

Moving RMS

The **Moving_RMS** command computes the linear envelope of a signal (typically an EMG signal) by calculating the RMS value within a moving window. The RMS is described [here](#)

The number of frames in the window must be an odd number. The output value at each frame is the RMS of the window centered on that frame.

Pipeline Command

The **GCVSPL** command can be found in the Pipeline Workshop within the Signal Filter folder as so:

```
Moving_RMS
! /Signal_Types=
! /Signal_Folder=ORIGINAL
! /Signal_Names=
! /RESULT_TYPES=
! /RESULT_FOLDERS=PROCESSED
! /RESULT_NAME=
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=
! /NUM_WINDOW_FRAMES=3
! /IF_EVEN_FRAMES_INCREMENT_1=FALSE
;
```

Command Parameters

The parameters that can be modified for this command are as follows:

Signal Types	The type of signal to be processed
Signal Folder	The name of the origin signal folder
Signal Names	The names of the signals to be processed
Signal Components	Which components of the signal to be filtered
Result Types	The signal type of the resulting signals
Result Folder	The data tree folder in which the resulting signal is placed
Result Name	The name of the resulting signal that is produced
Apply As Suffix To Signal Name	True or False: Apply Result Name as a suffix to the original signal name
Num Window Frames	Number of frames included in moving window
If Even Frames Increment 1	True or False: If frame window is an even number add 1 to make it odd and centered around middle frame

Dialog

The command can be defined by editing as text using the parameter definitions above, or using the dialog box:

Result Name	Signal Type	Signal Folder	Signal Name
-------------	-------------	---------------	-------------

Frame Window

The command Moving_RMS computes the RMS of a signal by calculating the RMS value for short successive time periods (the time period is described as a moving window). In Visual3D the windows overlap. The output value at each frame is the RMS of the window centered on that frame (e.g. the RMS value for each window is plotted at the time corresponding to its center) so the number of frames in the window must be an odd number.

So, if the window was 11 frames.

- The first output frame is frame 6 computed from frames 1-11.
- The second output frame is frame 7 computed from frames 2-12 Etc.

Typically, this window is reported as a time. Since the number of frames is dependent on the sampling rate, you must convert the time (in seconds) to frames. The sampling rate is specified in the C3D file under PARAMETER::ANALOG::RATE.

To convert time to frames, you must multiply the time by the sampling rate and add 1 (time 0.0 is equal to frame 1). This as an expression is:
$$(\text{time} * \text{PARAMETER::ANALOG::RATE}) + 1$$

To specify a frame window of 100 ms (0.1 seconds), would be specified as:
$$\text{/NUM_WINDOW_FRAMES} = (0.1 * \text{PARAMETER::ANALOG::RATE}) + 1$$

Example: Pipeline commands for specifying a window of 100 ms

A more typical window for the RMS filter is 100 ms, which can be achieved as follows:

```
! -----
! compute a moving rms with a window of 100 ms
! -----

Moving_RMS
/SIGNAL_TYPES=ANALOG
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=EMG1+EMG2
/RESULT_TYPES=ANALOG
/RESULT_FOLDERS=RMS
/RESULT_NAME=
/APPLY_AS_SUFFIX_TO_SIGNAL_NAME=TRUE
/NUM_WINDOW_FRAMES=(0.1*PARAMETER::ANALOG::RATE)+1
! /IF_EVEN_FRAMES_INCREMENT_1=FALSE
;
```

Example: Pipeline commands for specifying a window of 150 ms

To specify a 150 ms (0.15 seconds) window for the RMS, the commands are as follows:

```
! -----
! compute a moving rms with a window of 150 ms
! -----

Moving_RMS
/SIGNAL_TYPES=ANALOG
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=EMG1+EMG2
/RESULT_TYPES=ANALOG
/RESULT_FOLDERS=RMS
/RESULT_NAME=
/APPLY_AS_SUFFIX_TO_SIGNAL_NAME=TRUE
/NUM_WINDOW_FRAMES=(0.15*PARAMETER::ANALOG::RATE)+1
! /IF_EVEN_FRAMES_INCREMENT_1=FALSE
;
```

From:
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:signal_commands:moving_rms&rev=1777311859

Last update: 2026/04/27 17:44

