Virgo Spectrogram tools

D. Verkindt

VIR-0012A-19

https://tds.ego-gw.it/ql/?c=13835

11 January 2019

Change Record

Version	Date	Section Affected	Reason / Remarks
v1	08/01/2019	All	First version
v2	11/01/2019	5	Added spectrovss

Table of Contents

1 Introduction

This document describes the various tools that I developed to display spectrograms (time-frequency visualisation) of the Virgo data. All of them use the usual Fast Fourier Transform. There is no use of wavelet or Wigner-Ville representations. This implies constraints on time versus frequency resolutions and some poor separation between signal close in time or frequency. But it provides helpful information for Virgo commissioning, either in online monitoring with the VIM page [1] or to do noise investigation or to help GW detection validation.

The input data is always frame formatted data [2] and the tools described here are ROOT scripts, that use Frame library (Fr) [3] to read datam Frame vector library (Frv) [4] and FFTW library to do the FFT computation and, in some cases, the vimplot library [5] to produce the plots. The list of spectrogram tools is the following:

- **spectro_auto.C** : long duration spectrograms (typically day long)
- **minuteSpectro.C** : short duration spectrograms (typically hour long)
- **spectro_brms.C** : fast band rms computation over long duration
- **spectrovss.C** : spectrum of the time evolution of a signal FFT
- **rayleighSpectro.C** : Rayleigh spectrum or spectrogram, that evaluates noise stationarity

2 spectro_auto.C

2.1 Description

The purpose of this tool is to visualize long duration spectrograms (over more than two hours).
It uses as input the FFTs (0.1 Hz of frequency resolution), averaged over 300 seconds, computed online for a subset of the Virgo data channels.

Those FFTs are frame formatted and stored in the files pointed by /virgoData/ffl/spectro.ffl

The subset of Virgo channels is listed in /virgoData/VirgoOnline/SpectroMoni.cfg and is also available from the SpectroMoni configuration available from the VPM web interface.

The last version of spectro_auto.C is used to produce the VIM spectrogram plots and is thus available in /opt/MonitoringWeb/vim/spectro. The command line to use this tool is:

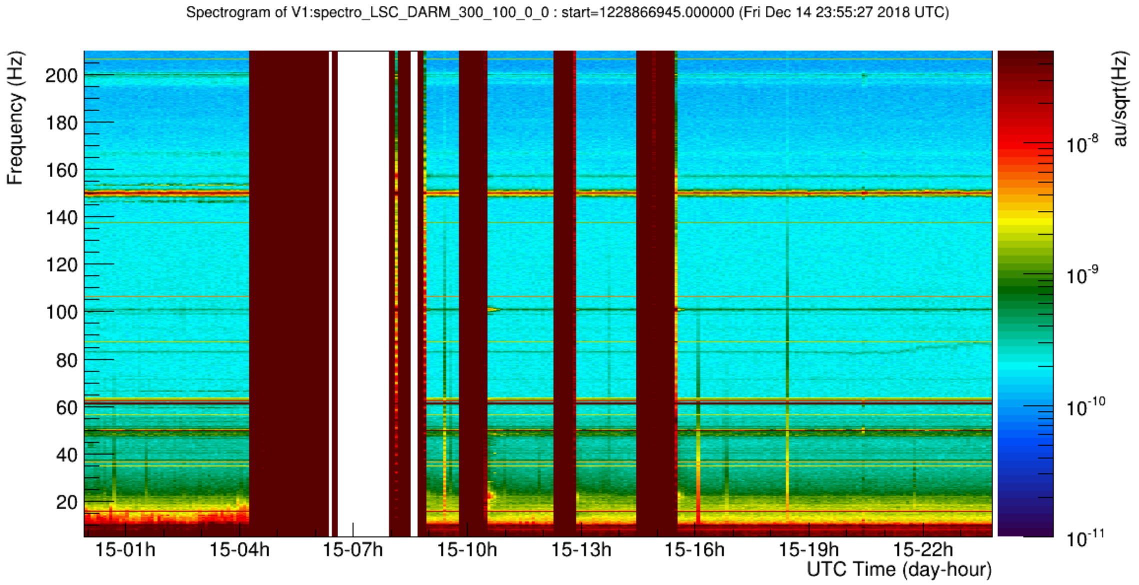
```
root –l /opt/MonitoringWeb/vim/spectro/spectro_auto.C\(channel, startgps, duration, deltaFnew,
fmin, fmax, zmin, zmax, zanim, medianflag\)
```

channel = input data channel (but only from the subset listed in SpectroMoni.cfg)
startgps = GPS time of the start of data read (if startgps=0 the last data available are read)
duration = Number of seconds of data read.
If duration<0, an average spectrum is computed instead of a spectrogram
deltaFnew = frequency resolution of the spectrogram (if deltaFnew=0 a resolution of 0.1Hz is used)
fmin = lower bound of frequency axis of the spectrogram
fmax = upper bound of the frequency axis of the spectrogram
zmin = lower limit of the color scale of the spectrogram
zmax = upper limit of the color scale of the spectrogram
zanim = if flag is positive, an animated gif is computed with a ramp on zmax
medianflag = if flag is positive, each frequency bin in the spectrogram is normalized by its median value

2.2 Examples

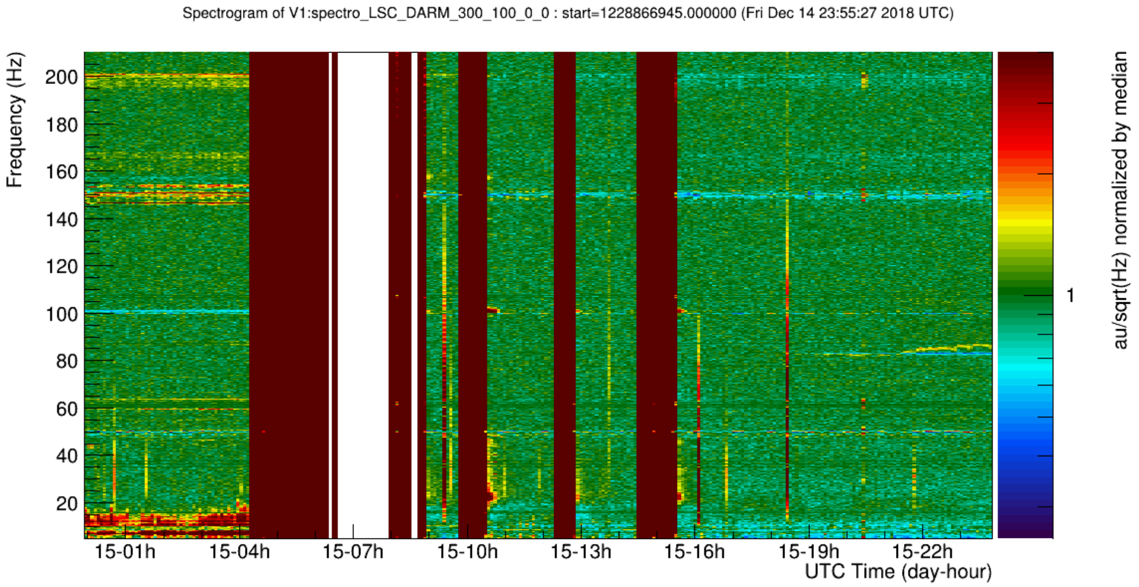
```
root -l /opt/w3/vim/spectro/spectro_auto.C\(\"LSC_DARM\",1228867218,86400,0.2,5,210,0,0,0,0\)
```

shows the spectrogram of one day of LSC_DARM channel, between 5 Hz and 210 Hz and with a frequency resolution of 0.2 Hz.



```
root -l /opt/w3/vim/spectro/spectro_auto.C\(\"LSC_DARM\",1228867218,86400,0.2,5,210,0,0,0,1\)
```

shows the same spectrogram but with the medianflag activated. This improves the constrast for any non-stationary noise like glitches or spectral lines changing in time.



3 minuteSpectro.C

3.1 Description

The purpose of this tool is to visualize short duration spectrograms (over less than two hours).
It uses as input the Virgo raw data pointed by /virgoData/ffl/raw.ffl
It computes the FFTs and takes thus much more time than spectro_auto.C but it allows to get spectrograms with a much better time resolution and can do spectrograms for any Virgo data channel.
It provides spectrograms which are similar to those provided by the Virgo dataDisplay [6].

The last version of minuteSpectro.C is used to produce the one hour VIM spectrogram plots and is thus available in /opt/MonitoringWeb/vim/spectro. The command line to use this tool is:

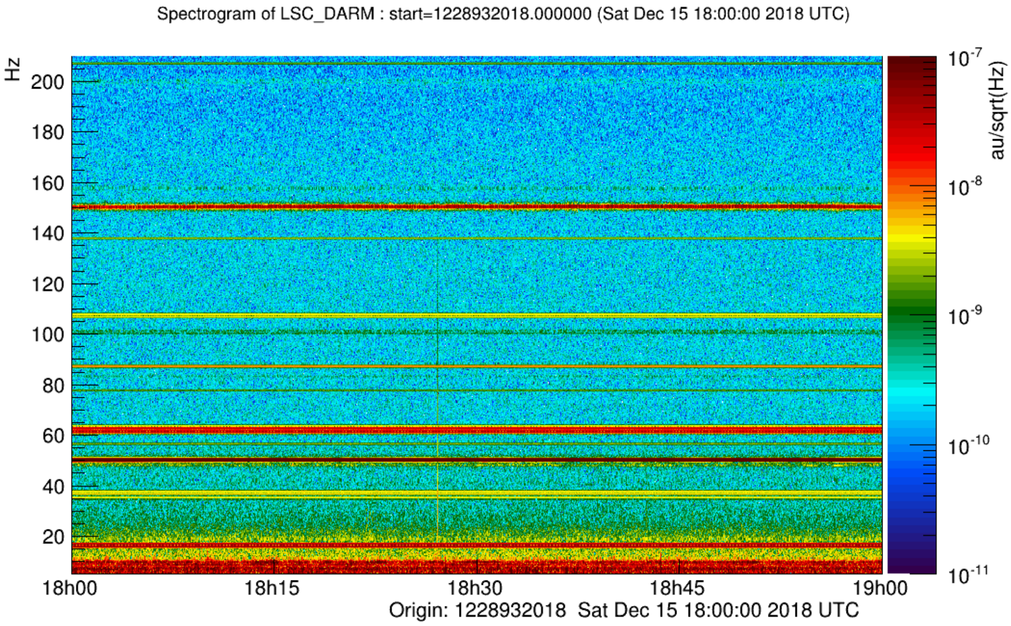
root –l /opt/MonitoringWeb/vim/spectro/minuteSpectro.C\(channel, startgps, duration, deltaFnew, fmin, fmax, fftlength, outputfile, zmin, zmax, medianflag, inputfile\)

where
channel = input data channel (but only from the subset listed in SpectroMoni.cfg)
startgps = GPS time of the start of data read (if startgps=0 the last data available are read)
duration = Number of seconds of data read.
If duration<0, an average spectrum is computed instead of a spectrogram
deltaFnew = frequency resolution of the spectrogram (if deltaFnew=0 a resolution of 0.1Hz is used)
fmin = lower bound of frequency axis of the spectrogram
fmax = upper bound of the frequency axis of the spectrogram
fftlength = the number of seconds of the time window over which the FFTs are computed
outputfile = path and name of the output file which will contain the plot, when used in batch mode
zmin = lower limit of the color scale of the spectrogram
zmax = upper limit of the color scale of the spectrogram
medianflag = if flag is positive, each horizontal line of the spectrogram is normalized by its median value
inputfile = path and name of the file containing the input data (default is raw.ffl)

3.2 Examples

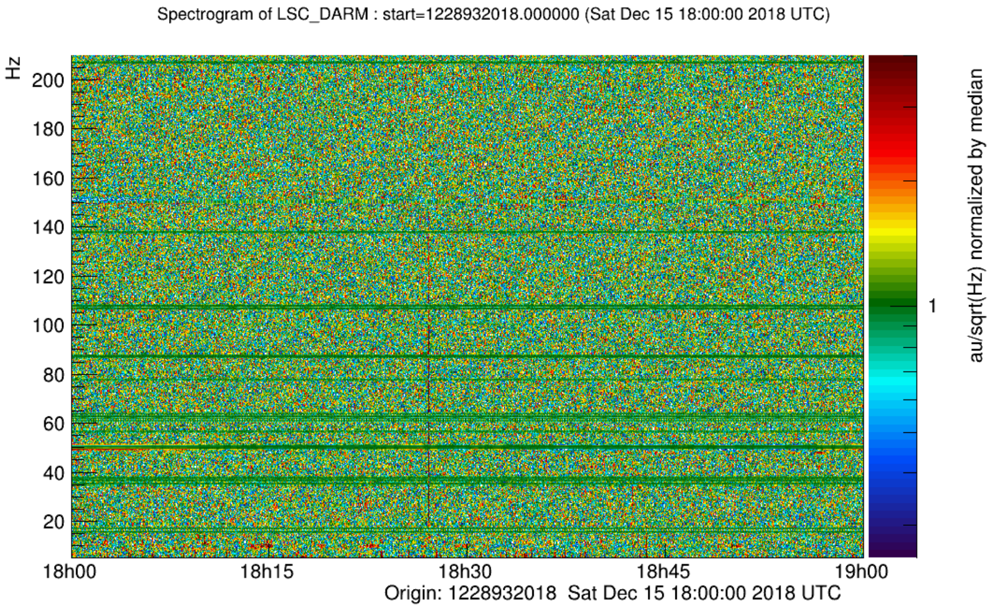
```
root -l /opt/w3/vim/spectro/minuteSpectro.C\(\"LSC_DARM\",1228932018,3600,0.5,5,210,0,0,0,0\)
```

shows the spectrogram of one hour of LSC_DARM channel, between 5 Hz and 210 Hz and with a frequency resolution of 0.5 Hz.



```
root –l /opt/w3/vim/spectro/minuteSpectro.C\(\"LSC_DARM\",1228932018,3600,0.5,5,210,0,0,0,1\)
```

shows the same spectrogram but with the medianflag activated. This improves the constrast for any non-stationary noise like glitches or spectral lines changing in time.



4 spectro_brms.C

4.1 Description

The purpose of this tool is to provide easily and quickly the computation of band rms in different frequency bands and over long duration, allowing thus to look over several days at the time evolution of the noise in a given set of frequency bands.

This is done by using the FFTs available in the frame formatted files pointed by /virgoData/ffl/spectro.ffl . It has thus the same constraint as spectro_auto.C : the use of only a subset of Virgo channels (whose list is available in /virgoData/VirgoOnline/SpectroMoni.cfg)

The last version of spectro_brms.C is used to produce some VIM plots and is thus available in /opt/MonitoringWeb/vim/spectro. The command line to use this tool is:

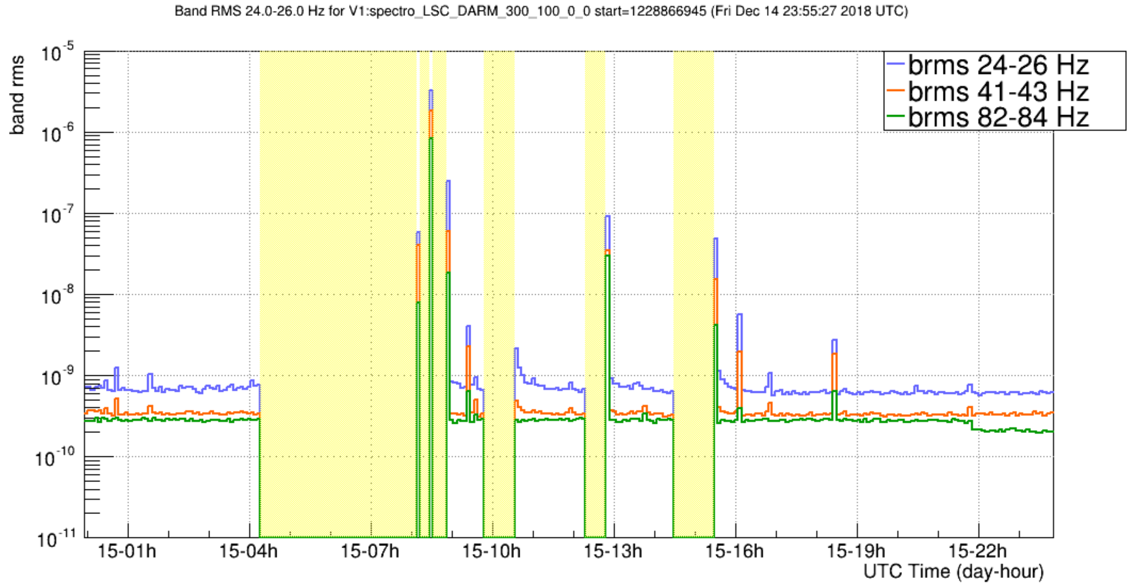
```
root –l /opt/MonitoringWeb/vim/spectro/spectro_brms.C\(channel, startgps, duration, bands, ymin, ymax\)
```

channel = input data channel (but only from the subset listed in SpectroMoni.cfg)
startgps = GPS time of the start of data read (if startgps=0 the last data available are read)
duration = Number of seconds of data read
bands = list of frequency bands over which band rms is computed
(it can be “20-30,70-80” or “20-30;70-80” or “20-30 70-80”)
ymin = lower bound of the band rms values visualized versus time
ymax = upper bound of the band rms values visualized versus time

4.2 Examples

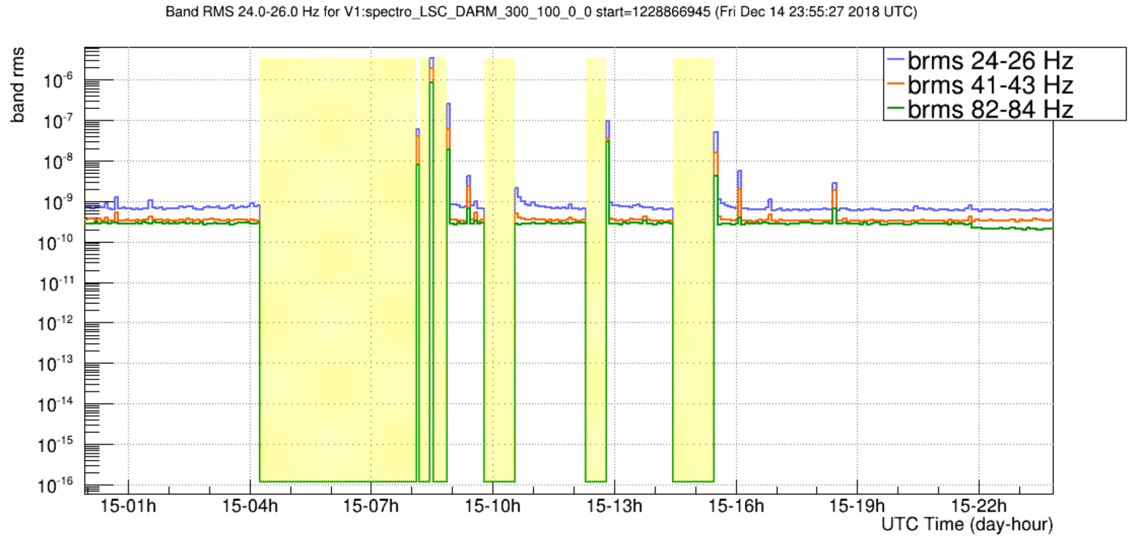
```
root –l  /opt/MonitoringWeb/vim/spectro /spectro_brms.C\(\"LSC_DARM\",1228867218,86400,
                                                         \"24-26\;41-43\;82-84\",1e-11,1e-5\)
```

shows the time evolution over one day of three band rms values computed every 300 sec. The yellow bands are the time periods where the ITF is not locked.



```
root –l /opt/MonitoringWeb/vim/spectro/spectro_brms.C\(\"V1:LSC_DARM\",1228867218,86400,
                                                       \"24-26\ 41-43\ 82-84\",-11111,-11111\)
```

shows same plot as before, but the ymin and ymax have their default values (automatic computation) and the frequency bands are separated by blanks instead of semicolons.



5 spectrovss.C

5.1 Description

The purpose of this tool is to compute the time evolution of each frequency bin of the FFT of a signal and to visualize, for each frequency bin, the spectrum of this time evolution. The result is a frequency-frequency representation where the frequency bins of the FFT of the signal are on the X axis and their spectra are on the Y axis.

It allows to see in a single eye's shot some periodicity in the non-stationarity of a signal in a given frequency band.

It uses as input the Virgo raw data pointed by /virgoData/ffl/raw.ffl
It computes FFTs of the signal and FFTs of the time series made by the time evolution of each FFT frequency bin. It takes thus much quite a long time and needs a long strectch of data to provide a result but it allows to focus on some low frequencies in the variation of non-stationary spectra lines.

The last version of spectrovss.C is used to produce some VIM noise monitoring plots and is thus available in /opt/MonitoringWeb/vim/noise. The command line to use this tool is:

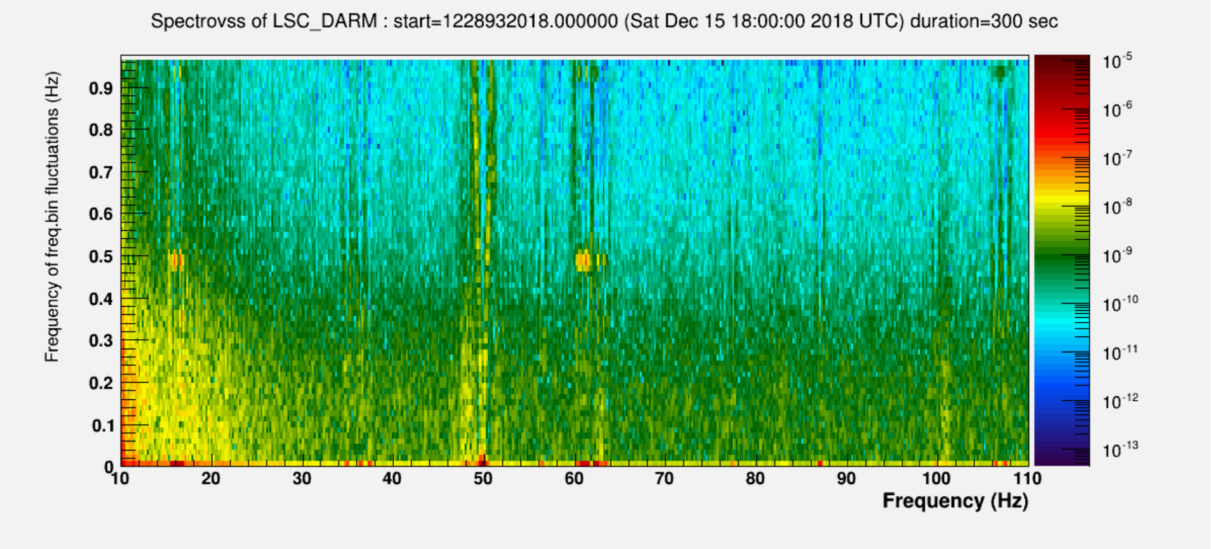
root –l /opt/MonitoringWeb/vim/noise/spectrovss.C\(channel, startgps, duration, deltaFnew, fmin, fmax, fftlength, nstep, nav, nav2, outputfile, inputfile\)

channel = name of input data channel
startgps = GPS time of the start of data read (if startgps=0 the last data available are read)
duration = Number of seconds of data read
deltaFnew = frequency resolution on the X axis (if deltaFnew=0 a resolution of 0.1Hz is used)
fmin = lower bound of frequency X axis
fmax = upper bound of the frequency X axis
fftlength = the number of seconds of the time window over which the signal FFTs are computed
nstep = lower limit of the color scale of the spectrogram
nav = number of signal FFTs averaged to get one sample of the time serie in each freq. bin
nav2 = number of freq bin time serie FFTs averaged to get the final spectrum of this freq bin.
outputfile = path and name of the output file which will contain the plot, when used in batch mode
inputfile = path and name of the file containing the input data (default is raw.ffl)

5.2 Examples

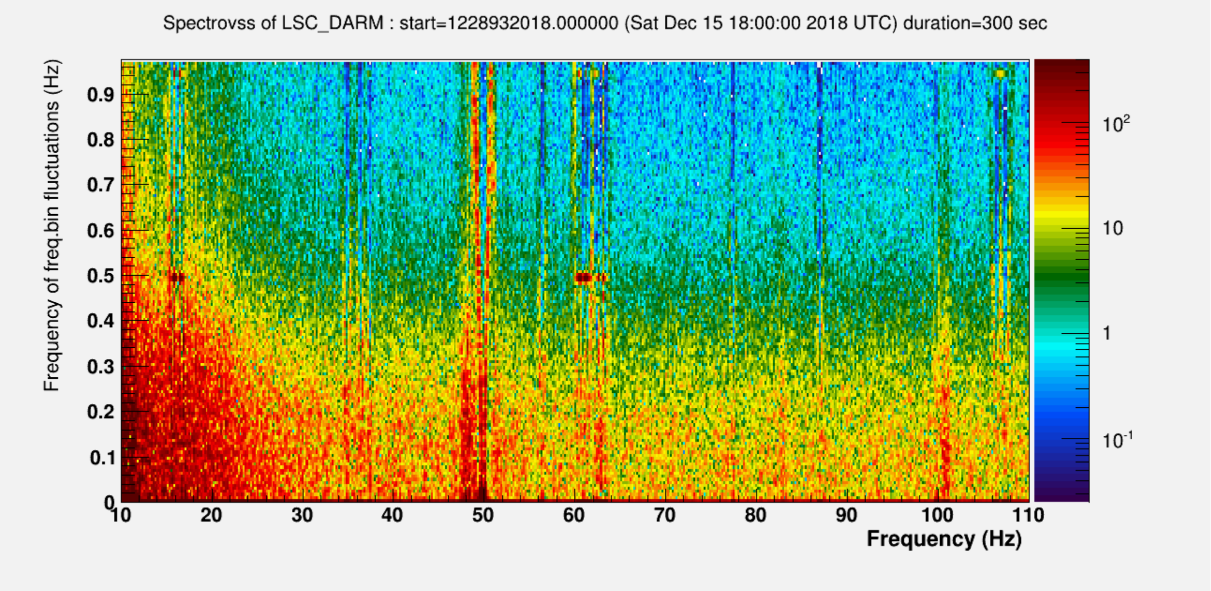
```
root -l /opt/MonitoringWeb/vim/noise/spectrovss.C\(1228932018,300,\"LSC_DARM\",
                                                  80,8,4,0.5,10,110,0,0,0\)
```

shows the Virgo Spectrum of Spectra of the LSC_DARM signal over 300 seconds in the frequency range 10Hz to 110 Hz, with a frequency resolution of 0.5Hz.



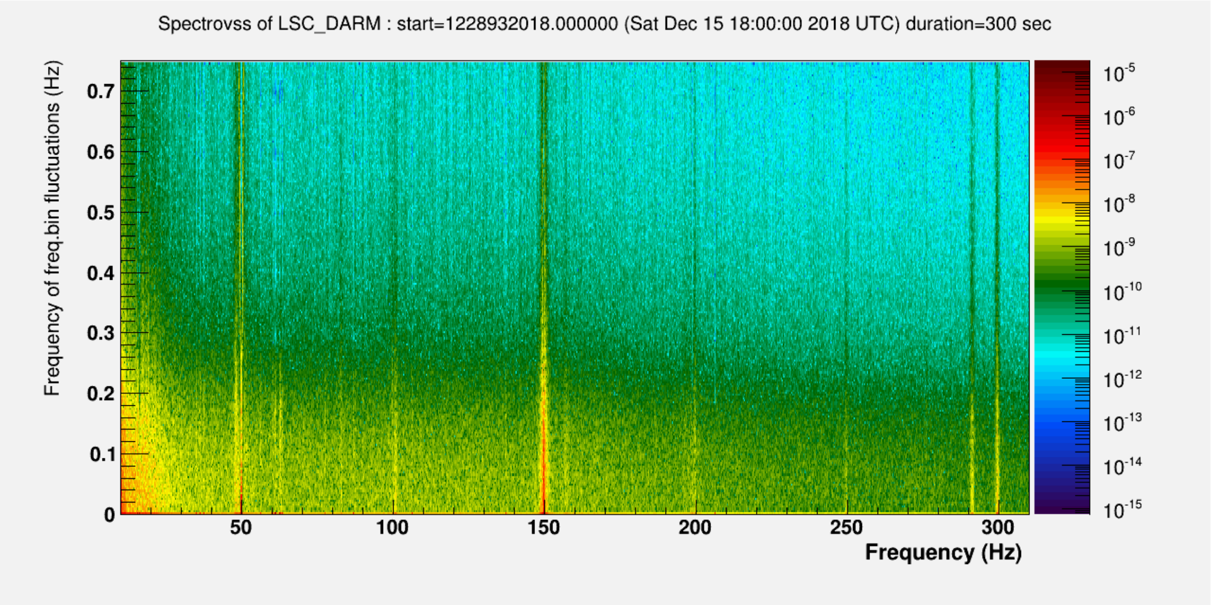
```
root -l /opt/MonitoringWeb/vim/noise/spectrovss.C\(1228932018,300,\"LSC_DARM\",
                                          80,8,2,0.5,10,110,0,0,\"/virgoData/ffl/raw.ffl\",1\)
```

Shows the same plot as before but with the medianflag set to 1. This gives the same weight to each frequency bin on X axis and allows to see with better contrast the fluctuations.



```
root -l /opt/MonitoringWeb/vim/noise/spectrovss.C\(1228932018,300,\"LSC_DARM\",
                                                          60,4,1,0.1,10,310,0,0,0\)
```

shows the Virgo Spectrum of Spectra of the LSC_DARM signal over 300 seconds in the frequency range 10Hz to 310 Hz, with a frequency resolution of 0.1Hz, with an other set of averaging and overlap parameters.



6 rayleighSpectro.C

6.1 Description

The purpose of this tool is to compute the Rayleigh spectrum or its time evolution (Rayleigh spectrogram) of a signal. In each frequency bin, it shows the ratio between the rms and the mean value of the FFT amplitude. As long as this ratio is near 0.5, the signal at this frequency can be considered as stationary.
It uses as input the Virgo raw data pointed by /virgoData/ffl/raw.ffl

The last version of rayleighSpectros.C is used to produce some VIM noise monitoring plots and is thus available in /opt/MonitoringWeb/vim/noise. The command line to use this tool is:

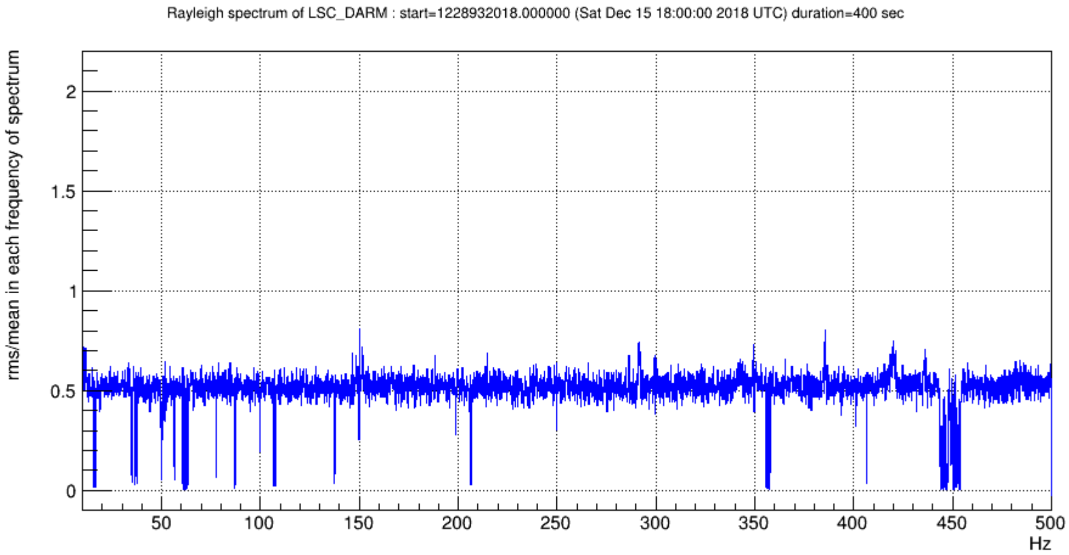
root –l /opt/MonitoringWeb/vim/noise/rayleighSpectro.C\(startgps, duration, channel, nav, fmin, fmax, fft_length, deltaFnew, path, inputfile, outputfile\)

startgps = GPS time of the start of data read (if startgps=0 the last data available are read)
duration = Number of seconds of data read
If duration<0, an average spectrum is computed instead of a spectrogram
channel = name of input data channel
nav = number of FFTs averaged to get the Rayleigh spectrum or each bin of the spectrogram.
fmin = lower bound of frequency X axis
fmax = upper bound of the frequency X axis
fftlength = the number of seconds of the time window over which the signal FFTs are computed
deltaFnew = frequency resolution on the X axis (if deltaFnew=0 a resolution of 0.1Hz is used)
nstep = lower limit of the color scale of the spectrogram
path = path of the output file
inputfile = path and name of the file containing the input data (default is raw.ffl)
outputfile = path and name of the output file which will contain the plot, when used in batch mode

6.2 Examples

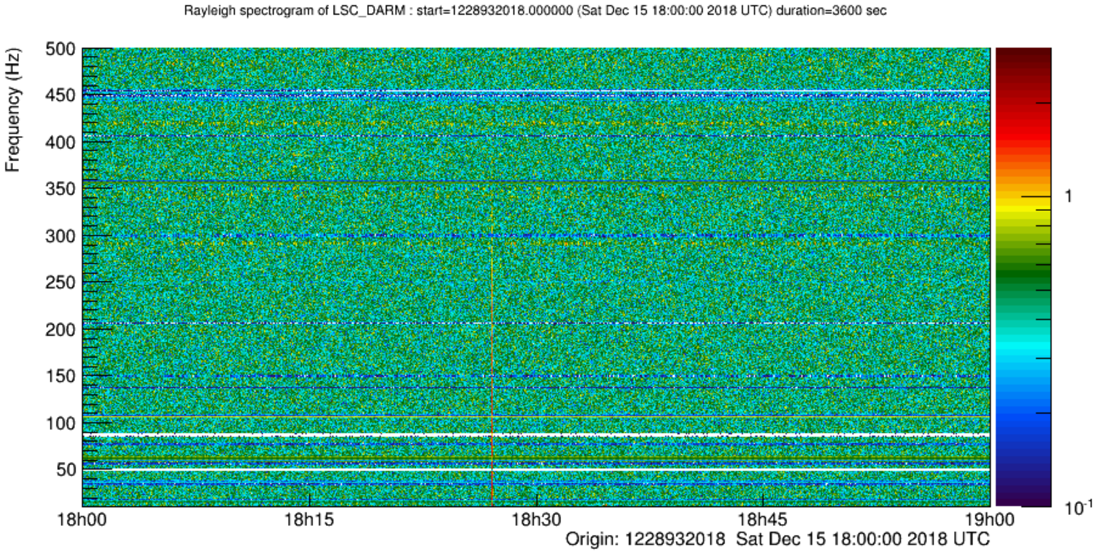
```
root -l /opt/w3/vim/noise/rayleighSpectro.C\(1228932018,-400,\"LSC_DARM\",0.1,10,500,10\)
```

shows the Rayleigh spectrum computed over 400 seconds of the LSC_DARM signal



```
root -l /opt/w3/vim/noise/rayleighSpectro.C\(1228932018,3600,\"LSC_DARM\",6,10,500,1,1\)
```

shows the time evolution of the Rayleigh spectrum over one hour of the LSC_DARM signal



7 Additional information

Those ROOT scripts are stored under Virgo svn in the vim package.Their latest versions are used in /opt/w3/vim/spectro or /opt/w3/vim/noise

A wiki page describes briefly how to use those scripts: https://wiki.virgo-gw.eu/DetChar/DetCharSpectro

8 Bibliography

[1] Virgo Interferometer Monitor (VIM): https://vim.virgo-gw.eu/?config=1

[2] The Frame Format: VIR-0067A-08 https://tds.virgo-gw.eu/ql/?c=2080

[3] The Frame library (Fr): http://lappweb.in2p3.fr/virgo/FrameL/

[4] The Frame Vector library (Frv): http://lappweb.in2p3.fr/virgo/FrameL/

[5] The vimplot library: VIR-05554A-16 https://tds.virgo-gw.eu/ql/?c=11877

[6] dataDisplay : VIR-0237A-15 https://tds.ego-gw.it/?content=3&r=11672