

==== Contents ====

- * [1 General Processing Steps](#)
- * [2 Helpful Pipeline Functions](#)
- * [3 Meta Commands](#)
- * [4 Removing Noise from EMG Signals](#)
- * [5 Normalization and Scaling Methods](#)

General Processing Steps

Processing EMG signals typically follows some variation of the following steps:

1. High Pass Filter 2. Low Pass Filter 3. Rectify Data 4. Filter Noise 5. Normalize and Scale Data It should be noted that within both this page and external literature, the term “Linear Envelope” refers to an EMG signal that has been rectified and low pass filtered.

Visual3D has a number of helpful pipeline functions (linked below) that can automate the general processing steps.

Helpful Pipeline Functions

The following links will redirect you to pages detailing the use of Visual3D pipeline commands that are commonly used in the general processing of EMG data:

- [High Pass Filter \(Removing DC Bias\)](#)
- [Lowpass Filter](#)
- [Rectify](#)
- [Moving RMS](#)
- [Teager Kaiser Energy Operator](#)
- [Assign Tags To File](#)

Meta Commands

This page deals with several examples of using Visual3D [Meta-Commands](#). Meta commands are essentially a way for you to write and upload your own function to the Visual3D pipeline. They have the file extension “.v3m”. Meta commands must be stored in a folder labelled **Meta-Commands**, which should be located in the Visual3D **Plugins** folder. The Plugins folder may not exist, so you may have to create it yourself. If you receive errors after running a meta-command, navigate to **Edit > Preferences** in Visual3D and check that the directory of the Plugins folder matches the location Visual3D is looking for it in. Restart Visual3D before running the command again.

Removing Noise from EMG Signals

Download the [EMG_NoiseReductionExample](#) here.

The text below describes two common methods of removing noise from EMG signals. Before using these methods, ensure you have performed a high and low pass filter on the raw data. You do not need to rectify your data to use the Root Mean Square method or the Teagan Kaiser Energy Operator method as they will both rectify your signals for you.

1. Root Mean Square (RMS): The RMS method is typically used when finding overall energy, muscle activation patterns, or for fatigue analysis. The RMS method takes the square root of the mean square of a set of values. Because it deals with the mean square of a data set, the input EMG signal does not need to be rectified before calculating the RMS. Example application of this command shown below:

```
! Example 1: Use the Moving RMS Command to Process EMG Data
! Apply a high pass filter with 50 Hz cutoff
Highpass_Filter
/Signal_Types=ANALOG
/Signal_Names=EMG1+EMG2+EMG3
/Signal_Folder=ORIGINAL
! /Result_Suffix=
! /Result_Folder=PROCESSED
! /Filter_Class=BUTTERWORTH
/Frequency_Cutoff= 50
/Num_Reflected= 0
/Total_Buffer_Size=100
/Num_Bidirectional_Passes=1
;
! Apply a low pass filter with 500 Hz cutoff
Lowpass_Filter
/Signal_Types=ANALOG
/Signal_Names=EMG1+EMG2+EMG3
/Signal_Folder=PROCESSED
! /Result_Suffix=
! /Result_Folder=PROCESSED
! /Filter_Class=BUTTERWORTH
/Frequency_Cutoff= 500
/Num_Reflected= 0
/Total_Buffer_Size=100
/Num_Bidirectional_Passes=1
;
! Moving RMS with 100 millisecond window
Moving_RMS
/SIGNAL_TYPES=ANALOG
/SIGNAL_NAMES=EMG1+EMG2+EMG3
/SIGNAL_FOLDER=PROCESSED
! /RESULT_SUFFIX=
/RESULT_FOLDER=RMS
```

```
/NUM_WINDOW_FRAMES= ( 0.1*PARAMETER::ANALOG::RATE ) + 1
;
```

2. Teager Kaiser Energy Operator (TKEO): The TKEO method is typically used when finding muscle onset/offset contracts and analyzing frequency data. The TKEO method is a signal processing technique used to extract information from a time-series signal. It involves calculating the Teager energy of the signal, which is a measure of the signal's instantaneous power, and then smoothing the energy values using a Kaiser window. The resulting smoothed energy values can be used to identify important features in the signal, such as peaks or changes in frequency. Example application of this command shown below:

! Example 2: Use the TKEO Command to Process EMG Data

```
Highpass_Filter
/SIGNAL_TYPES=ANALOG
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=EMG_C
/RESULT_FOLDER=PROCESSED
! /RESULT_SUFFIX=
! /FILTER_CLASS=BUTTERWORTH
/FREQUENCY_CUTOFF=20
/NUM_REFLECTED=0
! /NUM_EXTRAPOLATED=0
/TOTAL_BUFFER_SIZE=100
! /NUM_BIDIRECTIONAL_PASSES=1
;
Lowpass_Filter
/SIGNAL_TYPES=ANALOG
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=EMG_C
/RESULT_FOLDER=PROCESSED
! /RESULT_SUFFIX=
! /FILTER_CLASS=BUTTERWORTH
/FREQUENCY_CUTOFF=6.0
/NUM_REFLECTED=6
! /NUM_EXTRAPOLATED=0
/TOTAL_BUFFER_SIZE=6
! /NUM_BIDIRECTIONAL_PASSES=1
;
Teager_Kaiser_Energy
/SIGNAL_TYPES=ANALOG
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=EMG_C
! /RESULT_TYPES=
/RESULT_FOLDERS=PROCESSED
/RESULT_NAME=EMG_C_TKEO
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=
;
```

3. Integrating EMG Signals: Integrating EMG signals can be useful for quantifying muscle activity, comparing muscle activity between conditions, and assessing fatigue. The result of integrating an EMG signal (an iEMG) is a single metric value of microvolts or millivolts. While this isn't strictly a noise

reduction method, it does have applications once a signal has been filtered. An example of using Visual3D's pipeline commands to integrate an EMG signal can be found here: [EMG Integrate](#). **4. Removing Background Noise Example:** The [Removing Background Noise](#) page shows an example of using the Visual3D pipeline and Meta Commands to implement a noise reduction algorithm. The algorithm finds the mean of the data, rectifies the data around a mean baseline, and calculates threshold values to store as a global variable.

Normalization and Scaling Methods

Normalizing data is a useful transformation for comparing signals across groups of data. This section discusses several methods of using Visual3D's pipeline features to normalize and scale EMG signals after they have been filtered.

1. Normalize and Scale EMG Signal to Global Maximum: One method of normalizing EMG signals is to use the maximum voluntary contraction (MVC) value. MVC values are an EMG signal measuring during a maximal effort task, which is then used as a reference value to normalize EMG signals during all subsequent experimental tasks. What this translates to in Visual3D is first calculating the maximum value of the reference signal and store it in the Global workspace, then normalize your signals by dividing them by this global metric. Sample files and pipelines are available [here](#) and [here](#). **2. Compute Normalization Scale Factors:** Here are two examples of the steps that would be used to compute a normalization factor using MVC data. **Example 1: Single MVC Trial:** The process for computing a normalization factor from a single MVC value uses the following general steps:

1. Process EMG
 1. High pass filter (remove DC offset)
 2. Calculate RMS of predetermined time intervals (example: 0.02 seconds)
2. Normalize
 1. Calculate the average of the processed signal over 30 second intervals
 2. Get the maximum of the 30 second interval
 3. Normalize by dividing the average by the maximum

Example 2: Multiple MVC Trials

1. Process EMG
 1. High pass filter (remove DC offset)
 2. Calculate RMS of predetermined time intervals (example: 0.02 seconds)
2. Normalize
 1. Calculate the average of the processed signal over some time interval (in this example: 0.5 seconds)
 2. Get the maximum of the processed signal
 3. Calculate the maximum of the average signal
 4. Normalize by dividing the processed signal by the maximum average over the interval

Sample files and pipelines are available [here](#).

3. Normalize to Maximum within Gait Cycle: The image to the (right) shows four “checkpoints” in normalizing a signal to the maximum within a gait cycle.

* First graph: raw EMG signal with gait events RHS (pink) and RTO (purple). These correspond to the signal within ANALOG::ORIGINAL in the Sample_Workspace.cmz download.

* Second graph: the linear envelope (ANALOG::ENVELOPE) showing the rectified and smoothed signal with maximums A (~0.76) and B (~0.69).

* Third graph: all frames within each gait cycle set to the maximum value, and all other frames are set to zero.

* Fourth graph: the linear envelope signal is divided by the maximum of the corresponding gait cycle. Notice how the peaks of both cycles are now 1, and all values outside the defined event sequence are set to “data not found”.

Sample files and pipelines are available [here](#).

Retrieved from “”

From:

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

Permanent link:

https://has-motion.com/wiki/doku.php?id=visual3dtest:documentation:emg:processing_tools&rev=1718300352

Last update: **2024/06/13 17:39**

