Virgo Interferometer Monitor (VIM)
Plot generator Users Guide

G. Hemming, D. Verkindt

VIR-0554A-16

https://tds.ego-gw.it/ql/?c=11877

14 December 2016

Change Record

Version	Date	Section Affected	Reason / Remarks
v1	14/12/2016	All	First version
v2	28/12/2016	5	Added fft plot

Table of Contents

1 Introduction

This document describes mainly the library vimplot.C used to develop easily ROOT scripts that generate plots for the Virgo Interferometer Monitor (VIM). Section 2 also describes the set of bash scripts used to periodically generate the VIM plots and archive them on a daily basis.

More details about VIM, especially about the Web Interface, can be found in the document VIR-0546A-16 (https://tds.virgo-gw.eu/ql/?c=11869).

2 Organisation of VIM plots production

All the plots (and all the scripts used to produce them) are located in a single repository. This repository is currently **/opt/MonitoringWeb/vim** and is pointed by a link **/opt/w3/vim**.

They are in fact contained in subsystem subdirectories of /opt/w3/vim: daq, env, omicron, mbta, sus, inf, inj, etc… In each subdirectory, a bash script **vimstart.sh** is in charge of calling the ROOT scripts that generate the plots. Periodically, the vimstart.sh scripts are started sequentially by a command system from a process called **MdVim** visible in the VPM web interface of Virgo.

Each vimstart.sh script:

- **First:** calls the script /opt/w3/vim/util/viminclude.sh that manages the main parameters used: GPSSTART, DURATION, VIMOUTPUT (start GPS time, duration of data read, destination directory where are written the plots). Default value of VIMOUTPUT is "./plots" if GPSSTART=0, and "./plotsoff" if GPSSTART>0 (offline reprocessing).

- **Secondly:** calls the ROOT scripts producing the plots (with GPSSTART and DURATION as first and second arguments)

- **Thirdly:** calls the script /opt/w3/vim/util/plotarchive.sh which reads the content of VIMOUTPUT and copy each file, with a naming convention, into the directory /opt/w3/vim/archive/year-month-day/resources, where year-month-day corresponds to the value of GPSSTART (GPSSTART=0 means current day).

Most of the ROOT scripts generating plots have a simple structure, thanks to the use of the vimplot.C library.

3 Example of a simple plot generator

As soon as the library /opt/w3/vim/util/vimplot.C is included, the ROOT script generating the plots can have a quite simple structure, with three main calls, as shown in the example below:

- **mw_init()** : to initialize global variables, get Frame library, use a specific ROOT style

- **mw_common()** : to read data, prepare the ROOT canvas and draw the plots in the pads of the ROOT canvas.

- **mw_printplot("png") :** to print the plots in a png file.

Example of a simple plot generator that uses vimplot.C :

```
#include "/opt/w3/vim/util/vimplot.C"

vimexample(double gpsstart=0, double duration=86400)  // two input parameters

{

 mw_init();

 ////////////////////// Define input and output parameters

 fileNameBase = "magneto";                  // prefix of the output file (full name will be magneto.png)

 fi = FrFileINew("/virgoData/ffl/trend.ffl");  // Open the trend data file

 length = duration;                         // length of data to read is equal to duration

 start = FrFileITEnd(fi)-length;            // default start GPS time of data to read is 'end of file - length'

 if (gpsstart > 700000000) start = gpsstart;   // if a valid gpsstart is provided, use it as start GPS time

// start and length are global variables used in vimplot.C and available anywhere in this example script.

 ////////////////////////// Mandatory parameters

 nPadsx = 1; nPadsy = 2;                    // number of pads in the canvas is 2 (2x1)

 channel[0][0] = "V1:ENV_WEB_SEIS_N_mean";  // Set channel to be shown as first plot of pad 1
channel[0][1] = "V1:ENV_NEB_SEIS_N_mean";  // Set channel to be shown as second plot of pad 1

 channel[1][0] = "V1:ENV_CEB_SEIS_N_mean";  // Set channel to be shown as first plot of pad 2
```

Any line added to this example will then be for setting optional parameters or using ROOT commands to change the canvas, the plots, the pads or the legends parameters.

4 Mandatory parameters in vimplot.C

start : usually, it is equal to the first parameter (gpsstart) of your ROOT script

length: usually, it is equal to the second parameter (duration) of your ROOT script

viminput : name of the input frame file to read (usually you set it to /virgoData/ffl/trend.ffl)

fi: file pointer provided by a call to FrFileINew("name of your input data file"). If **viminput** is provided, then **fi** is redefined and points to the **viminput** file.

fileNameBase : prefix of the output file name. Usually, it is identical to the name of the ROOT script

nPadsx : number of pads on horizontal axis of the ROOT canvas (each pad contains at least one plot)

nPadsy : number of pads on vertical axis of the ROOT canvas (when nPadsx=2 and nPadsy=2, ipad=0 is the upper left pad, ipad=4 is the lower right pad)

channel[ipad][ich] : name of the input data channel to be read for the plot ich of the pad ipad

5 Optional parameters

Most of the optional parameters have default setting that are well chosen according to the number of pads and the number of plots in each pad. But you can modify some of them without any knowledge of the ROOT commands.

5.1 Optional parameters : general

path : output directory where are written the output files (default is set by VIMOUTPUT environment variable and is "./plots" when gpsstart=0).

title : global title of the canvas (default is empty string).

xsize : horizontal size of the canvas (default is 1000 if nPadsx=1 or 1500 if nPadsx>1)

ysize : vertical size of the canvas (default is 500 if nPadsy=1 or 750 if nPadsy>1)

5.2 Optional parameters for each plot

color[ipad][ich] : line color of the plot ich of the pad ipad

scale[ipad][ich] : the plot ich of pad ipad is multiplied by scale[ipad][ich]

bias[ipad][ich] : the value bias[ipad][ich] is added to the plot ich of pad ipad

padtitle[ipad] : contains the title shown above the plots of pad ipad.

plottype[ipad] : the type of the plots of pad ipad. Default is "time".
- If you set plottype[0]="fft", a spectrum plot will be made in pad 0. One FFT is computed for each 10000 samples (thus frequency resolution is the sampling frequency divided by 10000). The result is a mean spectrum over the full duration of the data read. You can modify the default value of 10000 samples by setting the value **nsample[ipad]**.
- If you set plottype[0]="2d", a 2D plot will be made in pad 0. channel[0][0] will be on X axis and channel[0][1] will be on Y axis.
- If you set plottype[0]="dq2d", a 2 dimensions plot will be made in pad 0. Time will be on X axis, channel[0][i] will be on Y axis, the values of channels channel[0][i] will be on Z axis (color axis).

nsample[ipad] : set the number of samples to use to compute FFT for a plot of type "fft".

plotMin[ipad] : Set the min value on Y axis for the plots in pad ipad (default is plotMin=0)

plotMax[ipad] : Set the max value on Y axis for the plots in pad ipad (default is plotMax=0). If plotMin>=plotMax, the min and max on Y axis are computed automatically.

calib[ipad] : if this value is positive, the data vectors shown in pad ipad are calibrated and the units of first plot of pad ipad is displayed on Y axis.

plotStat[ipad] : if positive, shows statistics box for first plot of pad ipad.

legendsize[ipad] : sets the size of the text in the legend box for plots of pad ipad.

logx[ipad] : if positive, sets log scale on X axis of plots of pad ipad

logy[ipad] : if positive, sets log scale on Y axis of plots of pad ipad

nologx[ipad] : if positive, the log scale on X axis of plots of pad ipad is shown without exponents

nology[ipad] : if positive, the log scale on Y axis of plots of pad ipad is shown without exponents

nolegend[ipad] : if not zero, the legend of plots of pad ipad is not shown

5.3 Optional parameters : X and Y axis

plotGrid[ipad] : if positive, sets the grid on X and Y axis of the pad ipad (default is plotGrid=1)

labelsizex[ipad] : sets the size of the X axis labels of plots of pad ipad

labelsizey[ipad] : sets the size of the Y axis labels of plots of pad ipad

labelX[ipad] : sets the title on X axis. For instance labelX[0]=”Volts”; (default is empty string).

labelY[ipad] : sets the title on Y axis. For instance labelY[0]=”Volts”; (default is empty string).

5.4 Optional parameters: myvect

You can create your own data vector and insert it as input for one plot. This allows you to do complex computation but still to use vimplot.C to make the plot, by putting the result in myvect[ipad][ich].

myvect[ipad][ich] : a data vector that contains user created data to be shown on plot ich of pad ipad. Should always be called before the call to mw_common().

An example is shown below, where in pad 3, the first plot is the channel V1:ENV_WEB_SEIS_N and the second plot is the data vector v3 made with the sum of 2 channels:

```
channel[3][0] = "V1:ENV_WEB_SEIS_N_mean";

FrVect *v1 = FrFileIGetVectD(fi,"V1:ENV_WEB_SEIS_N_mean", start, length);

 FrVect *v2 = FrFileIGetVectD(fi,"V1:ENV_CEB_SEIS_N_mean", start, length);

 FrVect *v3 = FrvAdd(v1,v2,NULL,"ENV_WEB_SEIS_N+ENV_CEB_SEIS_N");

 myvect[3][1] = v3;
```

6 Canvas, pads, legends and plots access

Once mw_common() has been called (thus plots have been drawn), the canvas, the plots, the pads and the legends properties can be modified using common ROOT commands. This must be done, of course before the call to mw_printplot("png").

can : pointer to the ROOT canvas containing the pads containing the plots. It can be used for instance to change the color of the border: can->SetBorderMode(kRed);

globalTitle : pointer to the global title of the canvas. It can be used for instance to change the size of the global title : globalTitle->SetTextSize(0.06);

pad[ipad] : pointer to the pad number ipad. It can be used for instance to set log scale on y axis: pad[0]->SetLogy(1);

plot[ipad] : pointer to the first plot of the pad ipad. It can be used for instance to set a new title on X axis : plot[0]->GetXaxis()->SetTitle("my new title");

legend[ipad] : pointer to the legend box showing the channels displayed. It can be used for instance to change the size of the legend text : legend[0]->SetTextSize(0.06);

7 Example with optional parameters

```
////////////////////////// Optional parameters
color[0][0] = kBlack;              // Set kRed as the color of the first plot of pad 0
color[0][1] = kRed;                // Set kRed as the color of the  second plot of pad 0
color[1][0] = kBlack;              // Set kRed as the color of the first plot of pad 1
path = ".";                        // path of the output directory where is saved the png file
title = "Seismometers";            // Title of the canvas is "Seismometers"
calib[0] = 1;                              // Data shown in pad 0 is calibrated
padtitle[0] = "NEB-WEB Seismic";  // Title of the pad 0 is "NEB and WEB Seismic"
padtitle[1] = "CEB Seismic";          // Title of the pad 1 is "CEB Seismic"
scale[0][1] = 1.2;                         // Multiply by 1.2 the data shown as second plot of pad 0
bias[0][1] = 0.33;                         // Add 0.33 to the data shown as second plot of pad 0
// plottype[0]="2d";                  // Use the first two channels defined in pad 0 to build a 2d plot
labelX[1] = "my VoltsX";              // Label on X axis of pad 1 is "my VoltsX"
labelY[1] = "my VoltsY";              // Label on Y axis of pad 1 is "my VoltsY"
```

```
 plotMax[0] = 30;              // max of the pad 0 is set to 30
 plotMin[0] = 20;              // min of the pad 0 is set to 20
 labelsizex[0] = 0.05;         // X axis of pad 0 has label size 0.05
 labelsizey[0] = 0.05;         // Y axis of pad 0 has label size 0.05
 legendsize[0] = 0.06;         // Legends of pad 0 have text size 0.06
logy[0] = 1;                   // Set log scale on Y axis of plots of pad 0
nology[0] = 1;                 // No log display on Y axis of plots of pad 0
 ////////////////////// Read data, do the plots
 mw_common();
 ////////////////////// Any modification once the plots have been drawn
 can->SetBorderMode(kRed);          // Set the border of canvas to kRed
 plot[0]->SetTitle("mytitle");      // Set title of pad 0 to "mytitle"
 plot[0]->SetTitleSize(0.03,"x");   // Set to 0.03 the size of title on X axis of plots of pad 0
 plot[0]->SetLabelSize(0.03,"x");   // Set to 0.03 the size of title on X axis of plots of pad 0
 leg[0]->SetFillColor(kGreen);      // Color in green the legend box of pad 0
pad[1]->SetFillColor(kRed);         // Color in red the pad 1
globalTitle->SetTextColor(kRed); // Color in red the text of the canvas title
```

8 Other VIM resources

VIM resources are not only plots made with vimplot.C. They can be produced with home-made ROOT scripts or any code (bash, matlab, python, etc…). They can be plots but also HTML or ASCII files, including tables or simple text with html links.

8.1 Home-made ROOT scripts

The library vimplot.C is recommended to create plot generators. If you know the ROOT commands, you can also created your own ROOT script. In this case, try to output PNG files and try to include anyway vimplot.C with a call to mw_init(). This will guarantee you some standard ROOT style and a standard way to manage the location of the output files.

8.2 Resource name convention

Those resources can be either static files or periodically updated files, but make sure that the file name does not change (for instance, do not use current GPS time in the file name).

8.3 Use of plotarchive.sh

Whatever the resources created, they should always be managed, whenever possible, by the vimstart.sh script. At least, a call to the utility **/opt/w3/vim/util/plotarchive.sh** is needed to transfer the resources to the vim archive with the right filename convention.

9 Reprocessing

If, for a particular day, some ressources have not been produced, they can be reprocessed using the utility **/opt/w3/vim/vimreprocess.sh**

It takes 3 arguments:

arg1 : name of the subsystem directory to reprocess (daq, env, sus, etc…). If arg1=”all”, then all the subsystems are reprocessed.

arg2 : date or GPS time within the day to reprocess (2016/11/30 or 1164500000)

arg3 : duration of the plots (default is 90000 sec)

By default, all the plots generated by the reprocessing will start at 00:00:00 UTC of the day reprocessed and will be 90000 sec long (one day + one hour).

10 Help, advices and utilities

If you plan to add a new resource generator, always work in a /opt/w3/vim subdirectory and use the the **virgorun account**. It is then easier for administrators to have the write access.

If you plan to add a new set of resources, that means a new subdirectory in /opt/w3/vim, **first contact one of the VIM administrators**.

The subdirectory **/opt/w3/vim/util** contains a set of utilities or examples:

- **startone.sh :** allows to start easily one plots generator based on ROOT
- **plotarchive.sh :** allows to archive in /opt/w3/vim/archive the resources generated locally.
- **vimreprocess.sh :** allows to reprocess the plots of one or several subsystems
- **viminclude.sh :** used only as a starting part by each vimstart.sh script
- **vimexample.C :** an example of a ROOT script that generates plots by using the vimplot.C functionalities. Each command is described by a comment line.
- **vimexample2.C:** an example of the use of extended functionalities of vimplot.C, like the 2D plots or the use of myvect[ipad][ich].

This documentation is available in the TDS but also as a PDF document in /opt/w3/vim/util/vimhelp.pdf

An Help panel is also available in the VIM web user interface: https://vim.virgo-gw.eu