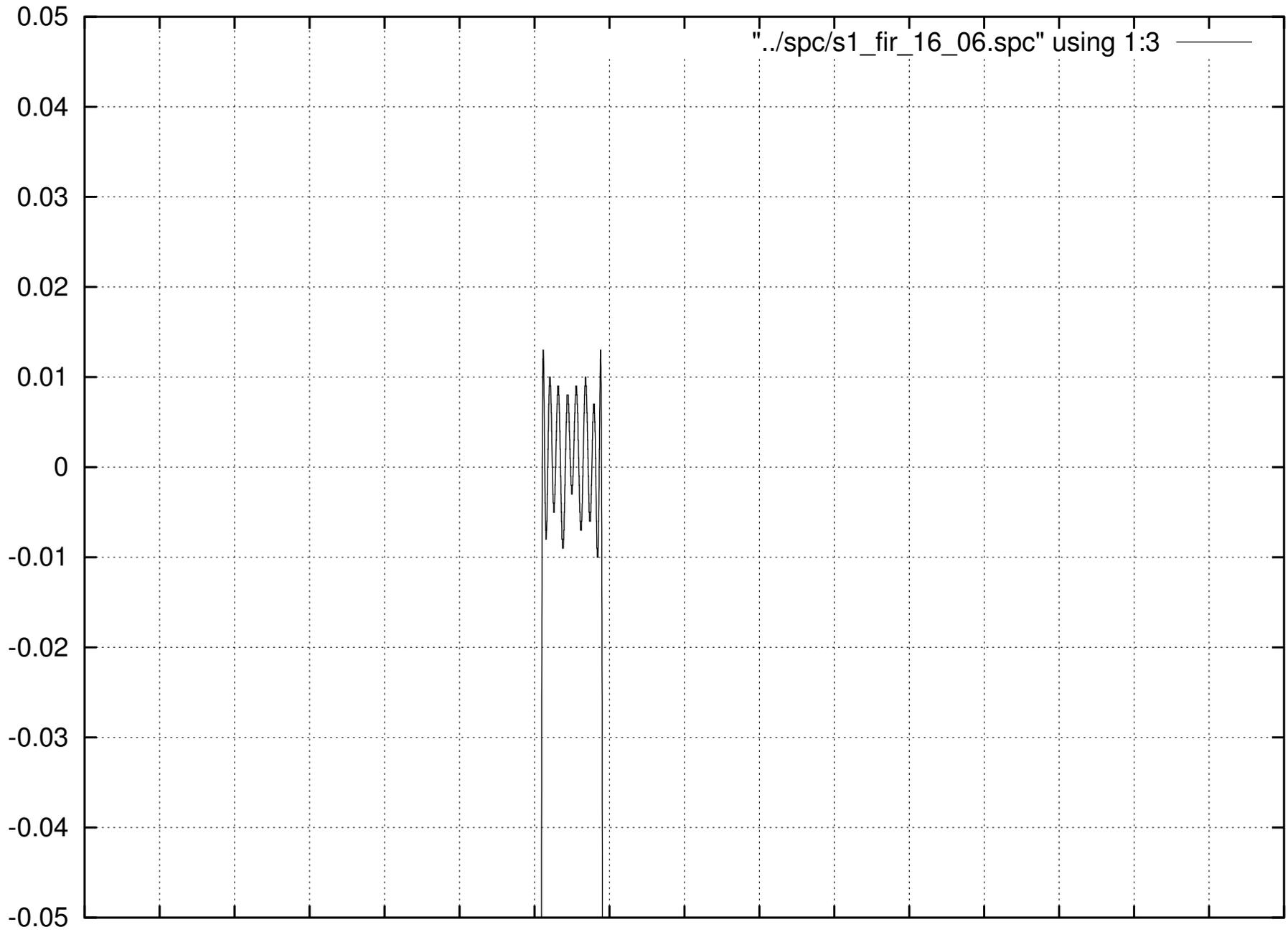


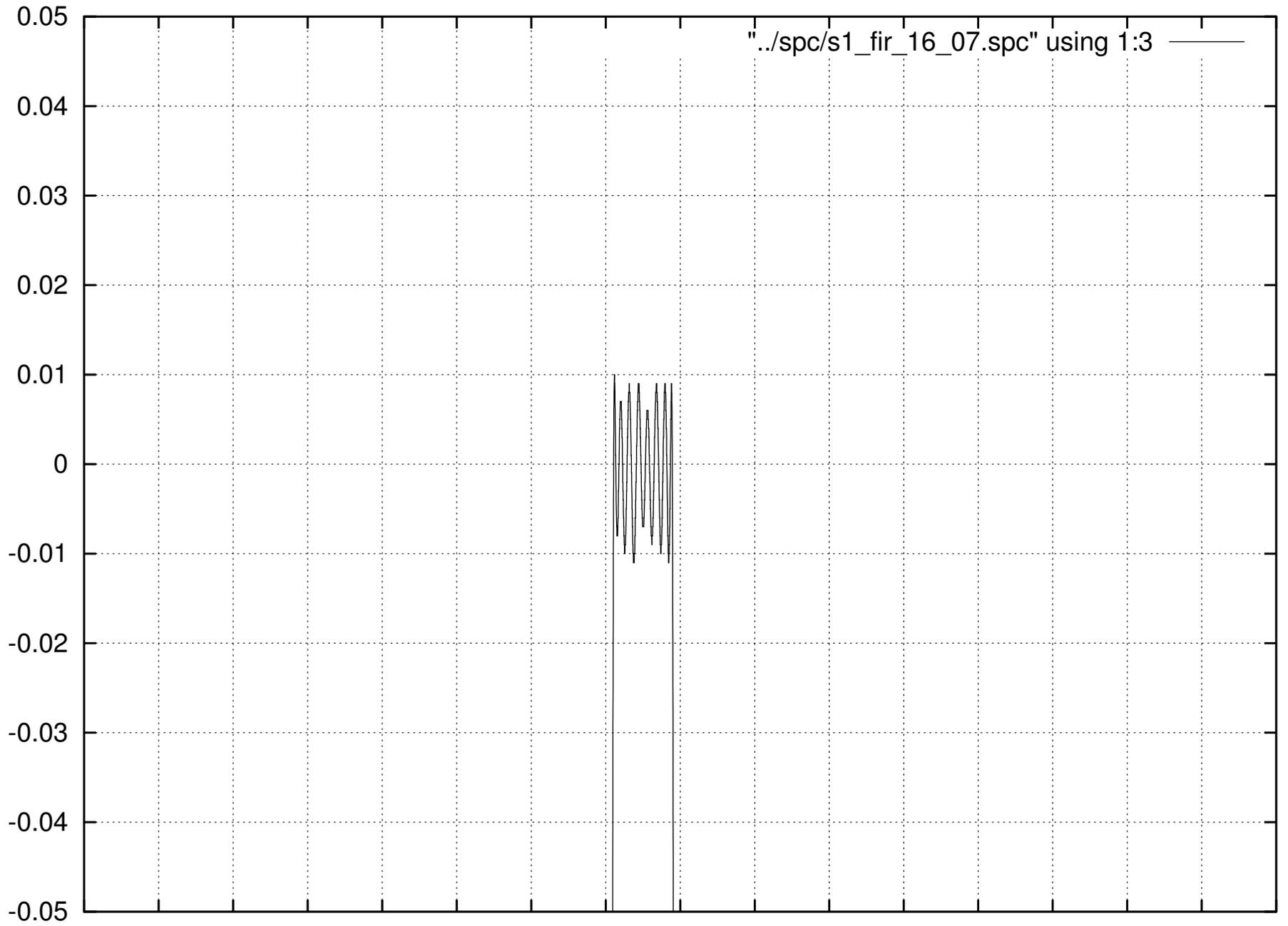


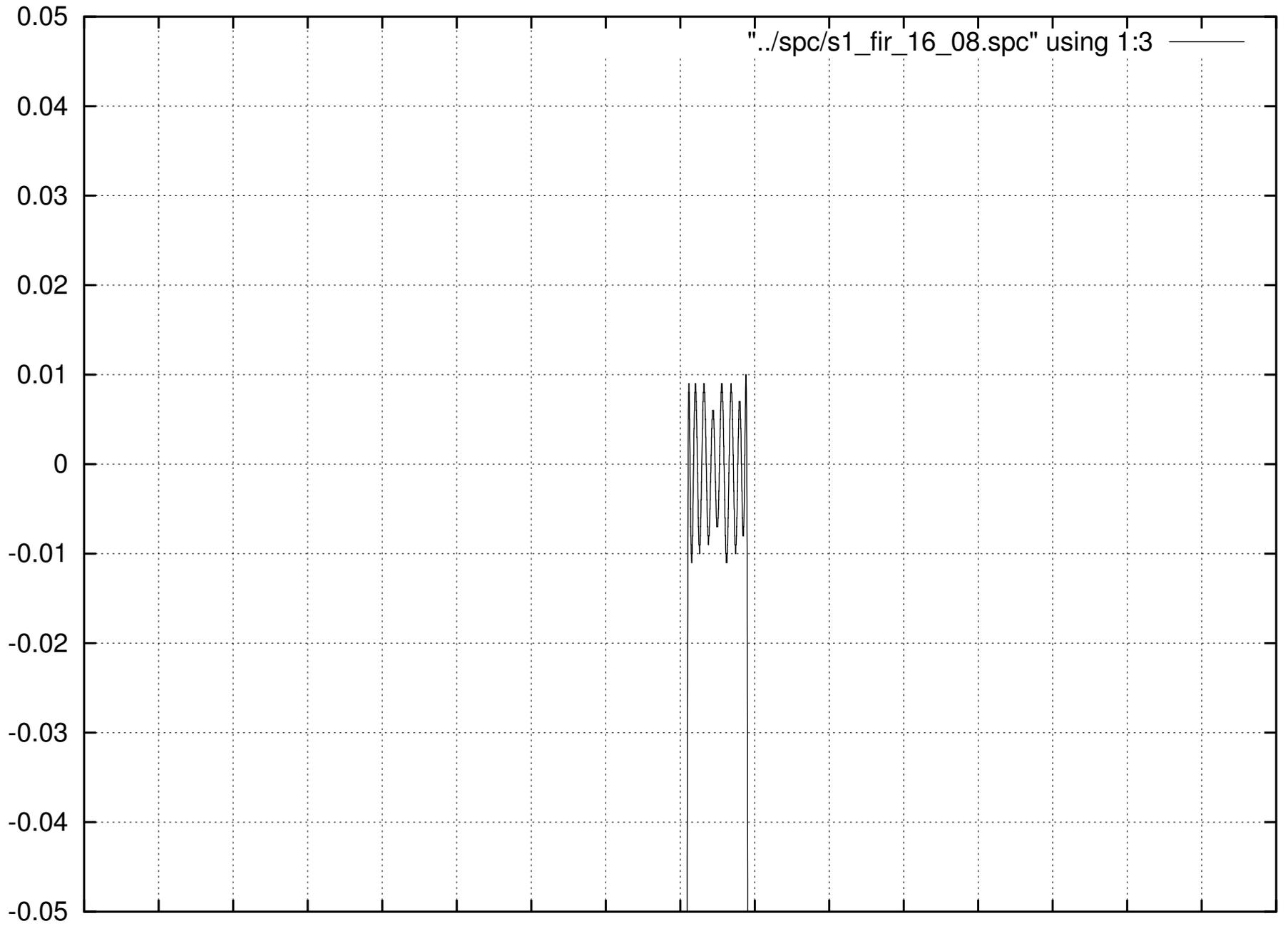


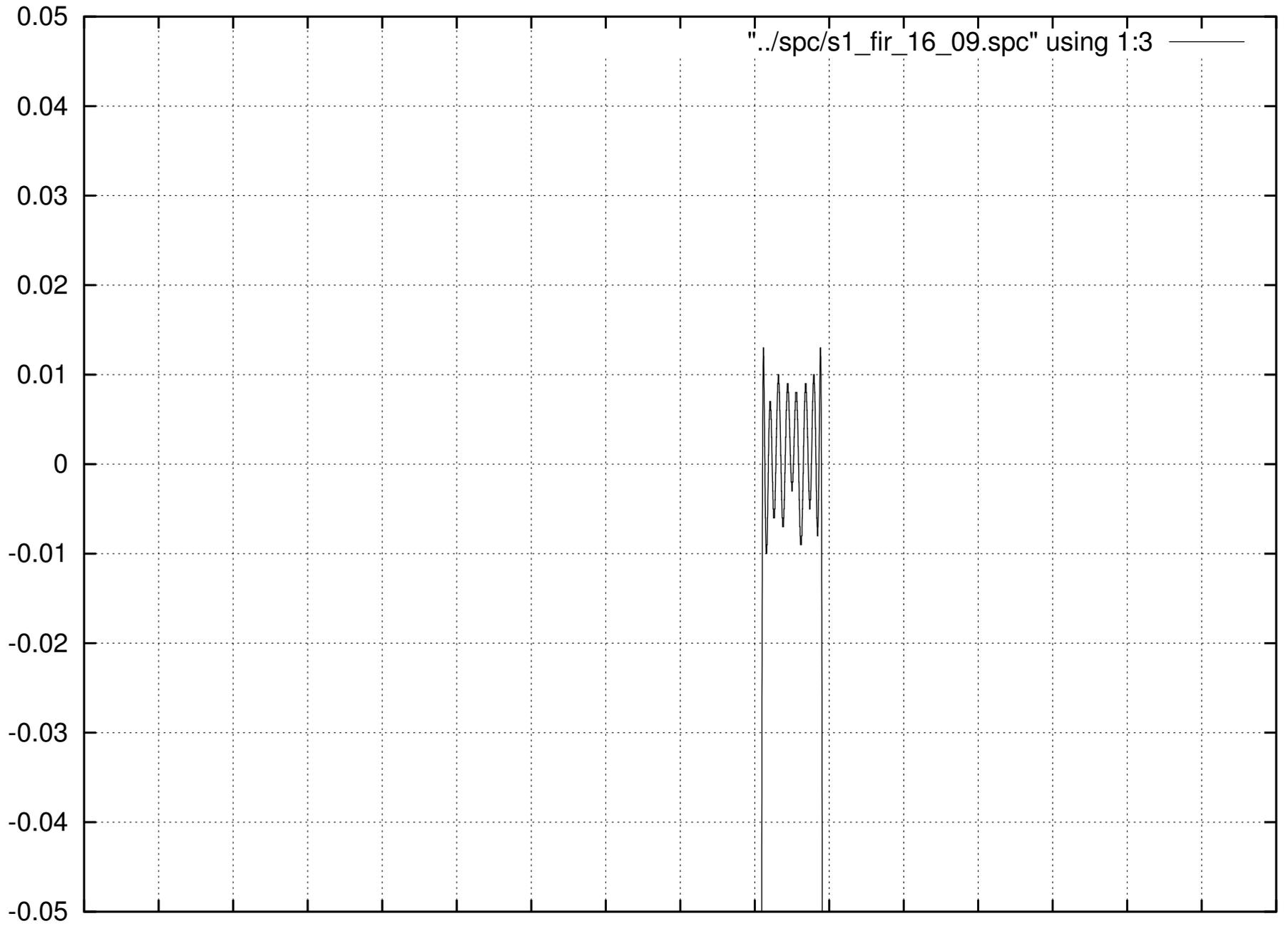


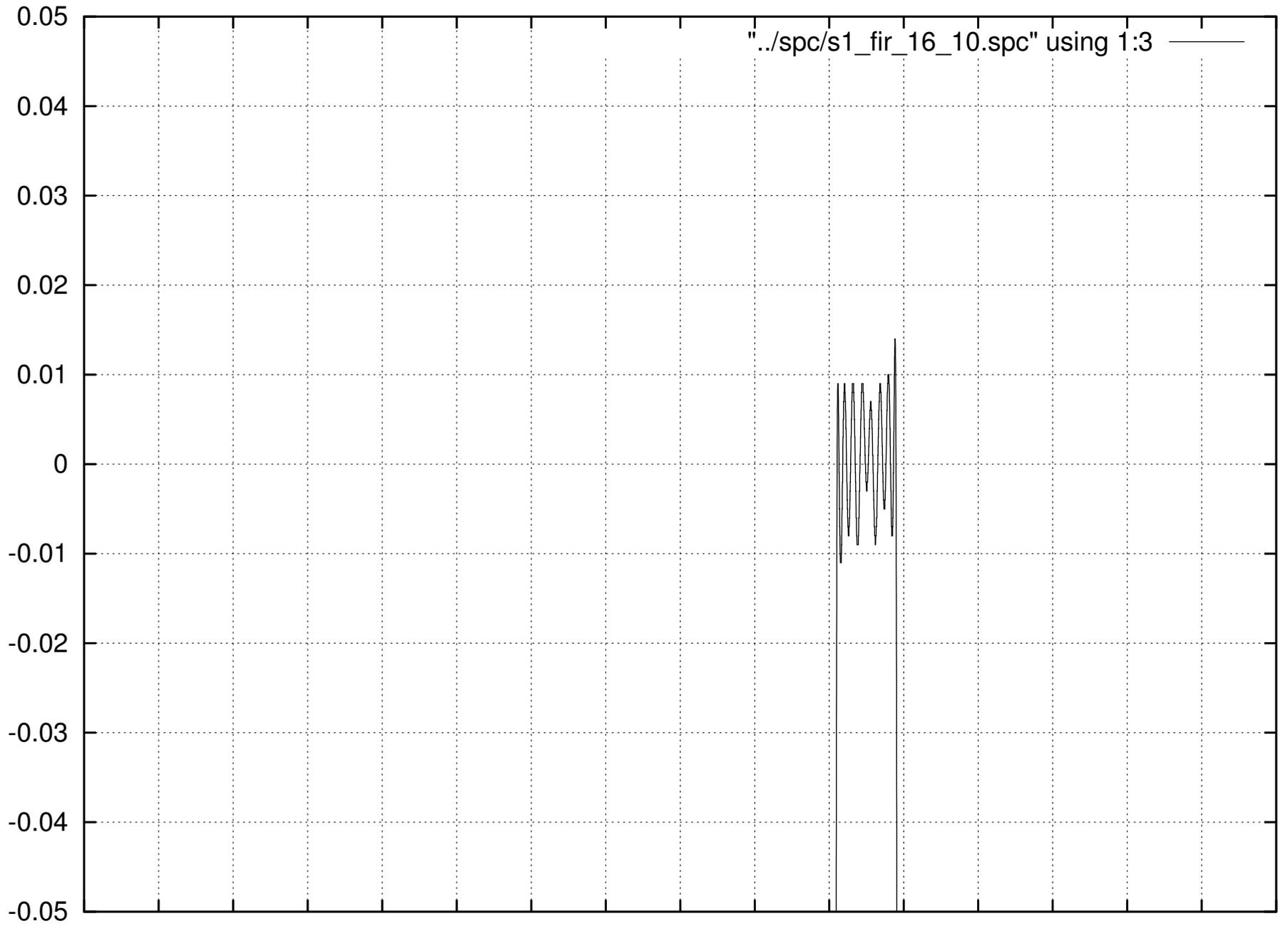


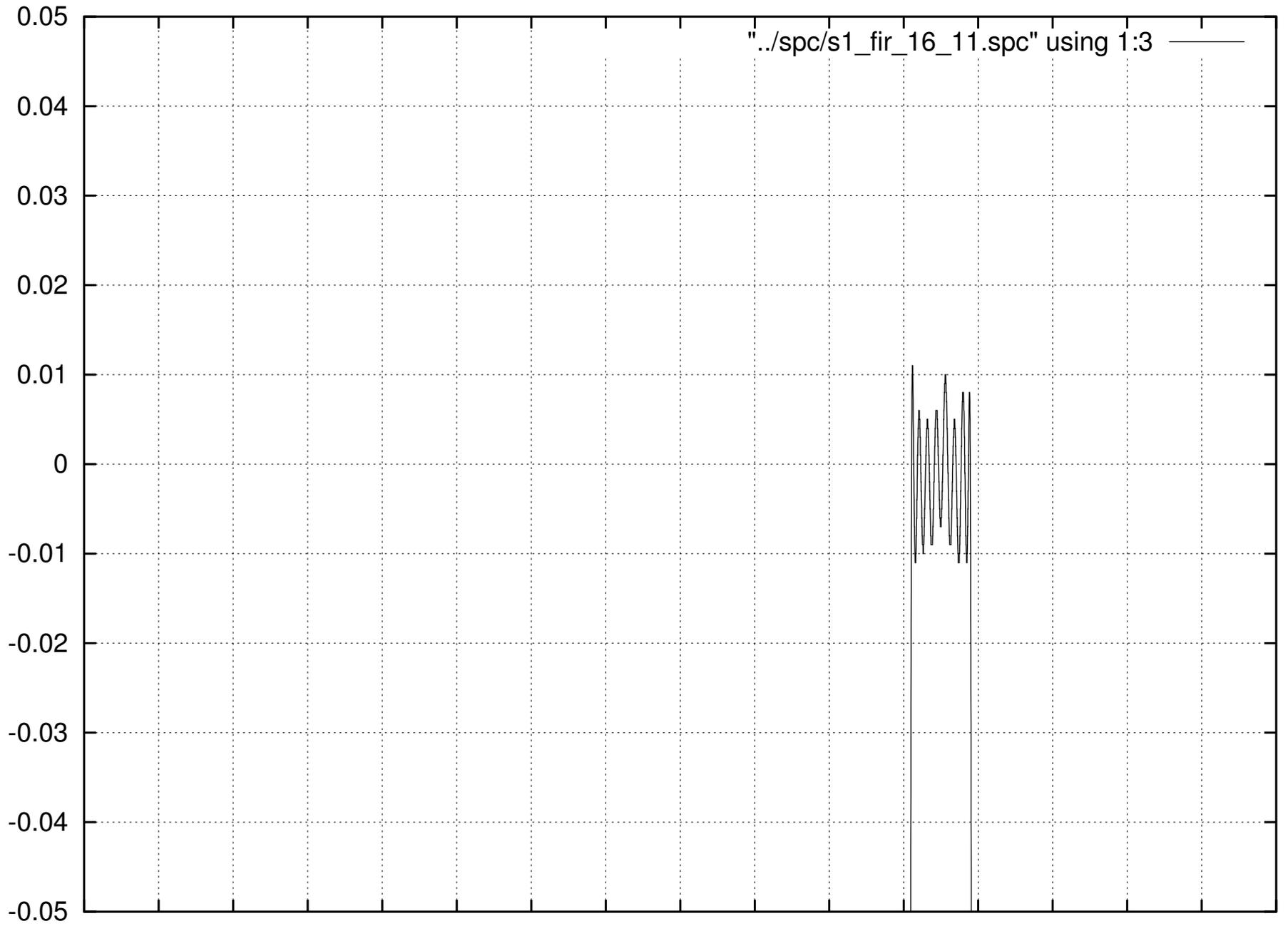


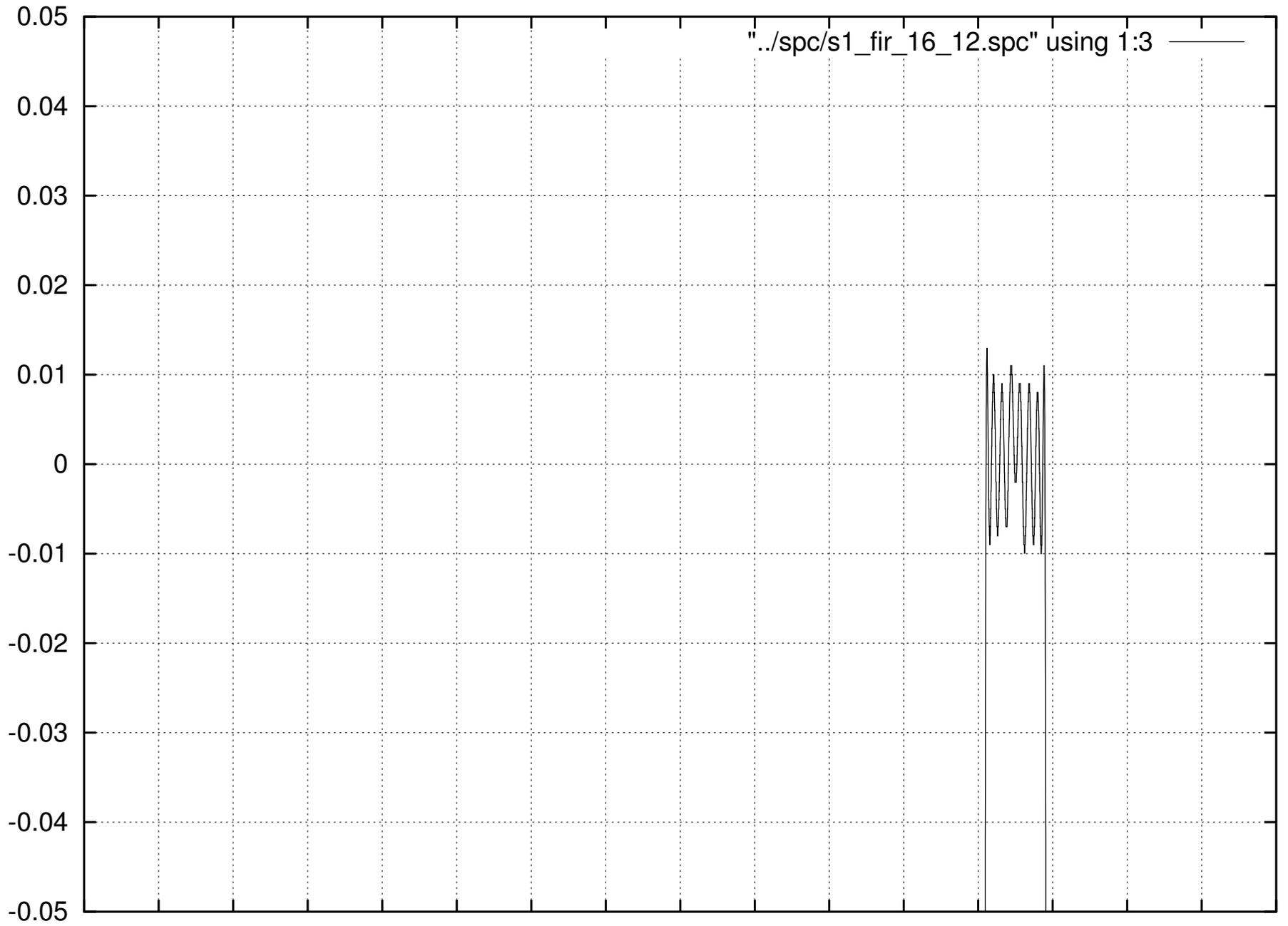


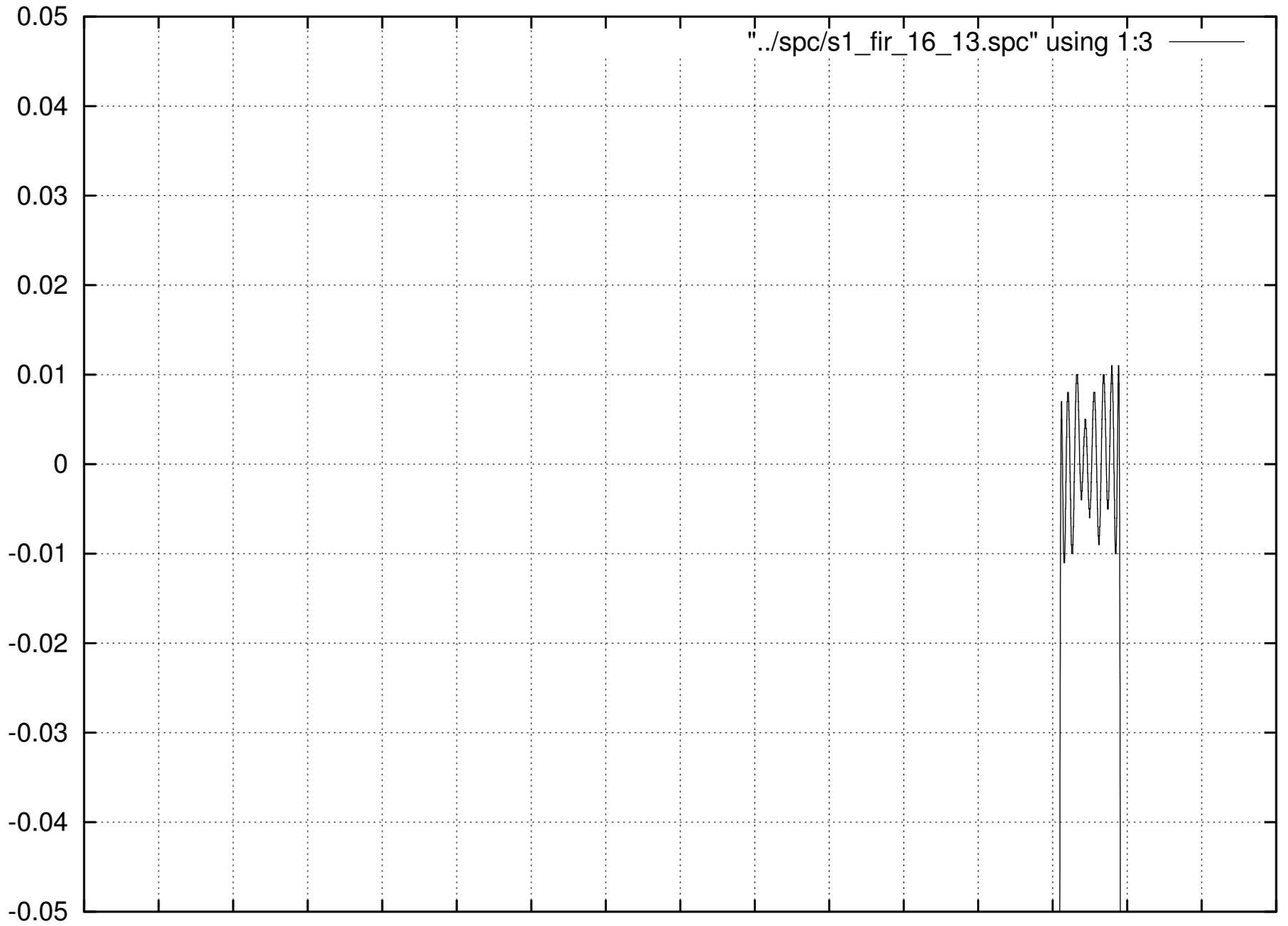


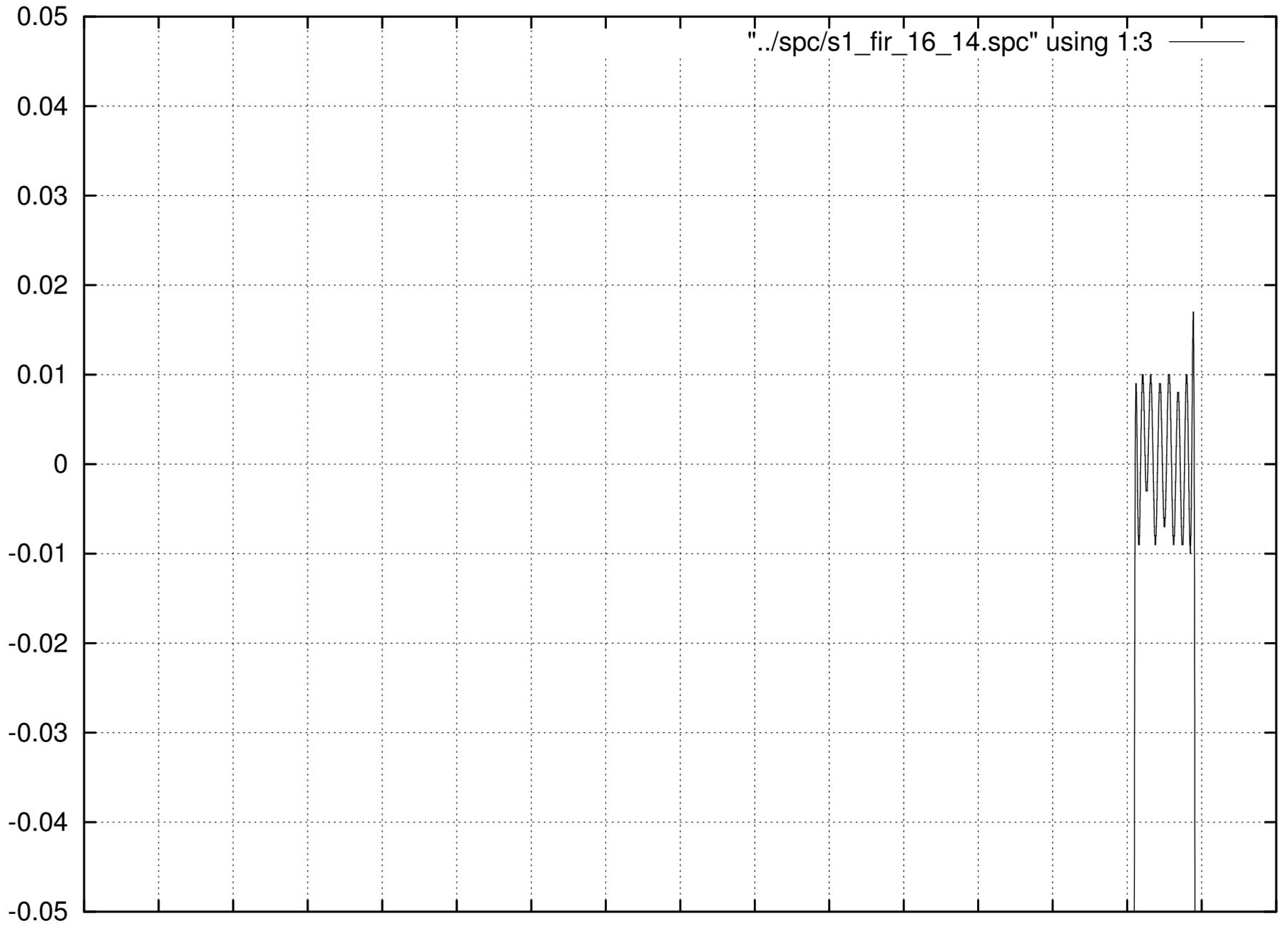


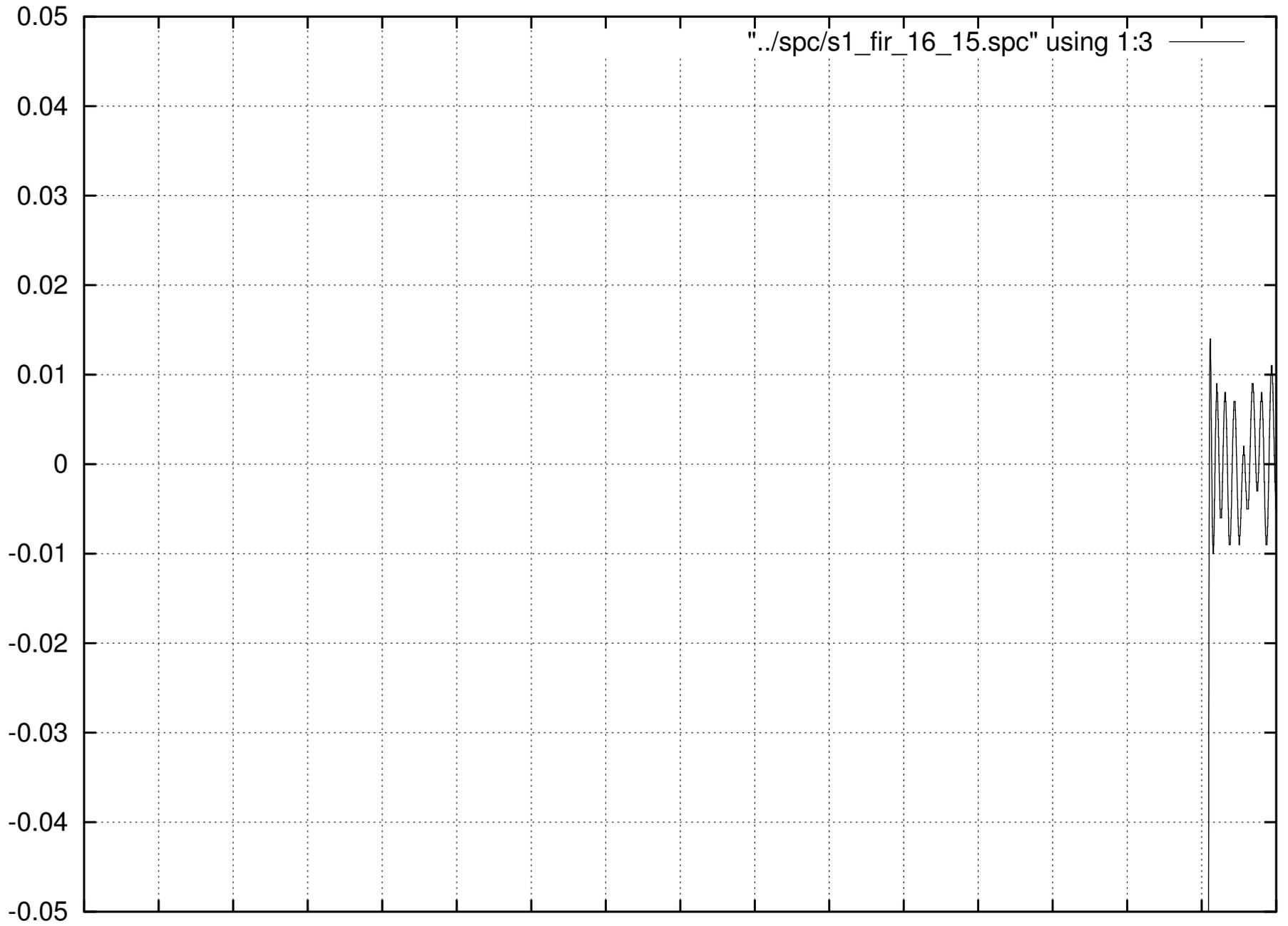


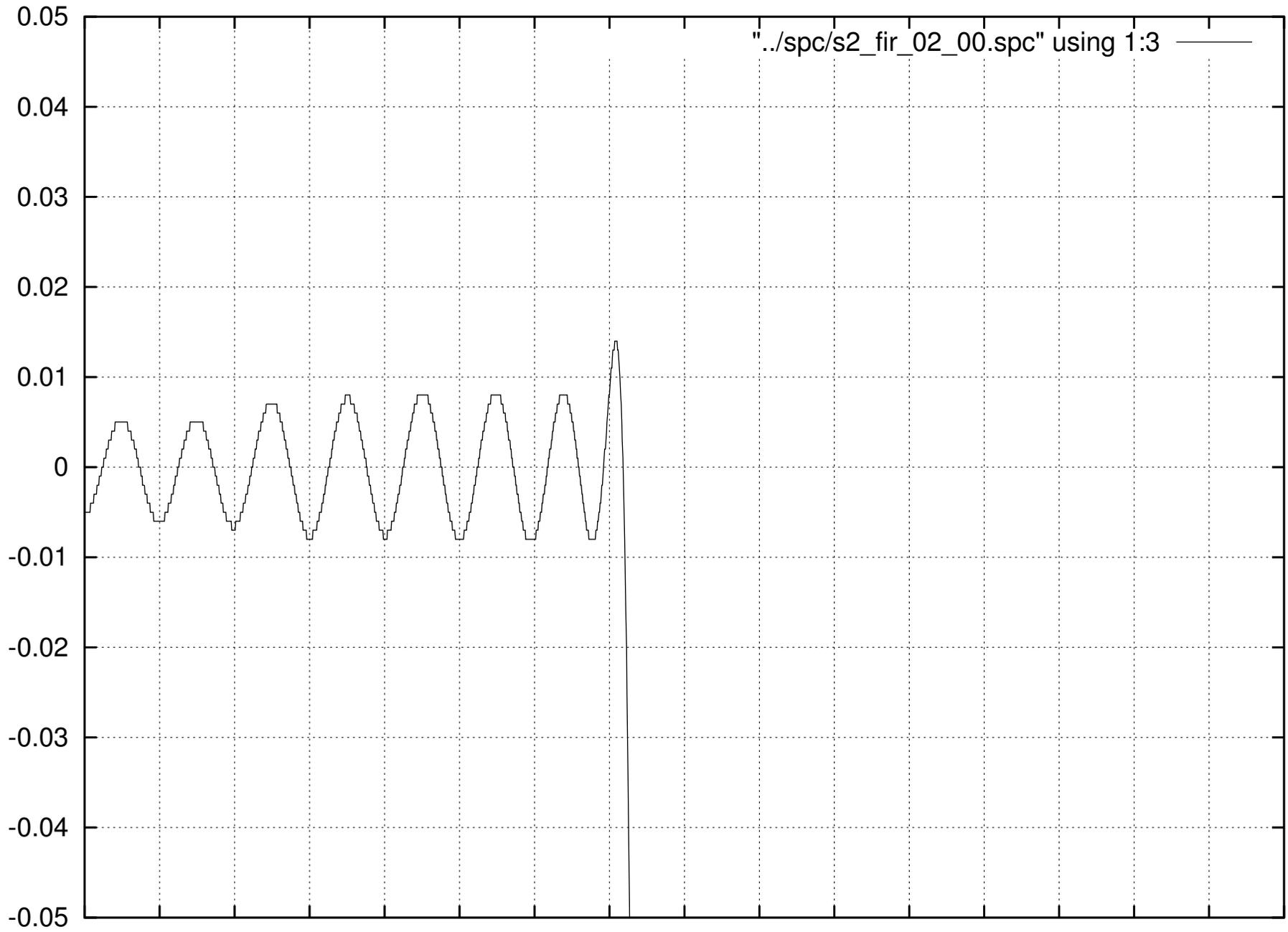


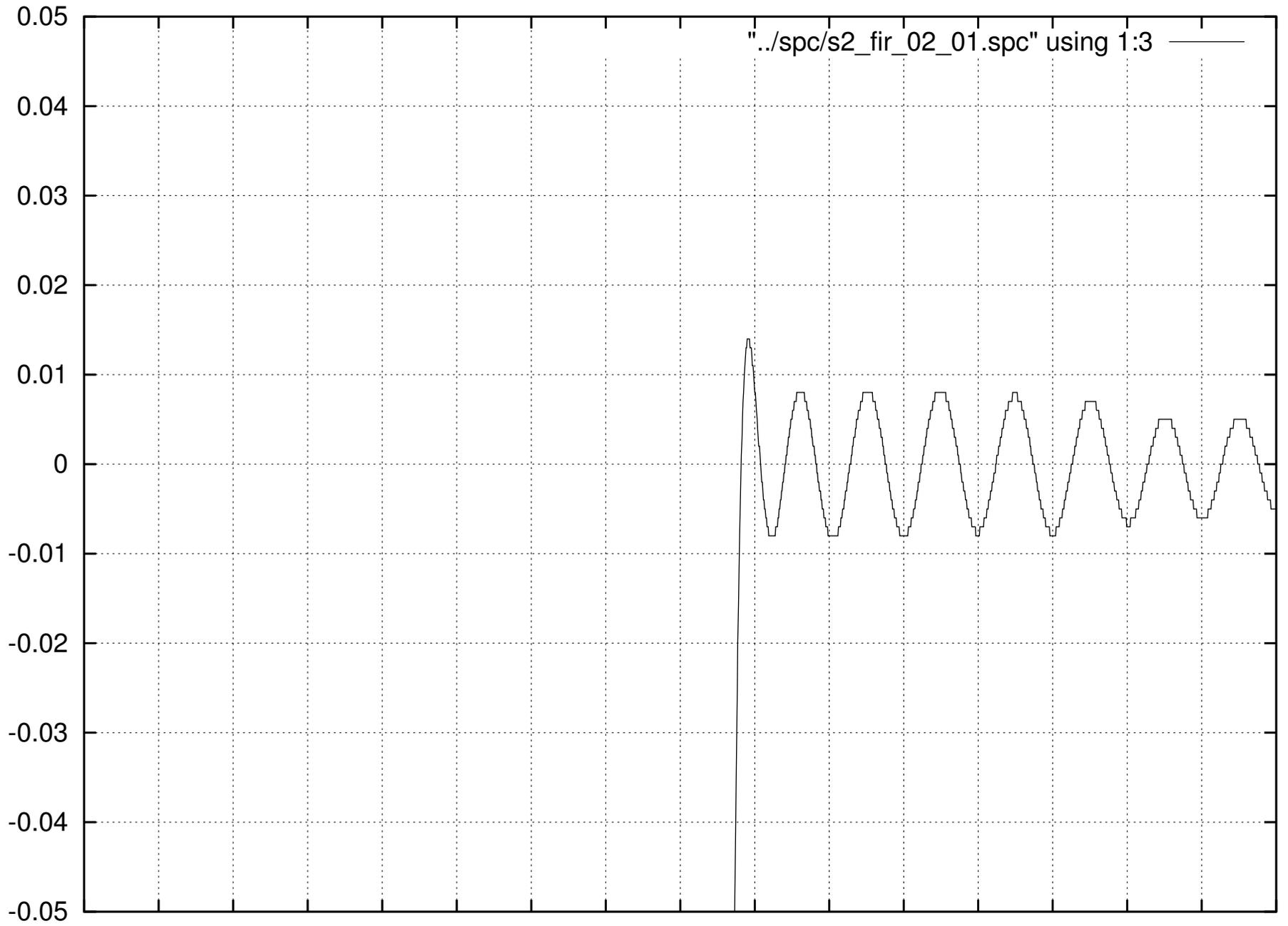




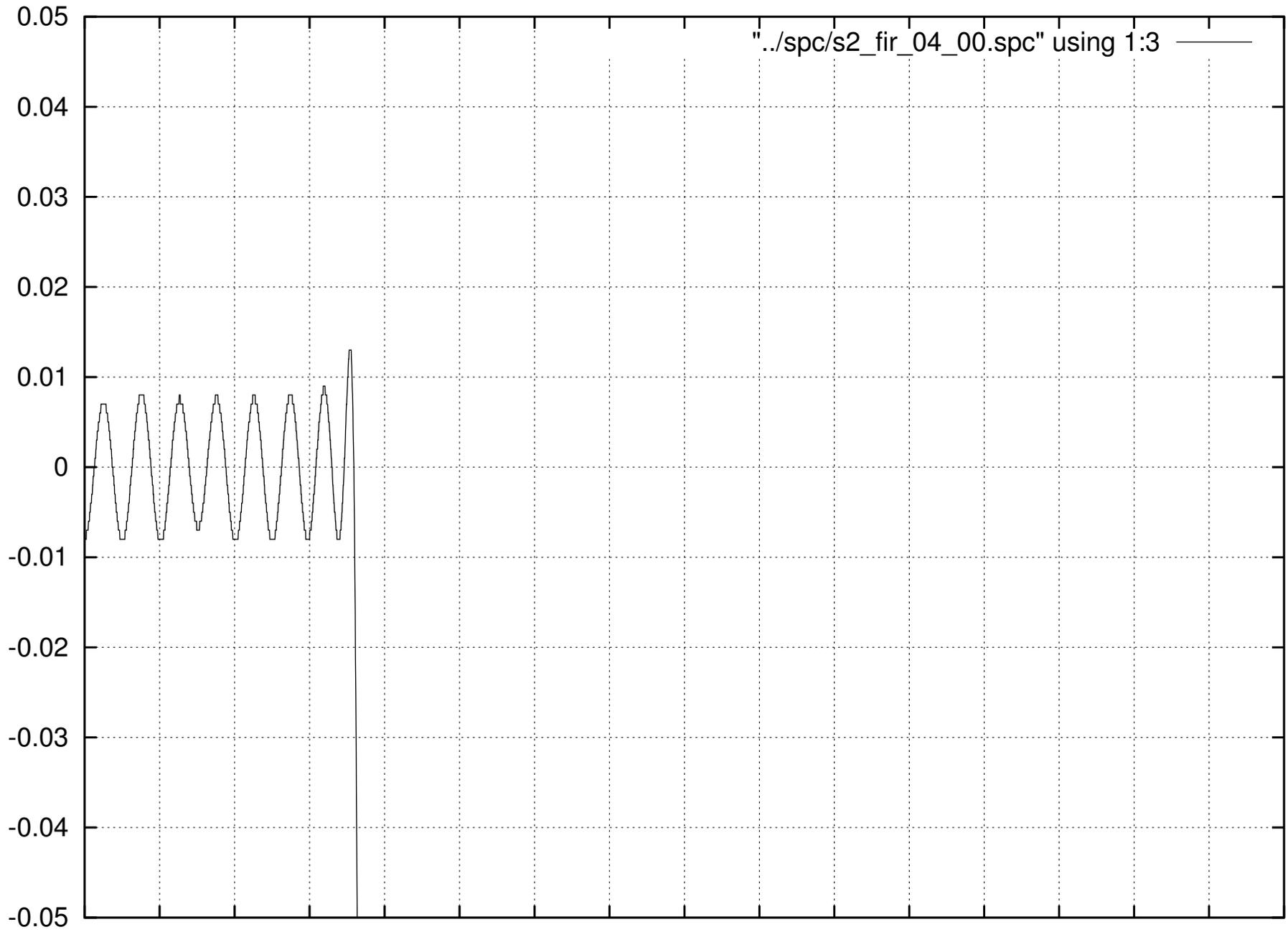


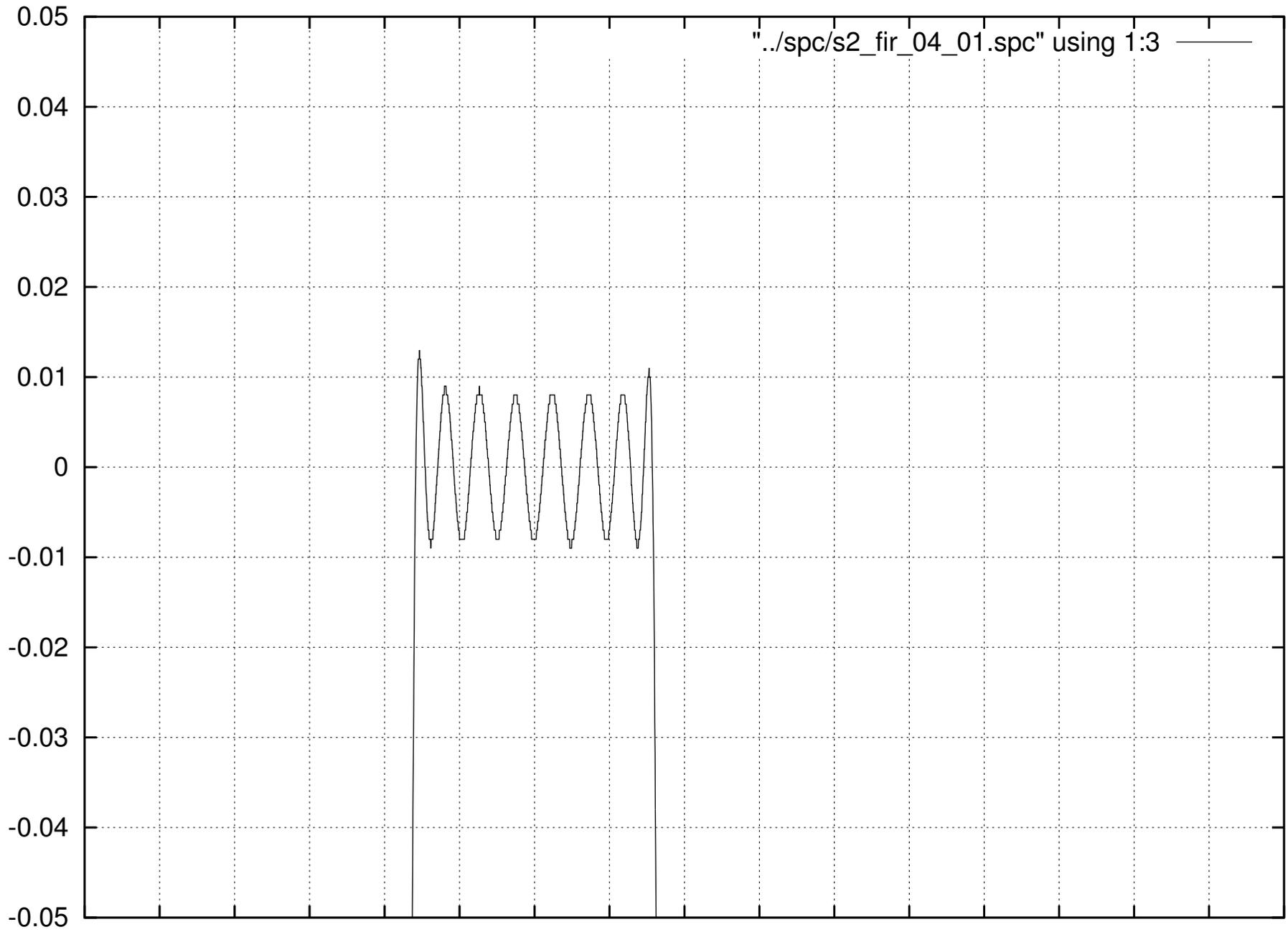


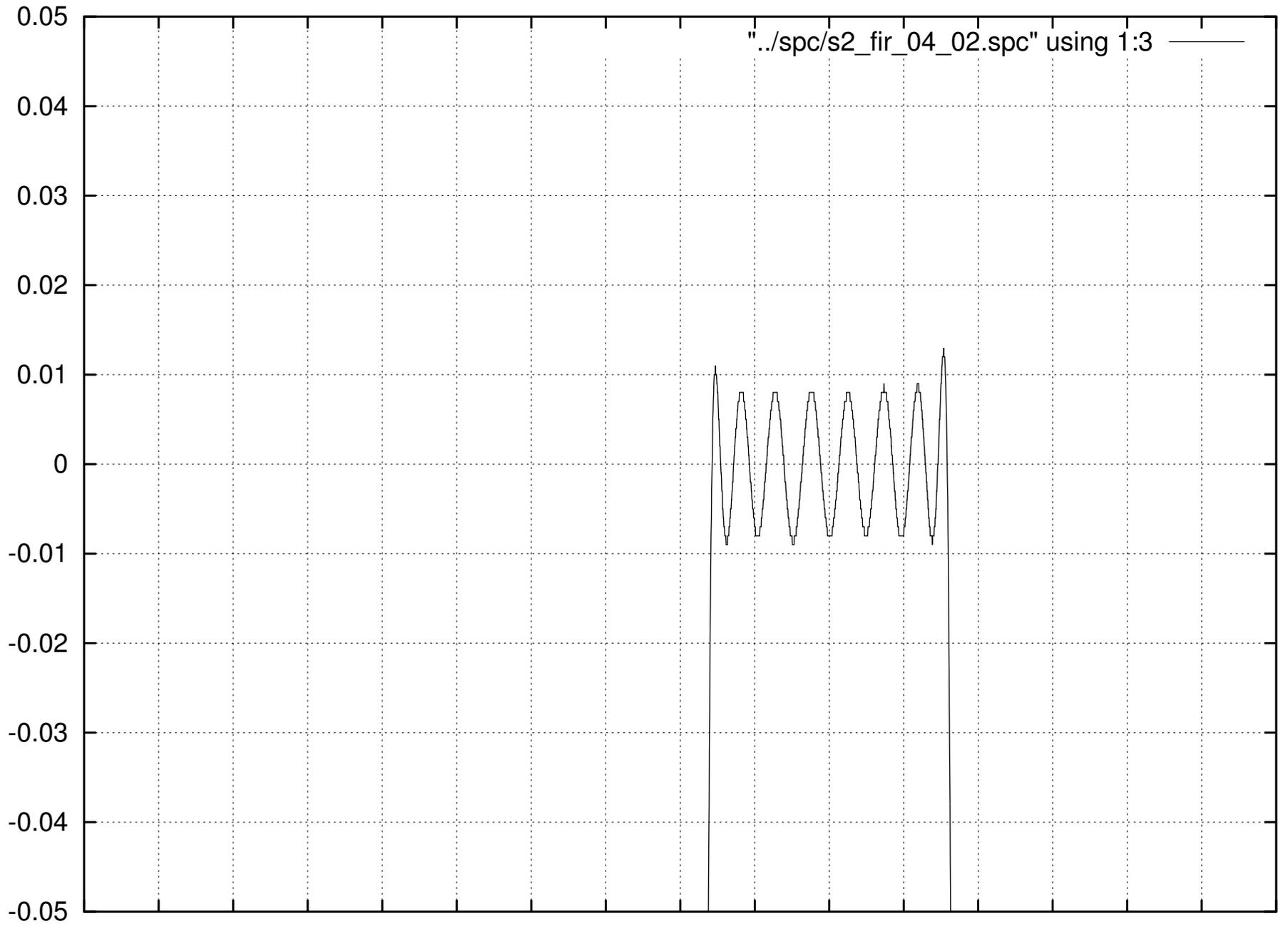


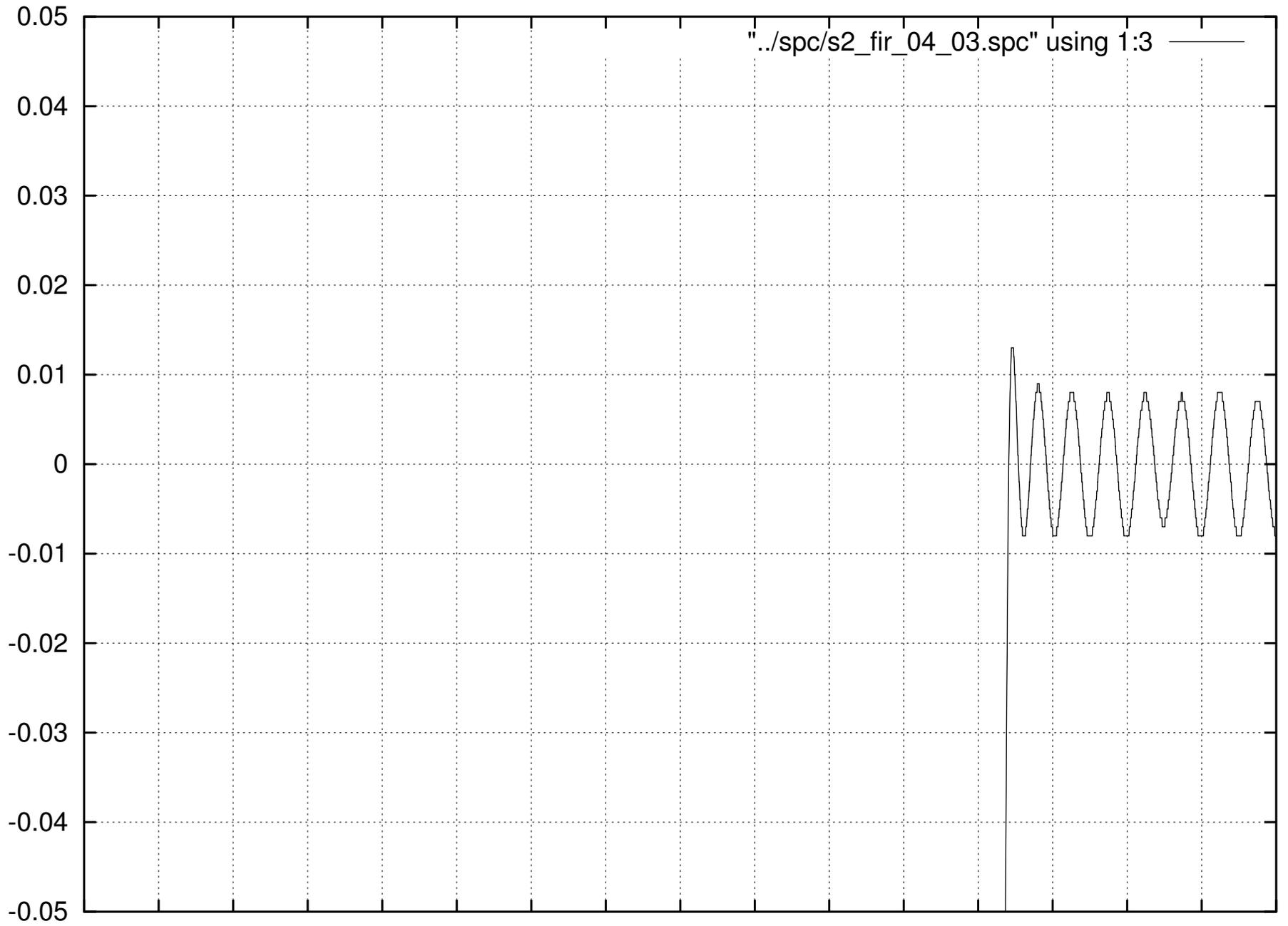


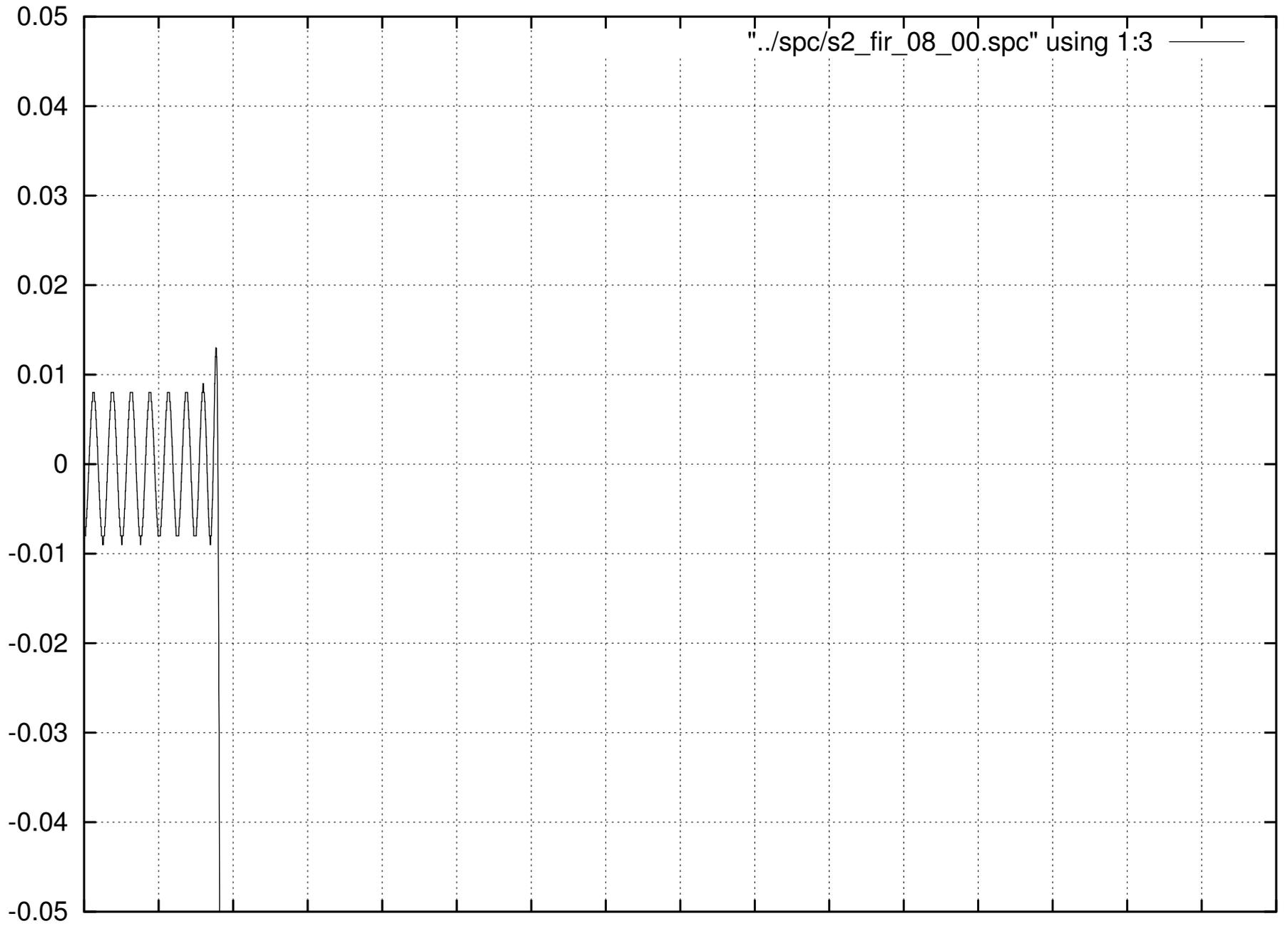
"././s2_fir_02_01.spc" using 1:3

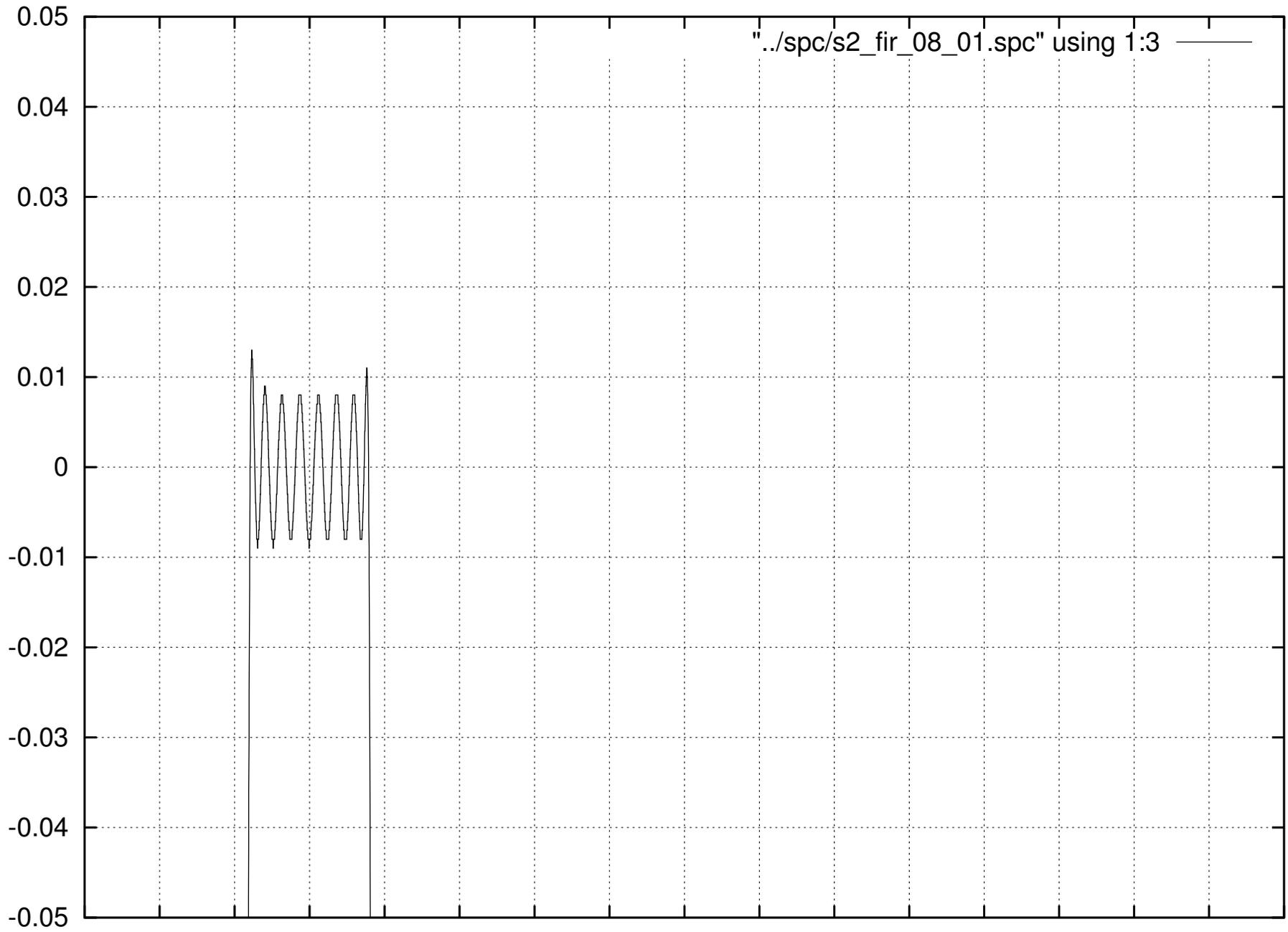


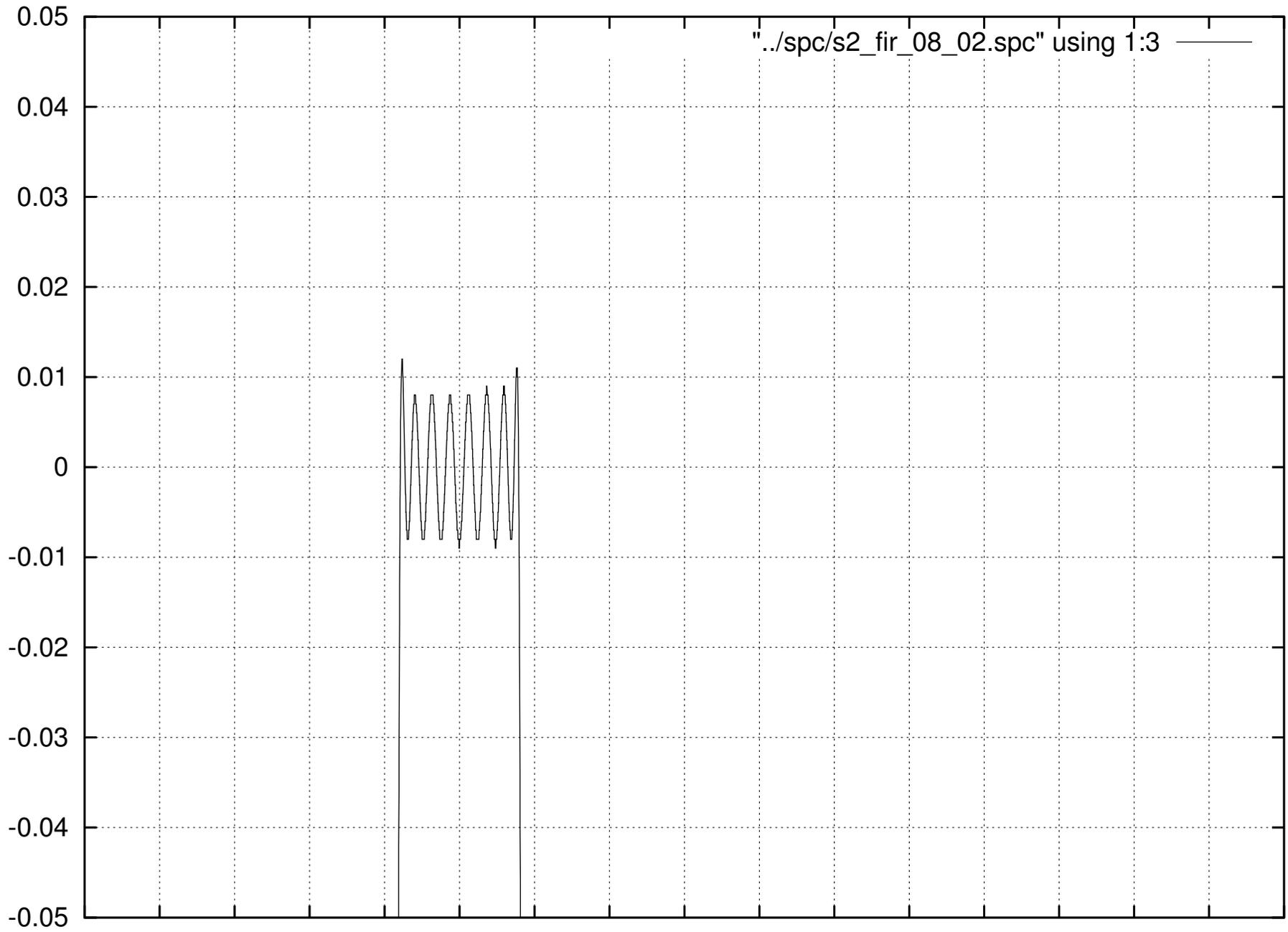


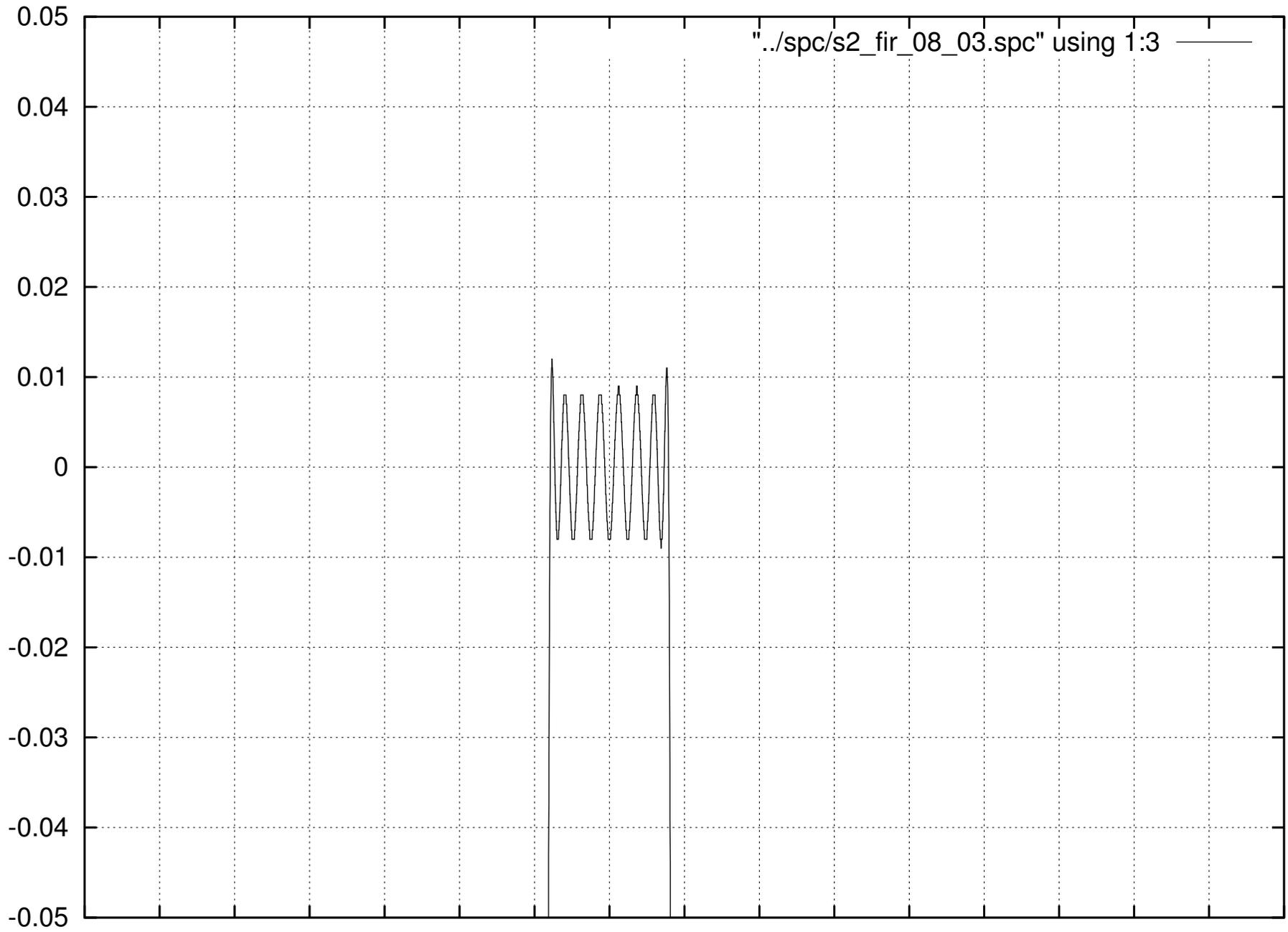


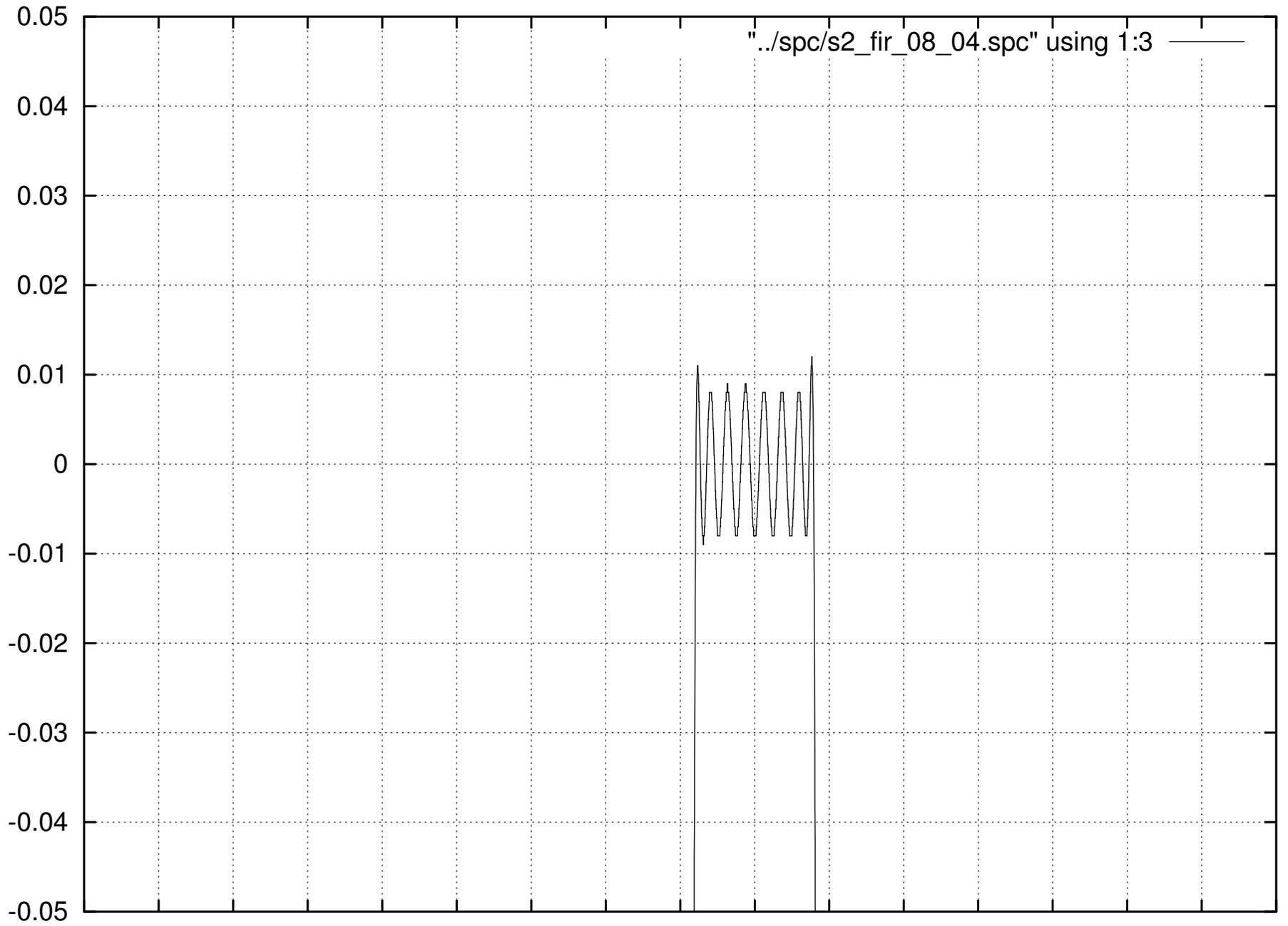


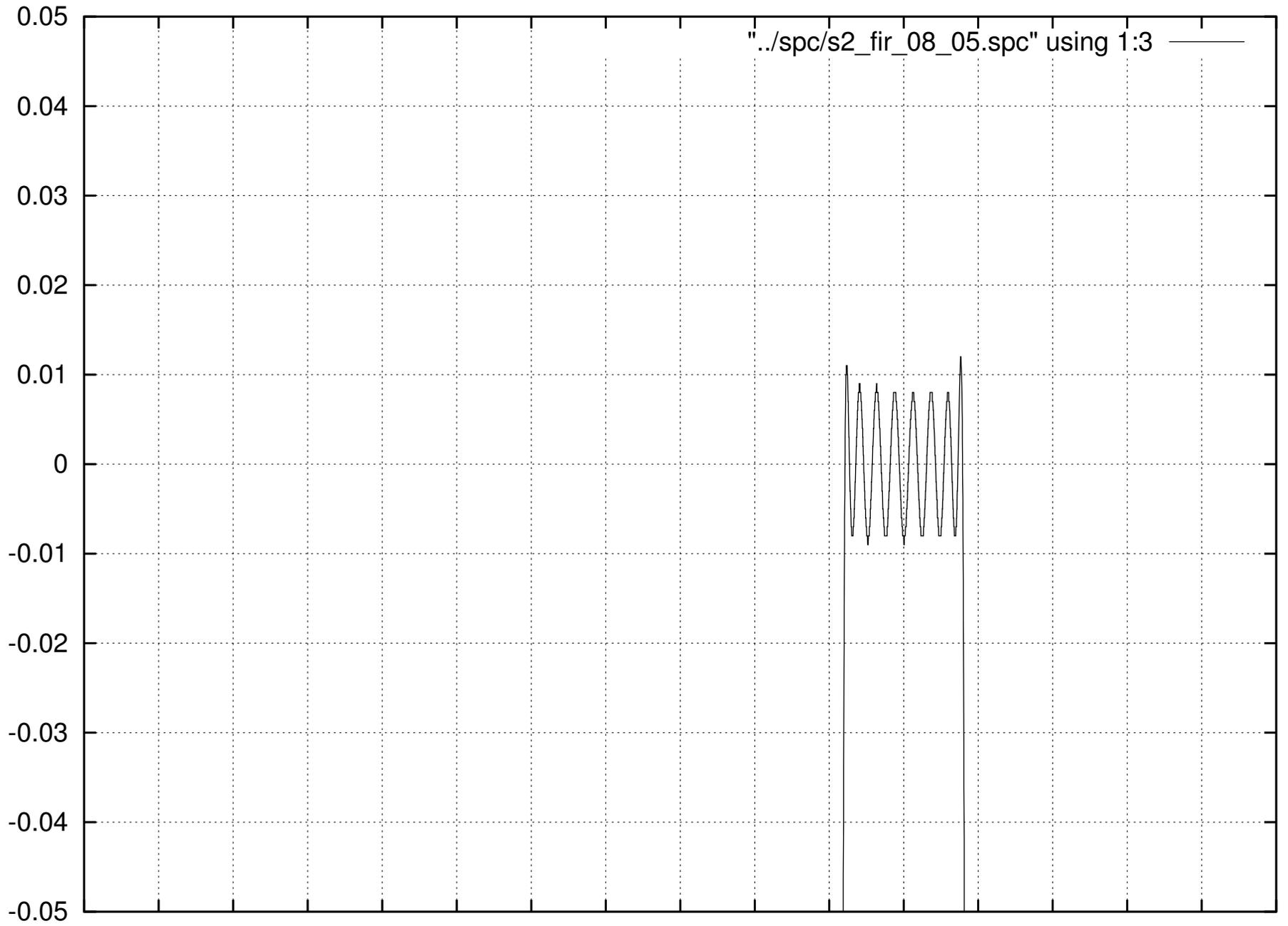


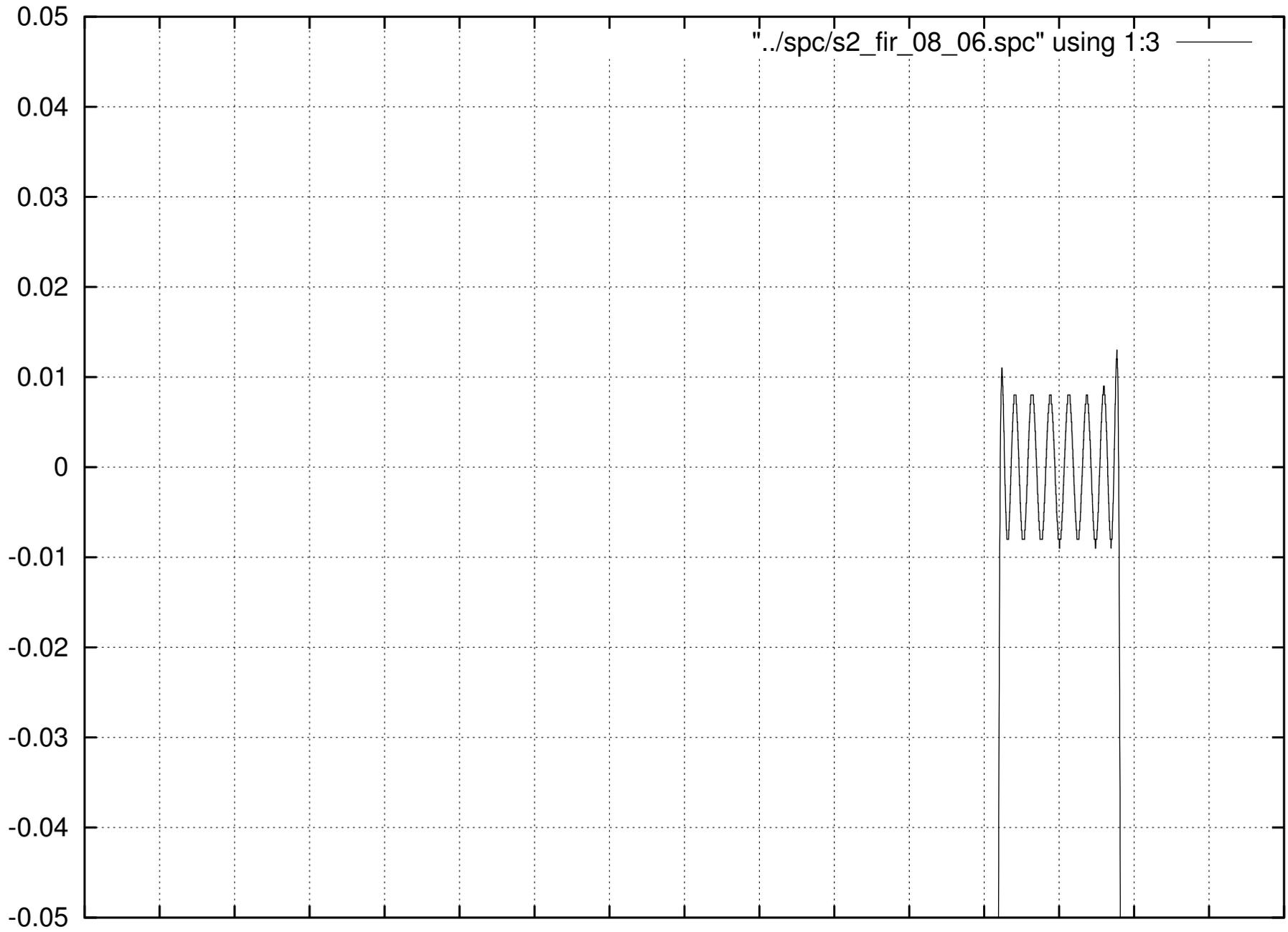


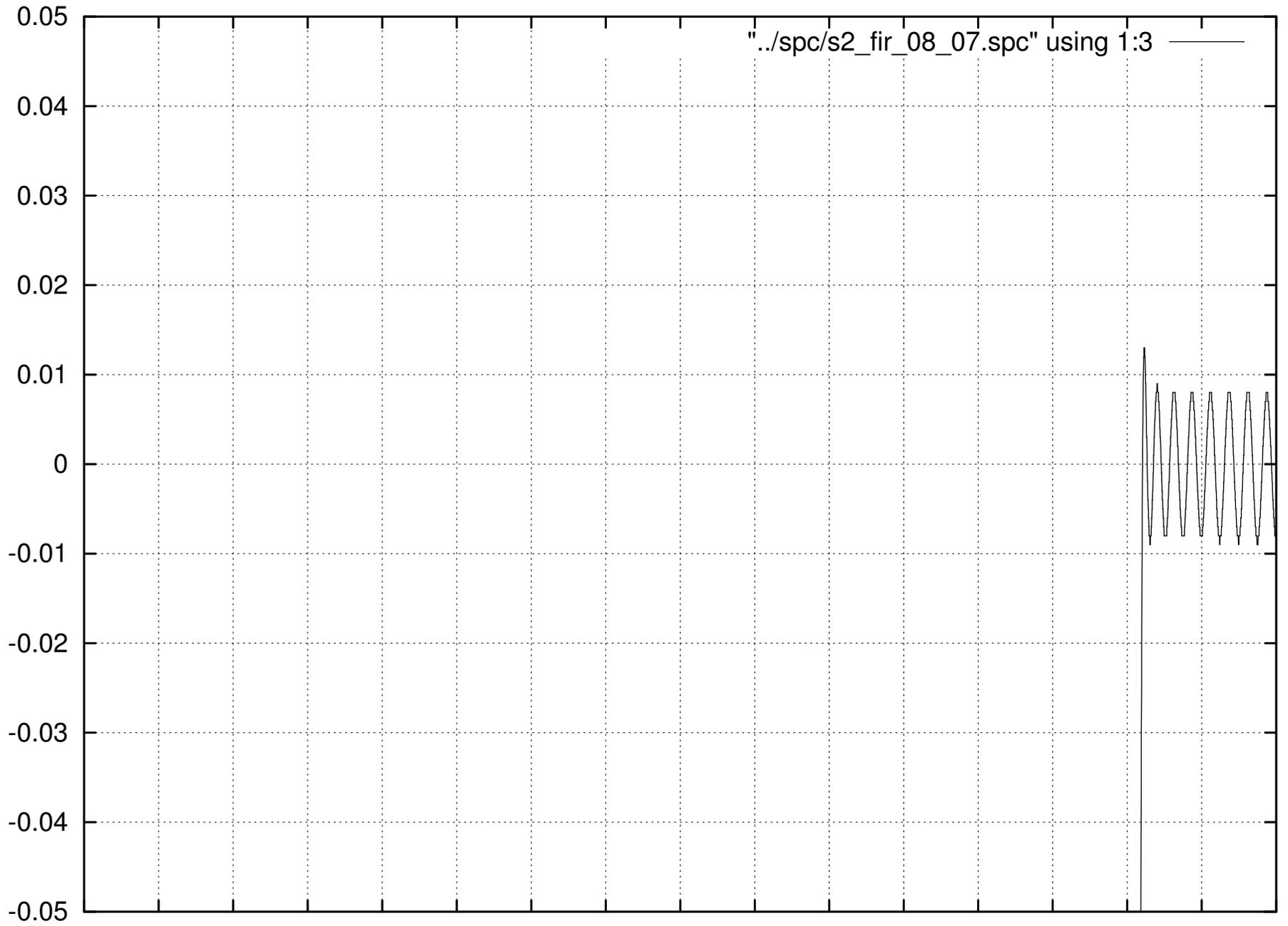


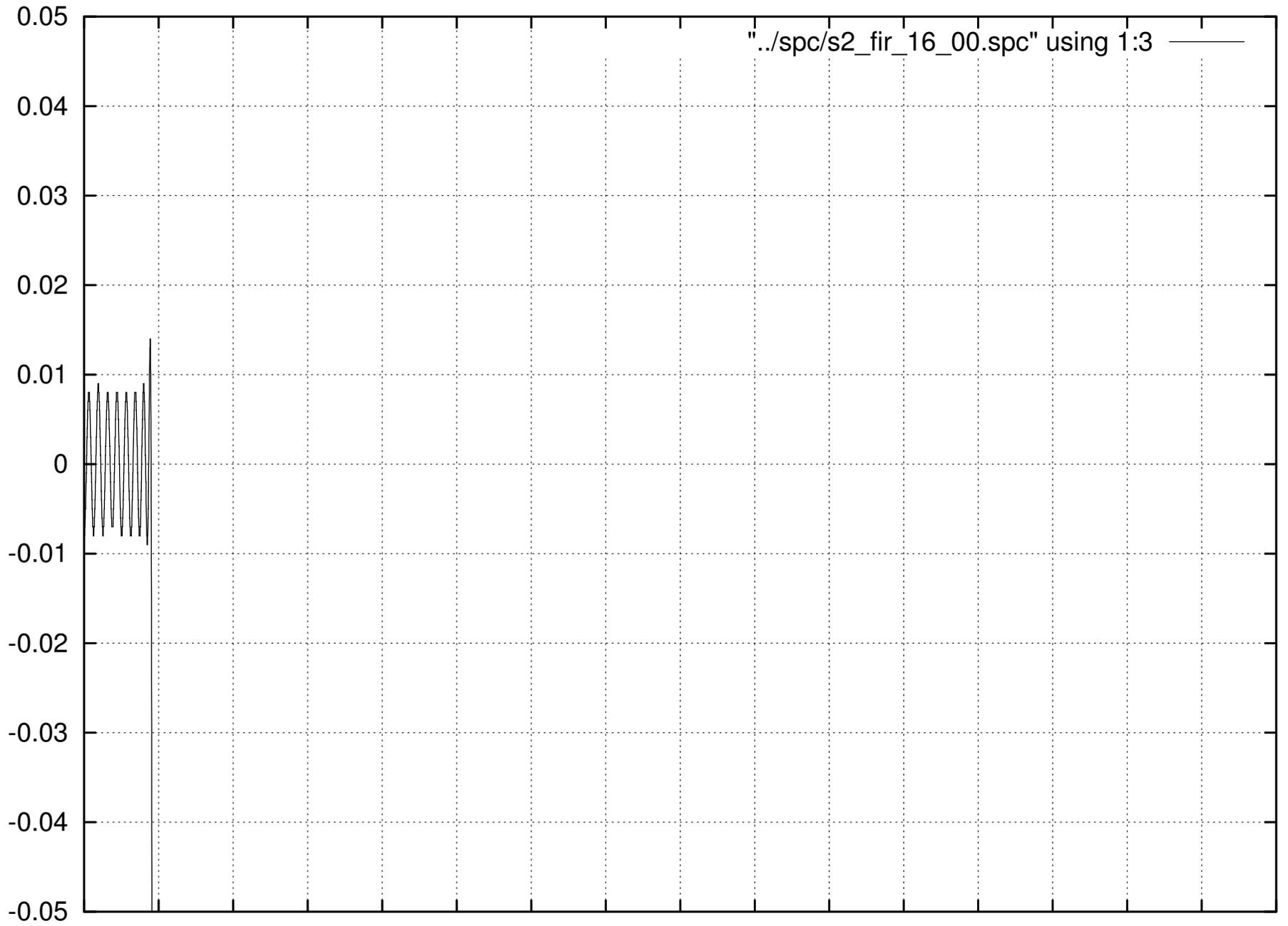


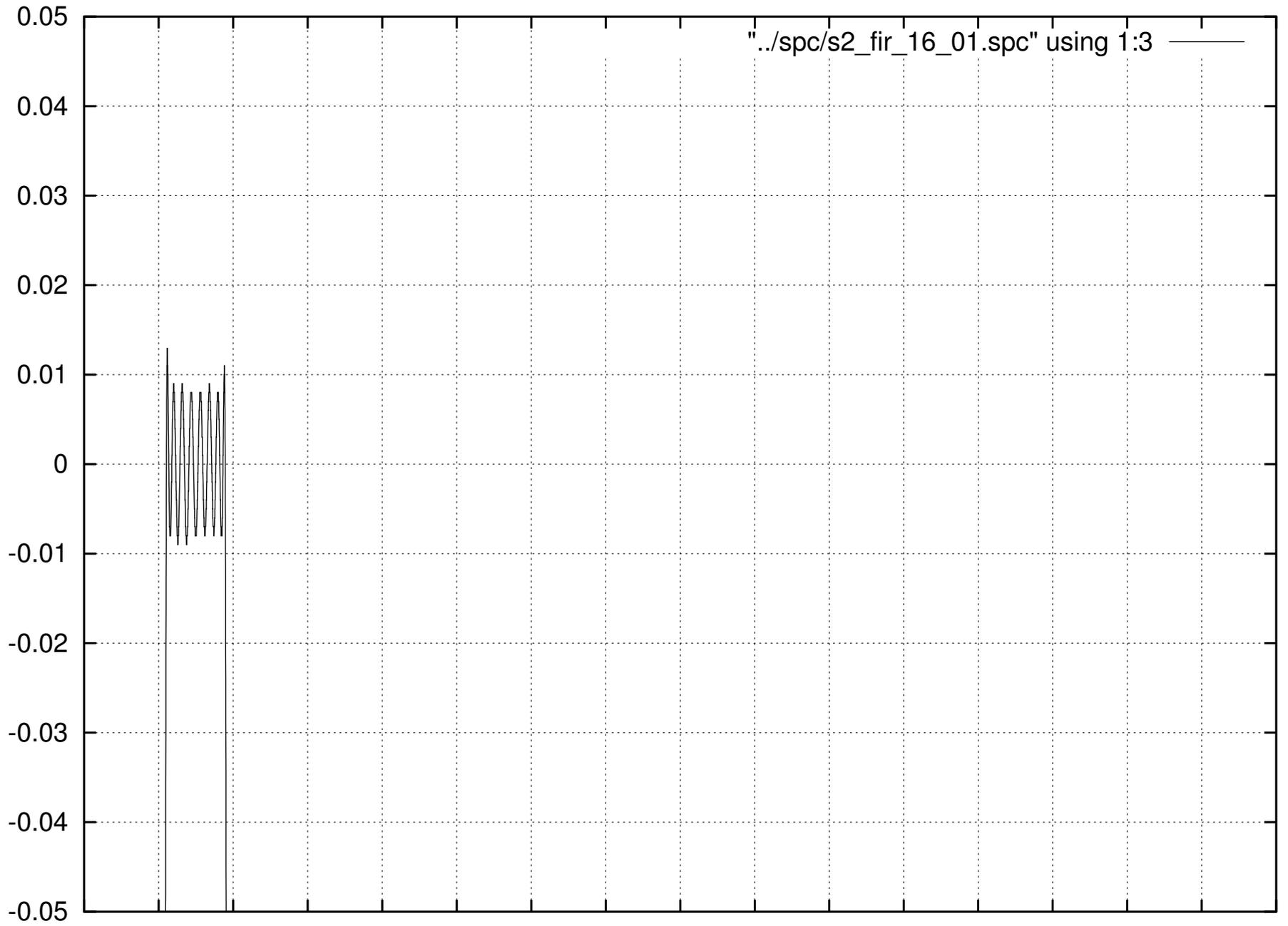


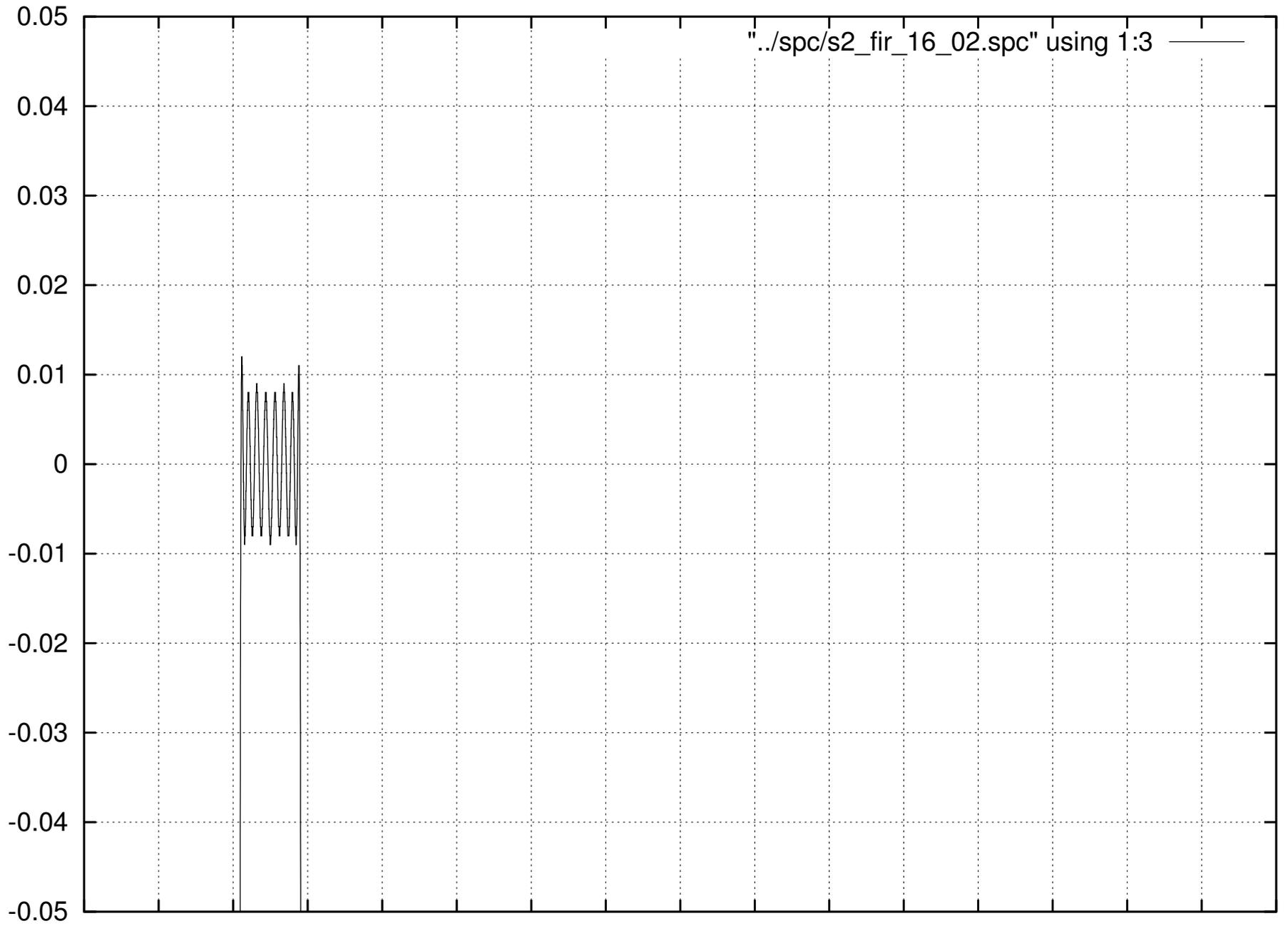


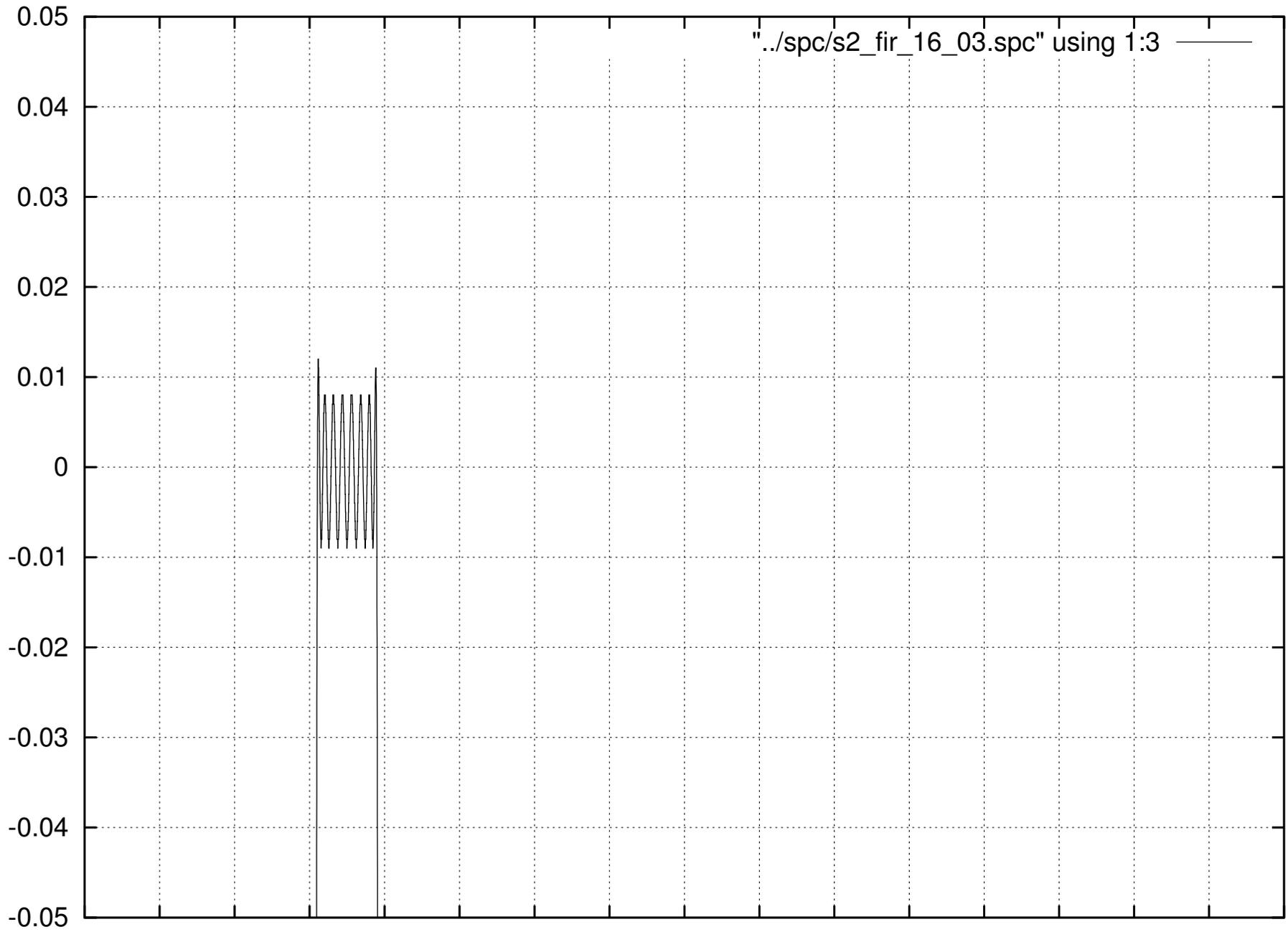


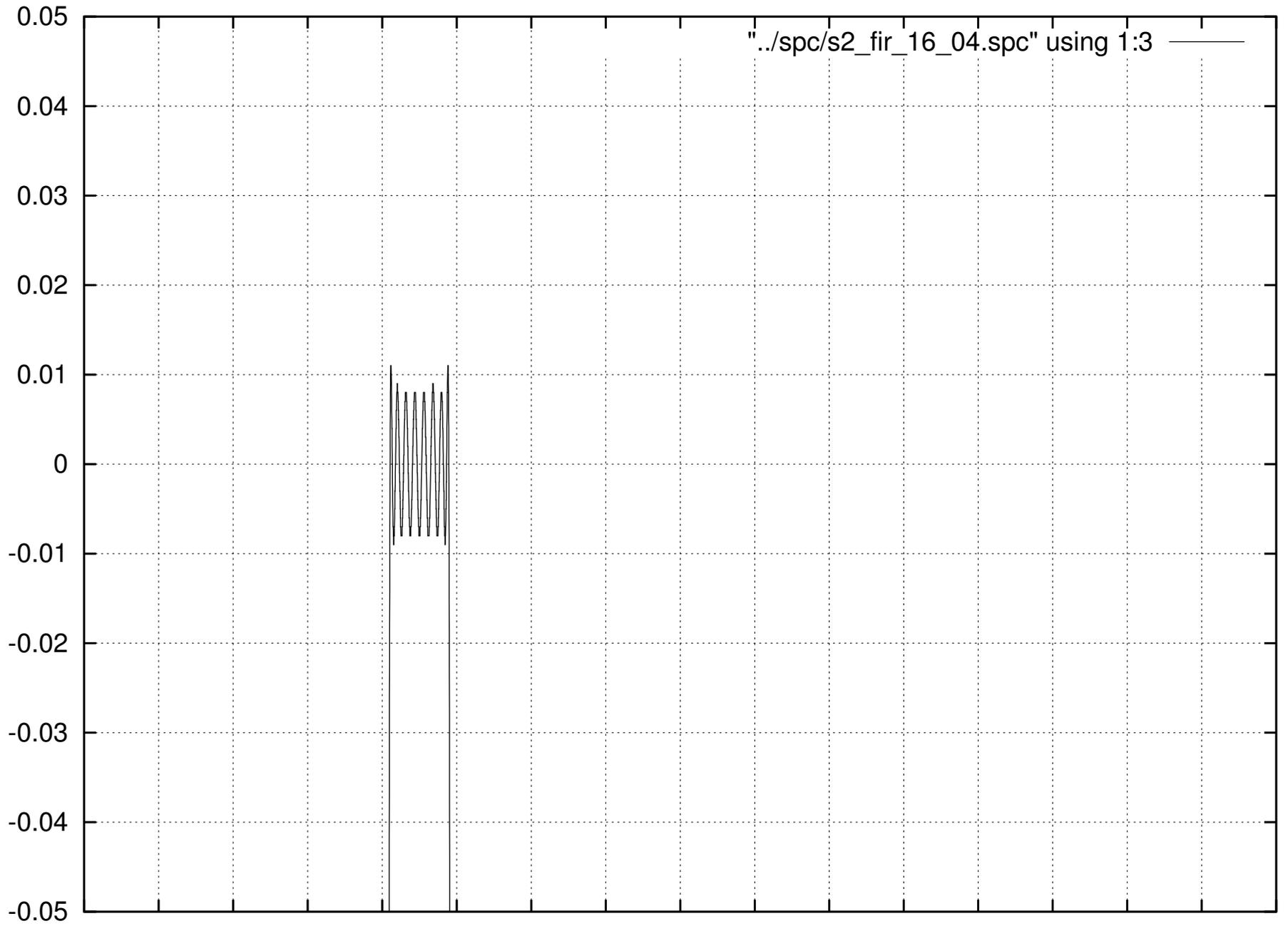


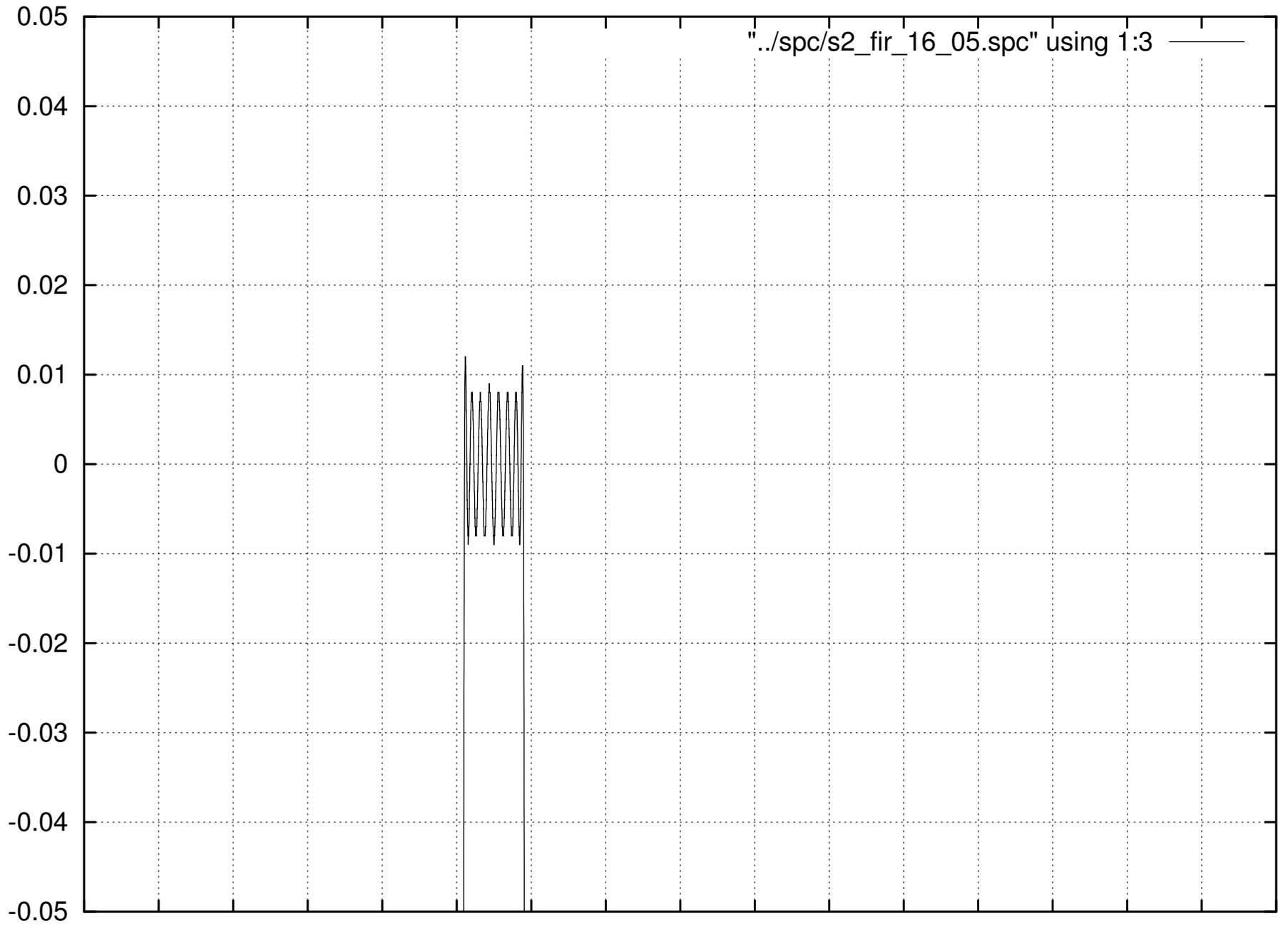


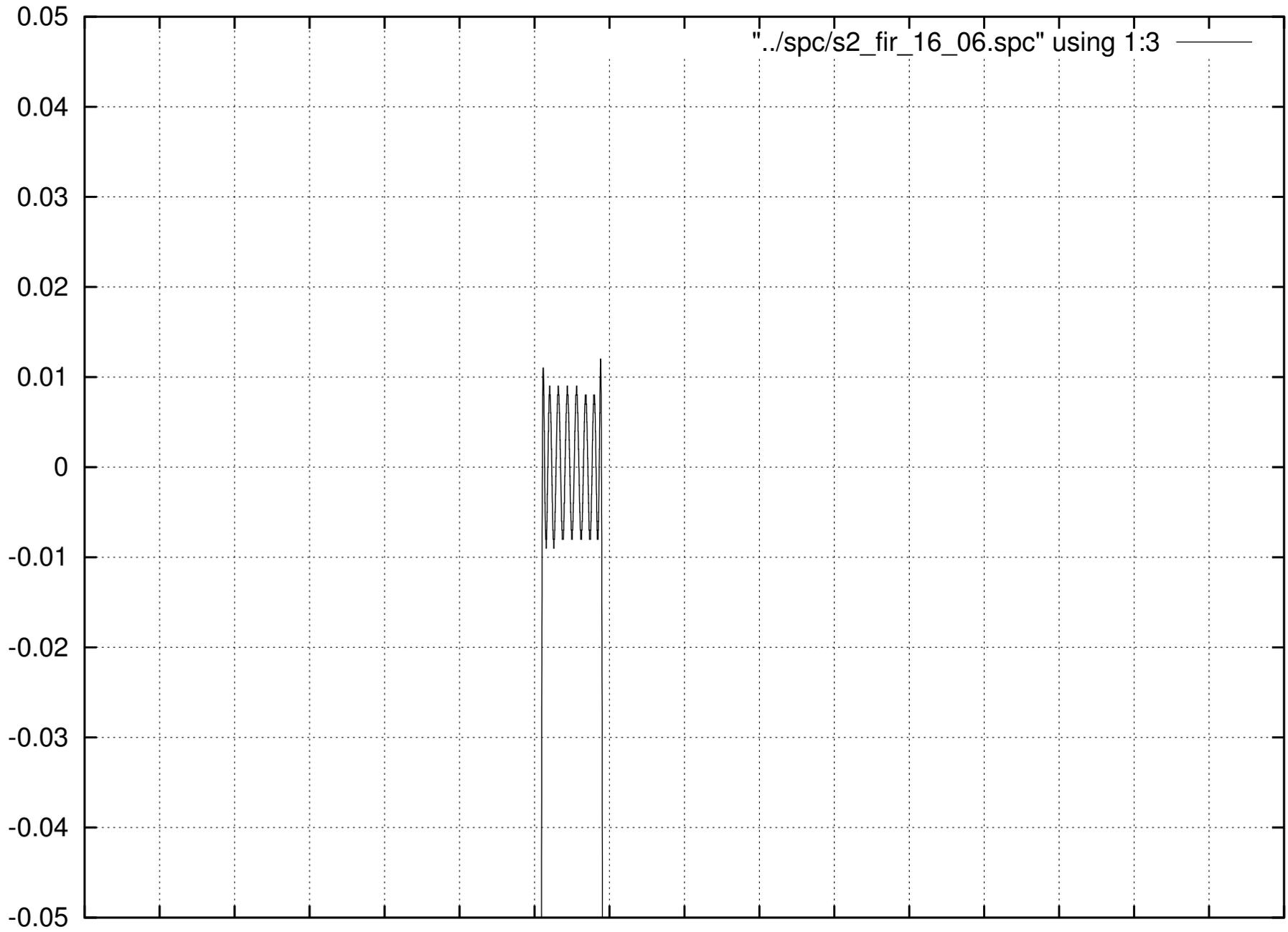


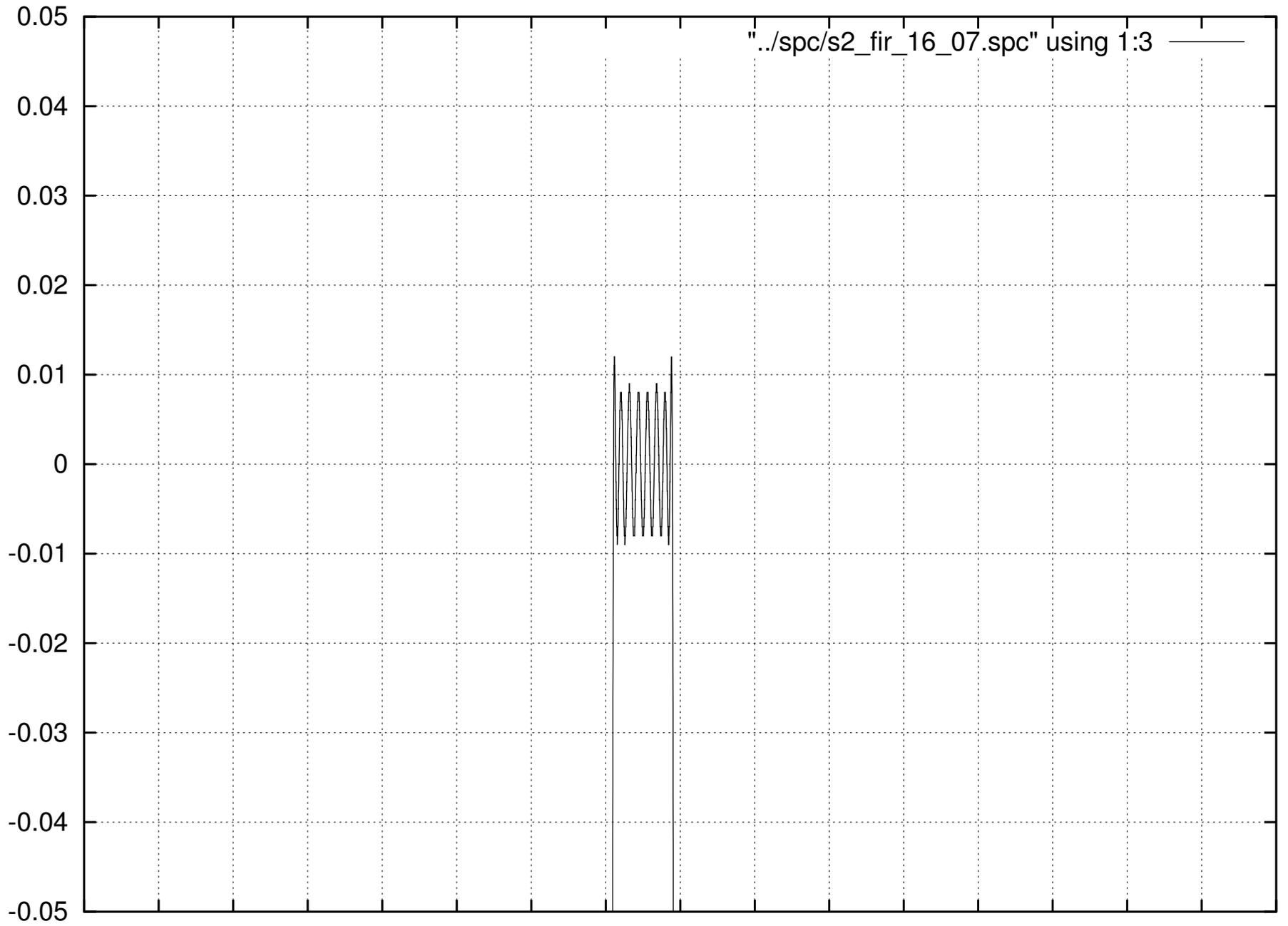


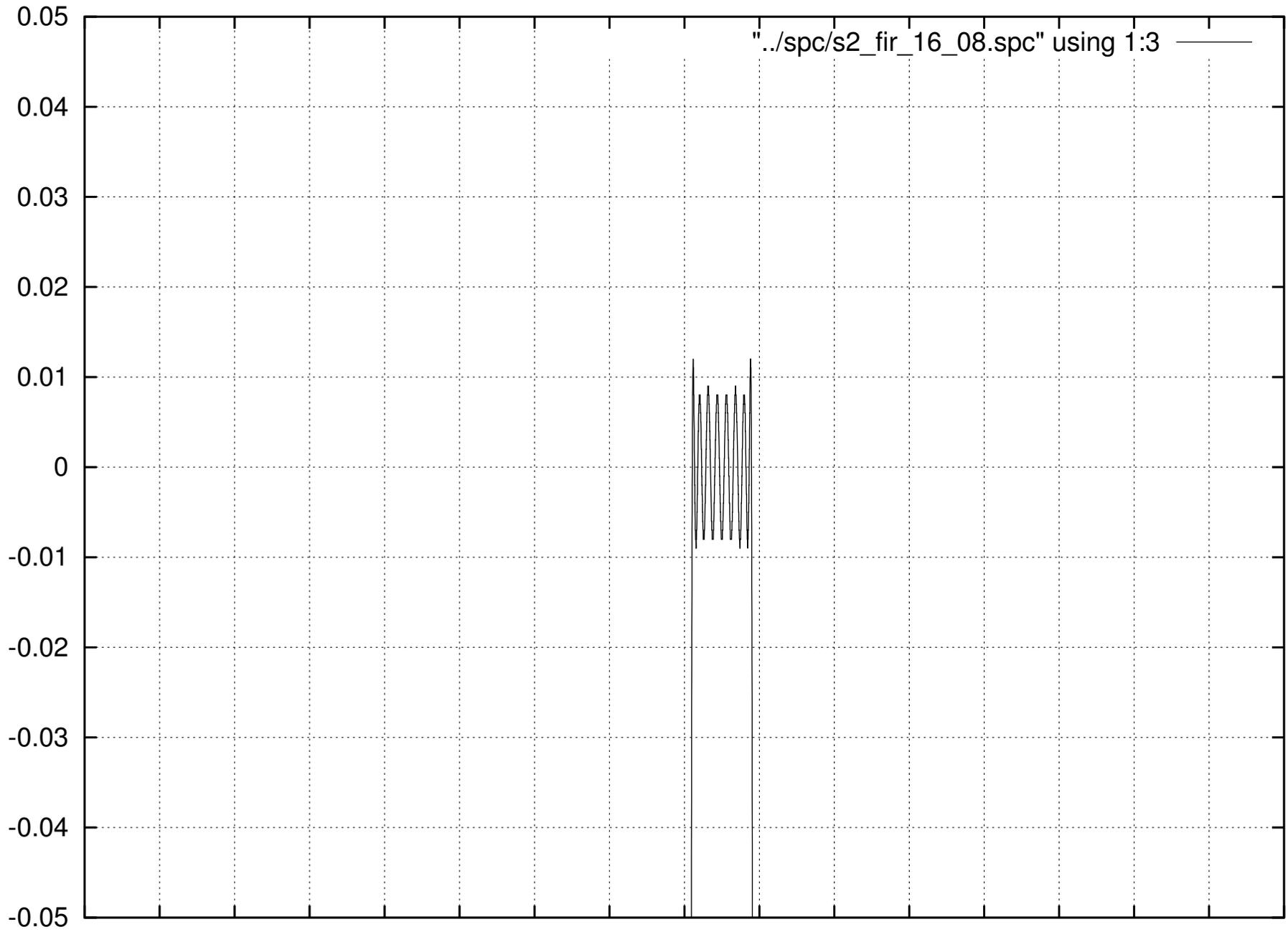


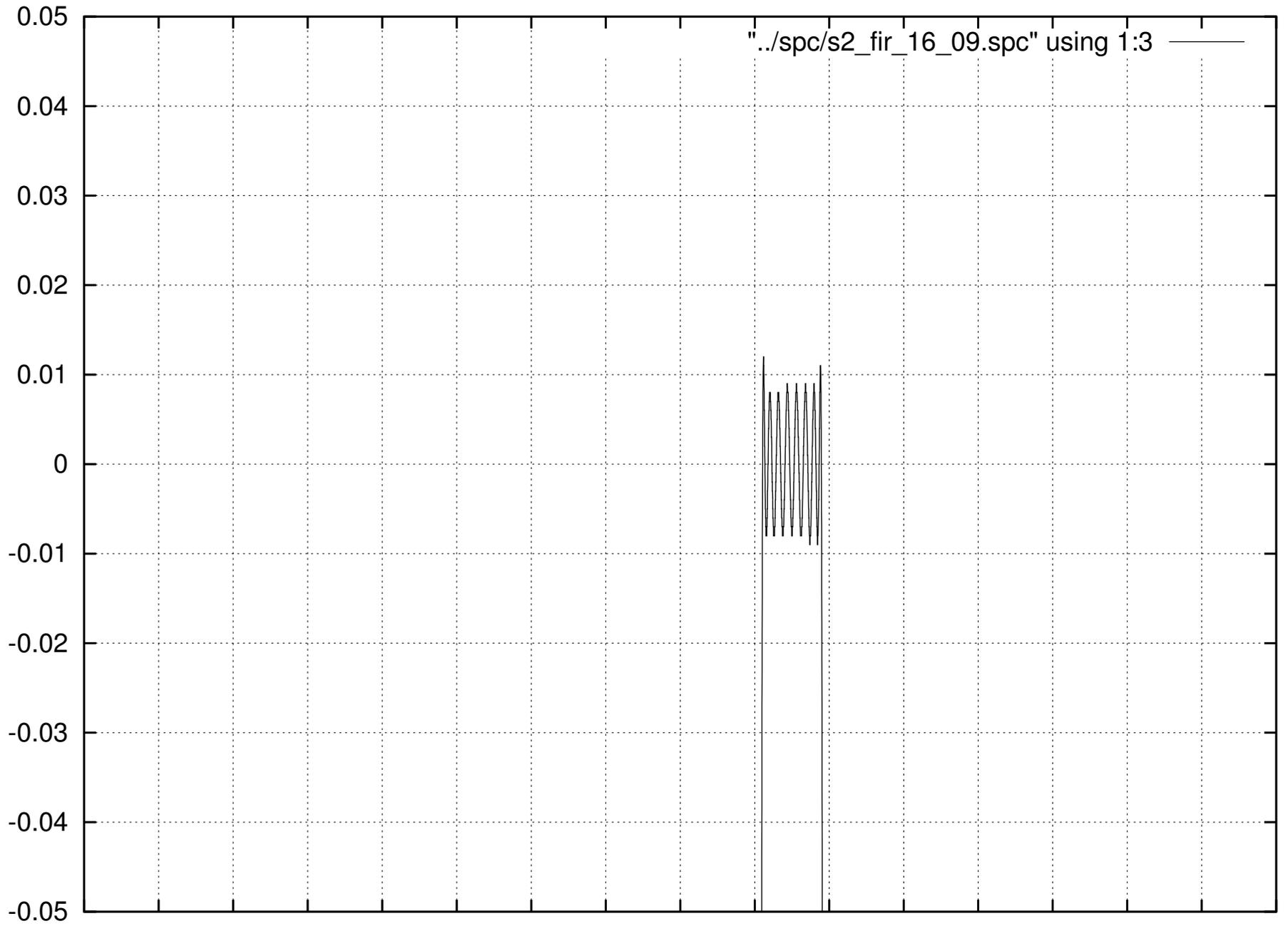


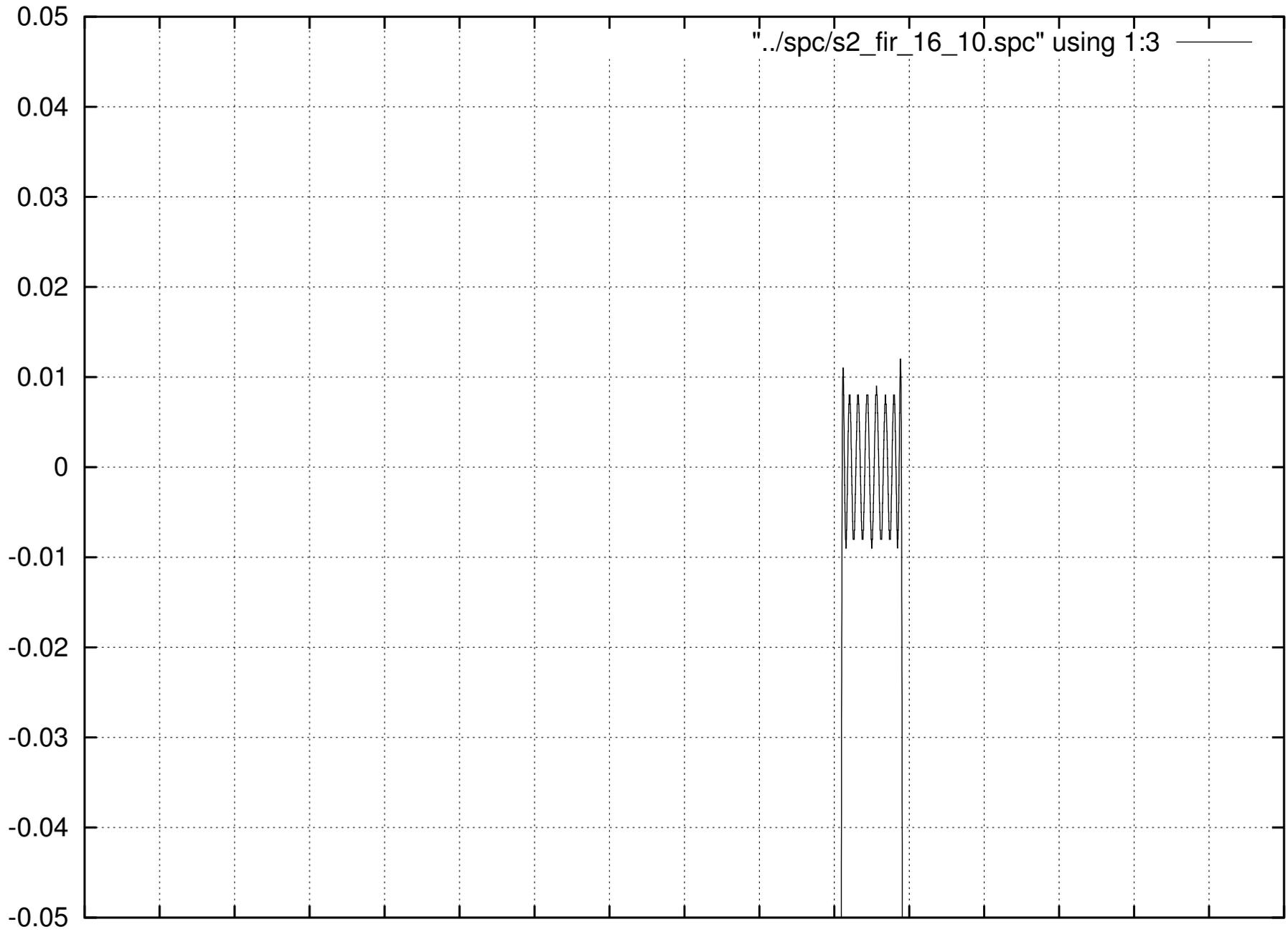


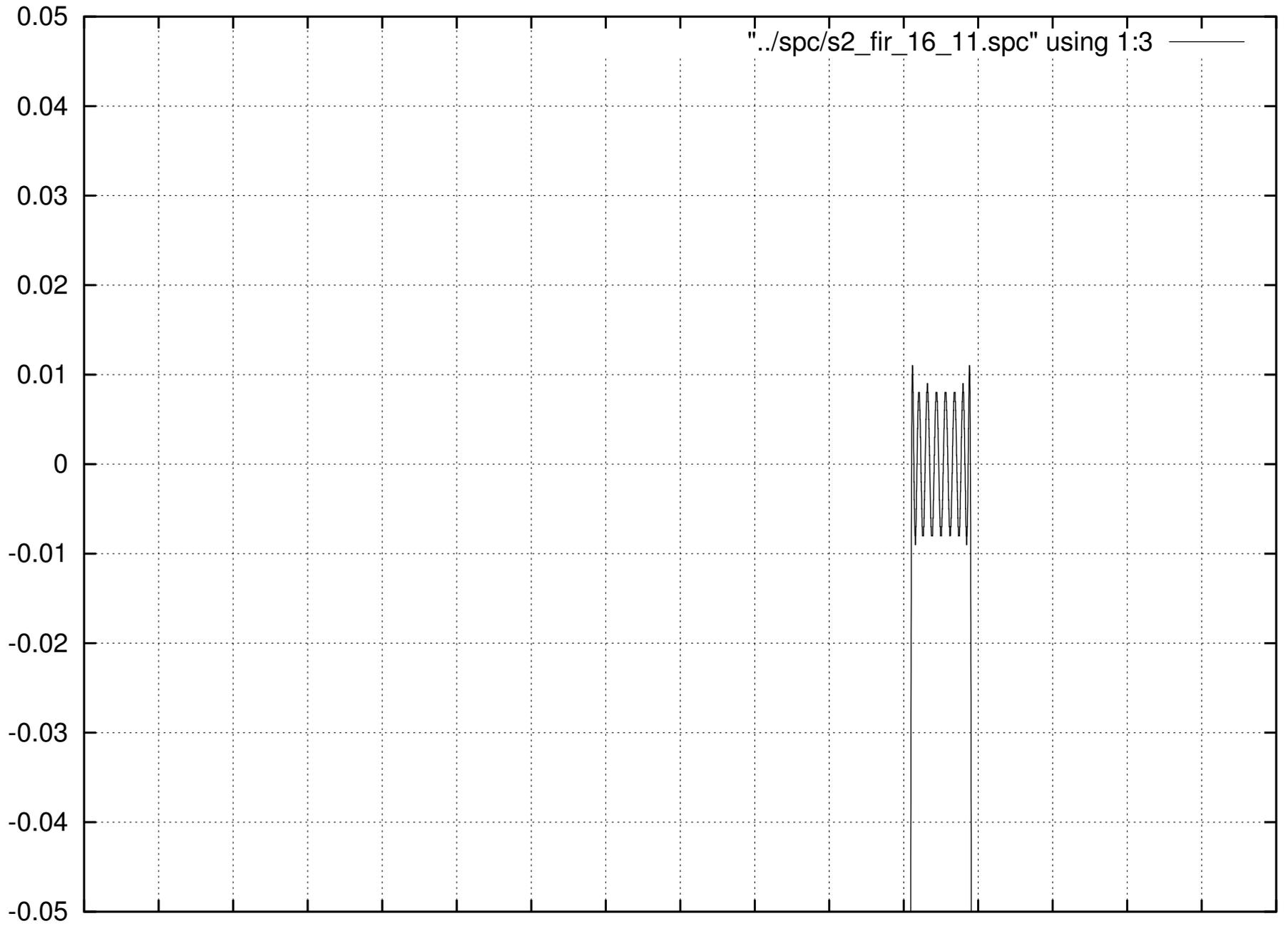


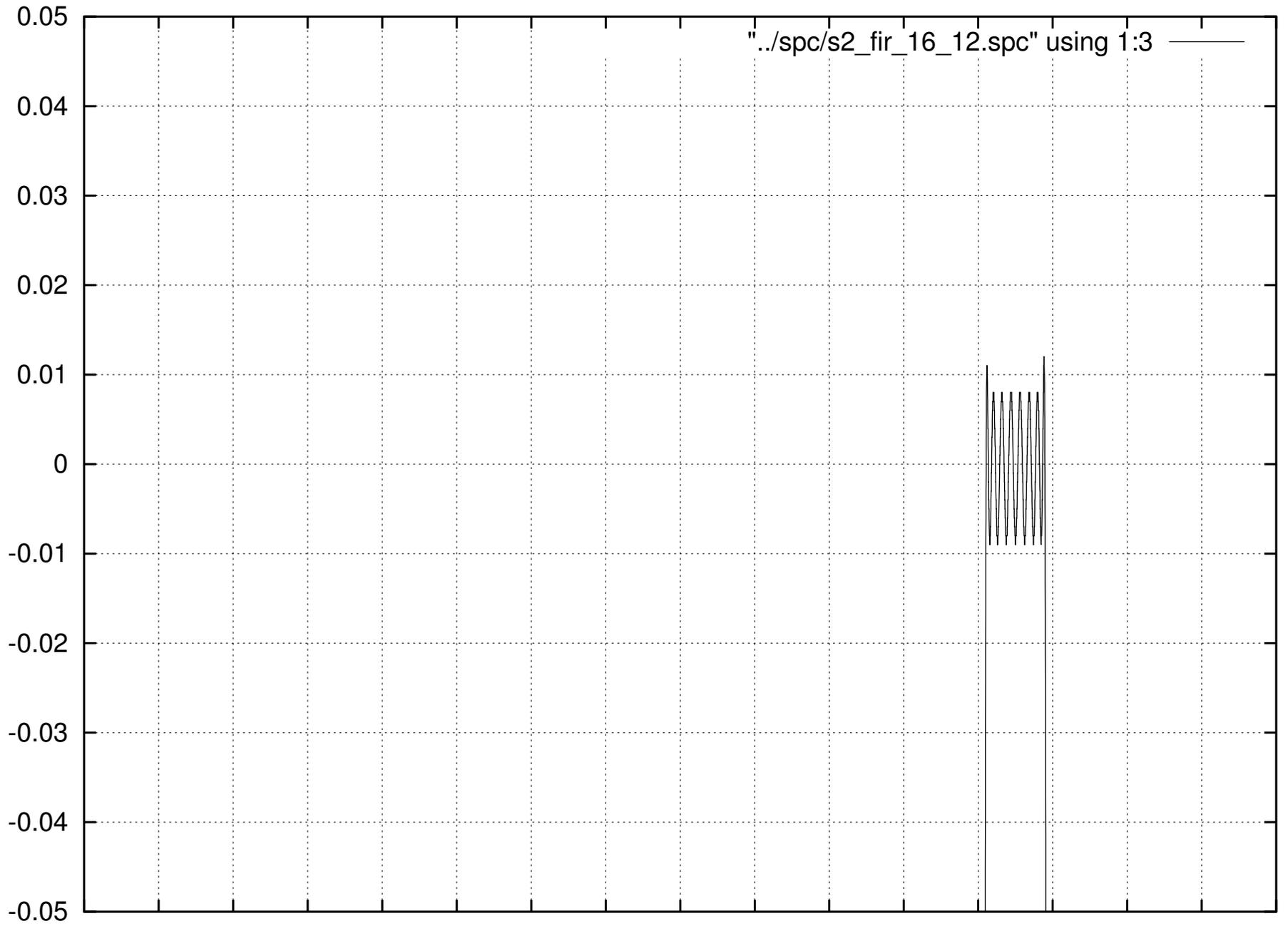


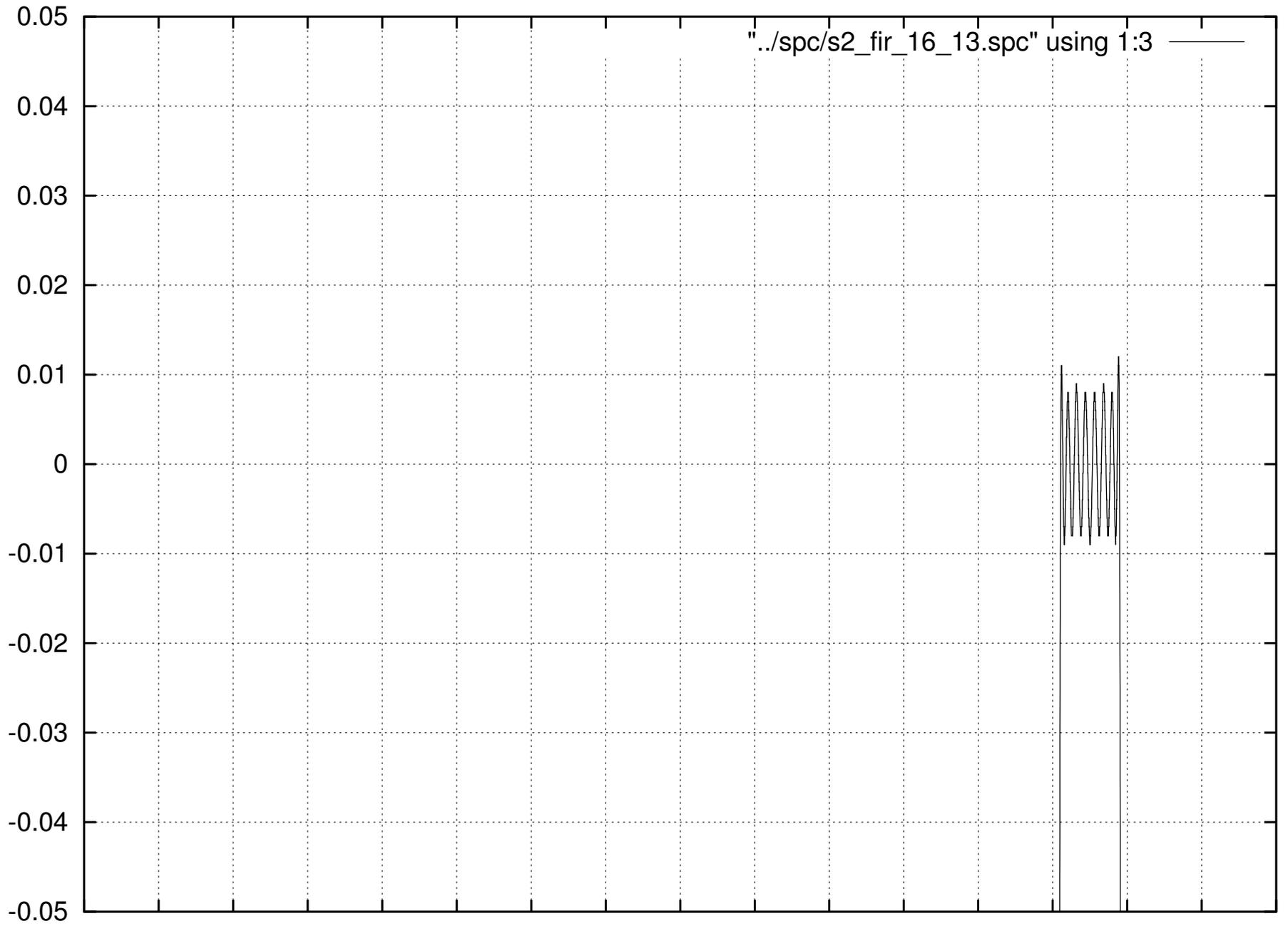


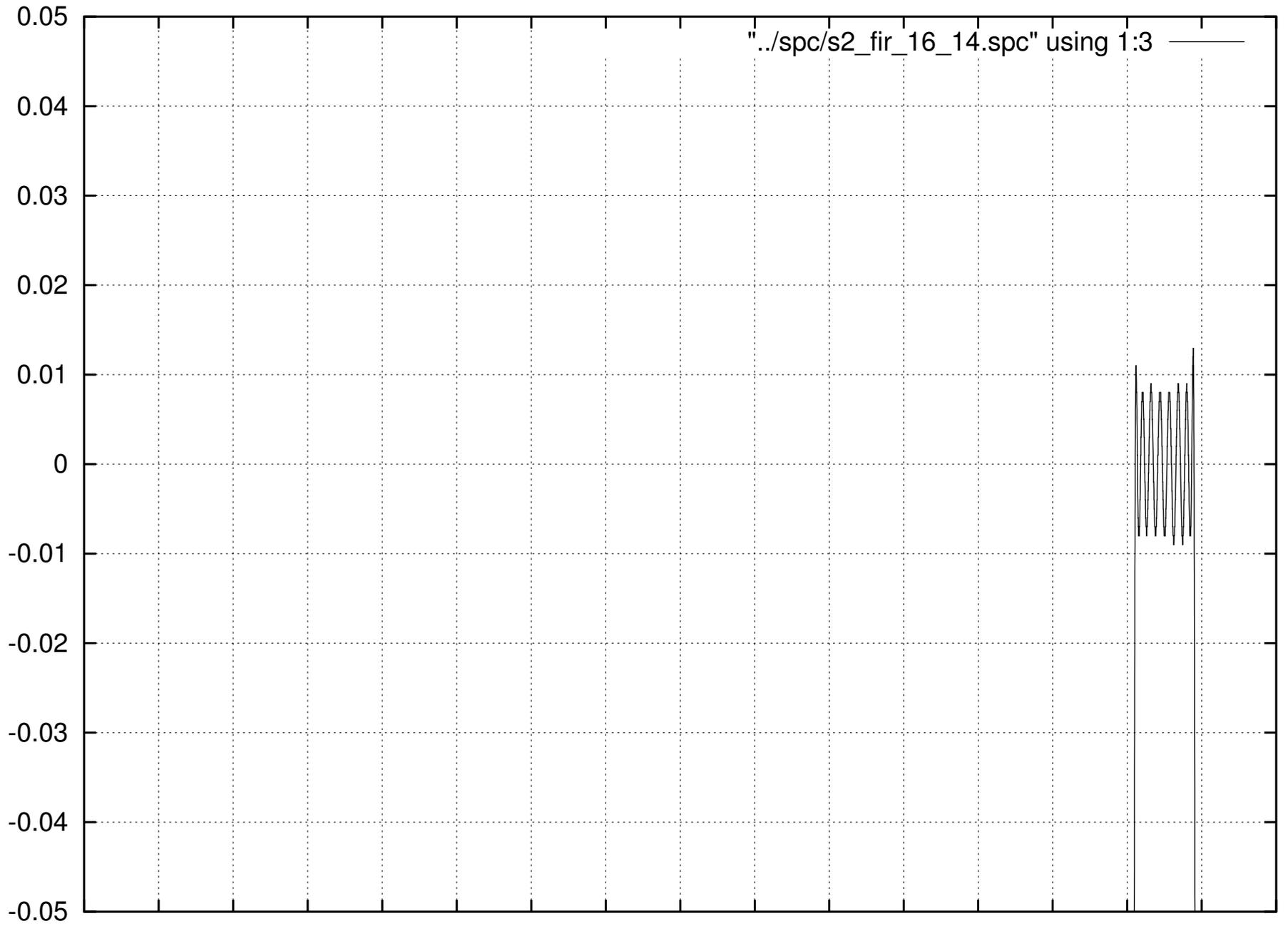


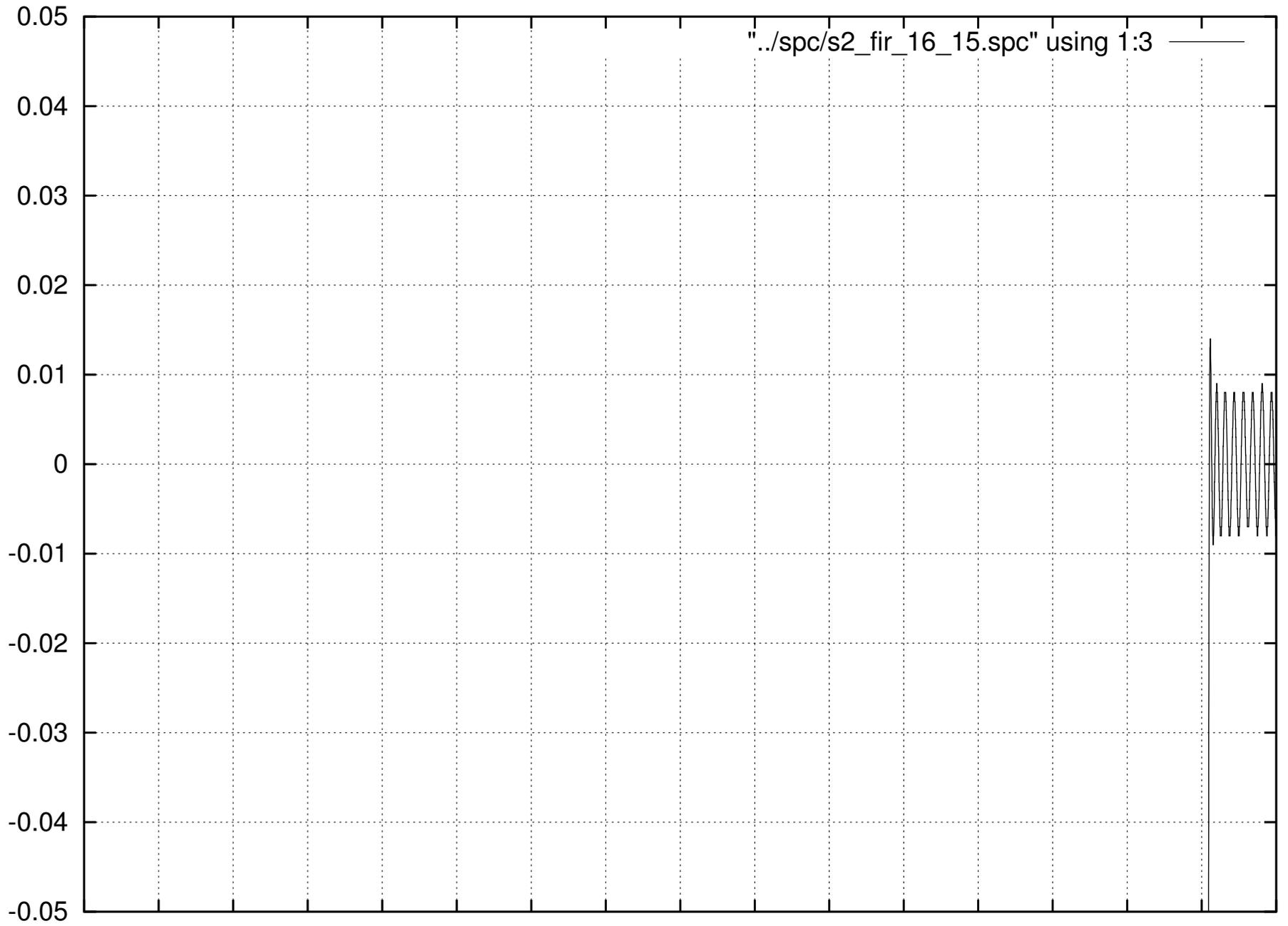


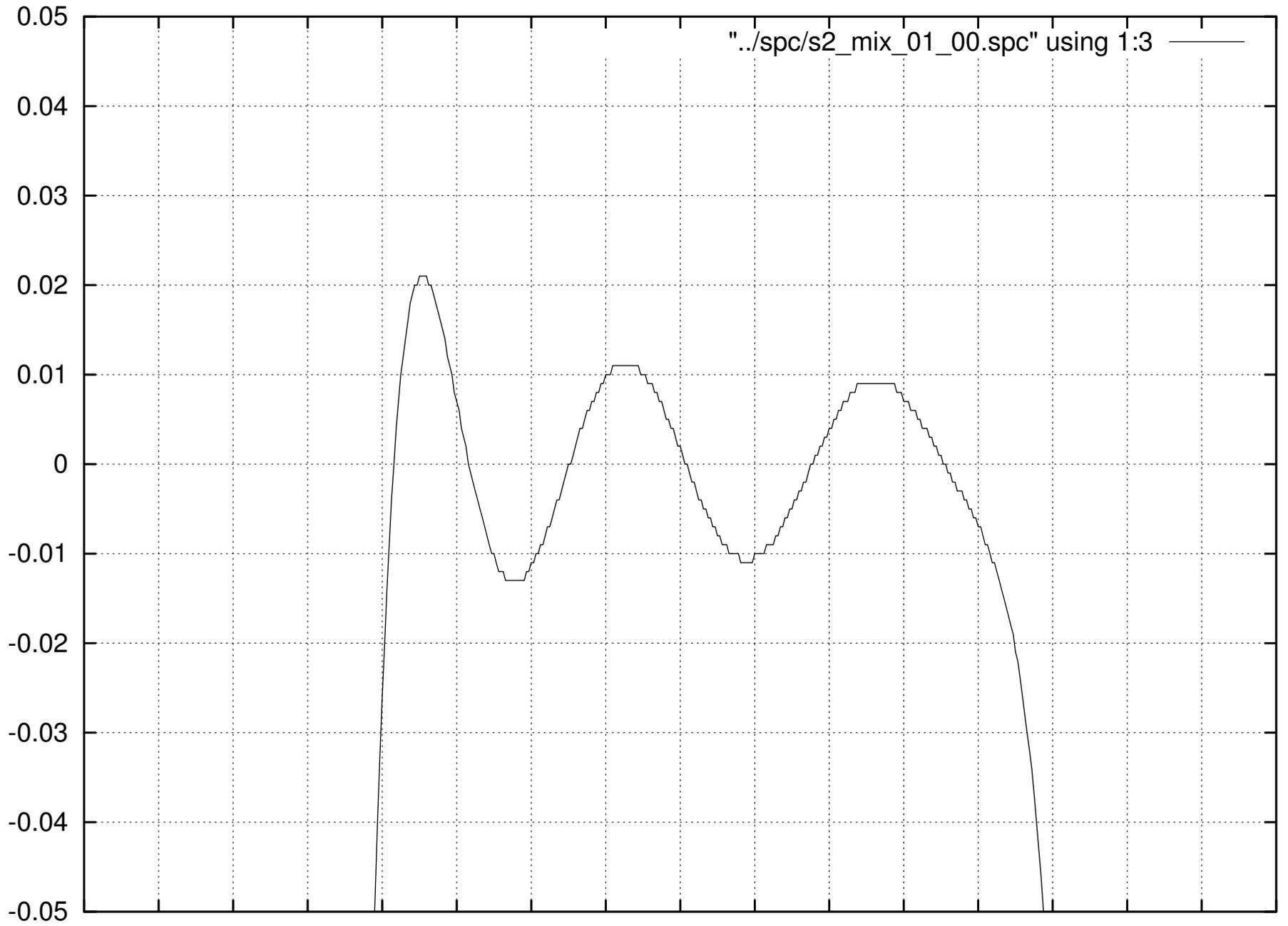












"../spc/s2_mix_01_00.spc" using 1:3

