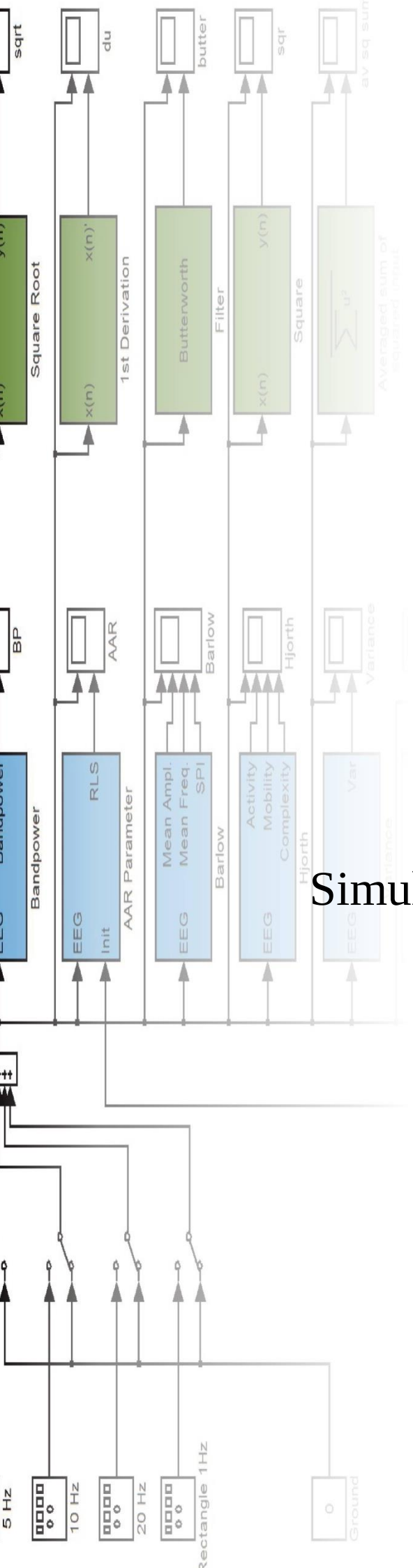


g.tec – medical engineering GmbH
Sierningstrasse 14, 4521 Schiedlberg, Austria
Tel.: (43)-7251-22240-0
Fax: (43)-7251-22240-39
office@gtec.at, http://www.gtec.at

Simulink Highspeed Library

USER MANUAL V3.16.01

Content

Release Notes

Release notes bring to your attention new features of, and changes to, the Simulink Highspeed Library when upgrading to a newer version of the driver.

New features

- g.SCOPE: visualize amplitude spectrum, power spectrum or power spectral density next to EEG data
- ParadigmPresenter: generic audio, video, text and picture paradigm presenter

Changes

- EEGlab importer: changed naming of events
- Online FFT

Hardware and Software Requirements

Hardware Requirements

g.tec Highspeed requires a PC compatible desktop or notebook workstation running Microsoft Windows.
The table below lists optimal settings:

Hardware	Properties
CPU	Pentium working at 3000 MHz
Hard disk	100 gigabyte
RAM	8 gigabyte
USB 2.0 high speed port	One free USB port for the Hardlock Dongle

Software Requirements

Make sure that the MATLAB installation works correctly before installing the g.tec Highspeed software. Depending on your Windows operating system, administrator rights might be necessary for the installation.

Software	Version
MATLAB	Release 2015a
Simulink	Release 2015a
Signal Processing Blockset	Release 2015a
DSP System Toolbox	Release 2015a
Windows	Windows 10 Pro Threshold 2 Win64
Acrobat Reader	2015.009.20069
.net Common Language Runtime	4.6.1

Files on your Computer

The default installation directory for g.tec products is `C:\Program Files\gtec\`.

Within this directory, a subdirectory named `gtecHS` is generated, where all library files are located:

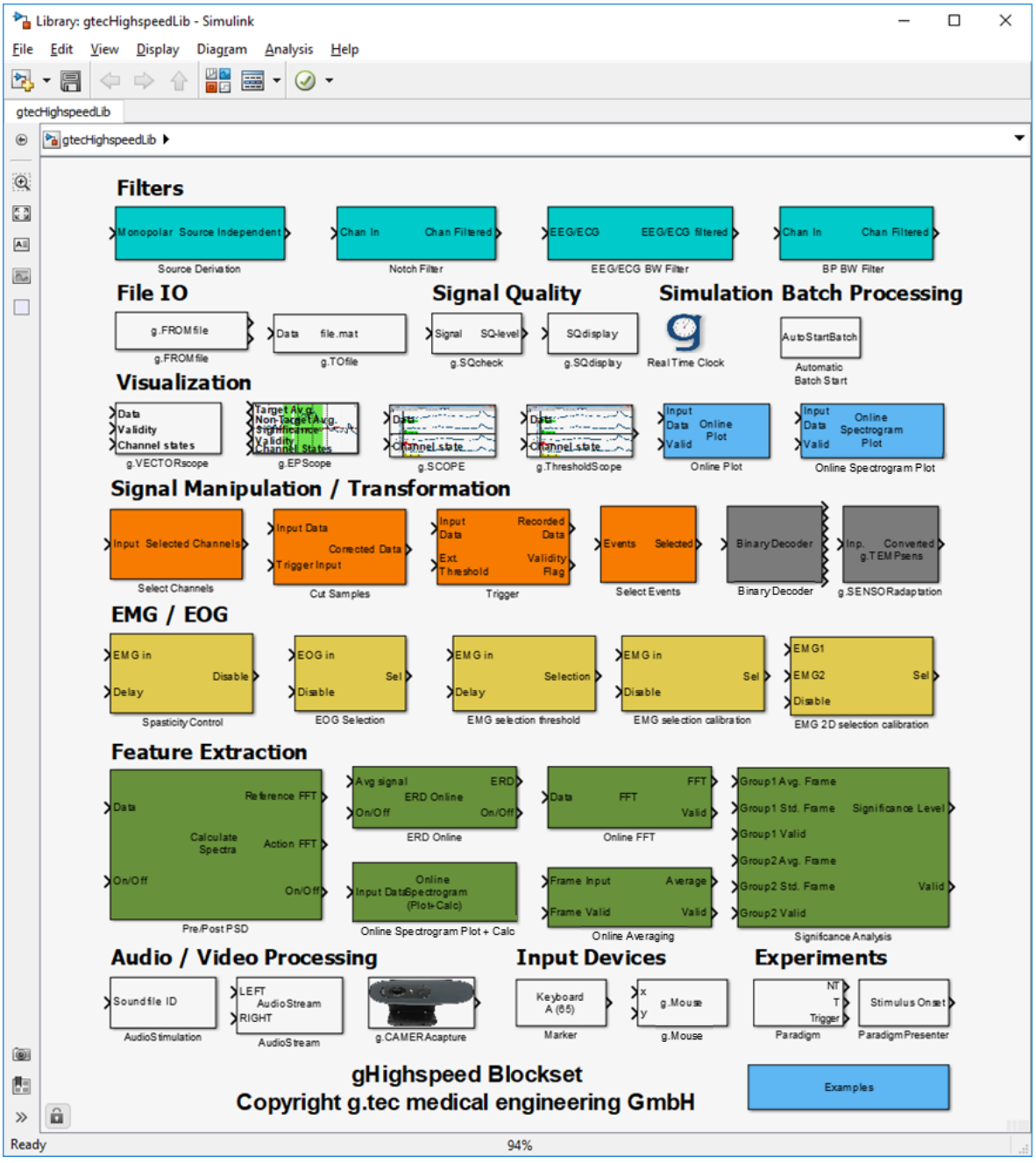
`..\gtecHS\Help`	Contains library block description and manuals of amplifiers
`..\gtecHS\Lib`	Contains binaries of library blocks
`..\gtecHS\EEGlab importer`	Contains functions for importing g.tec files to EEGlab

The users documents folder contains examples for the library blocks.

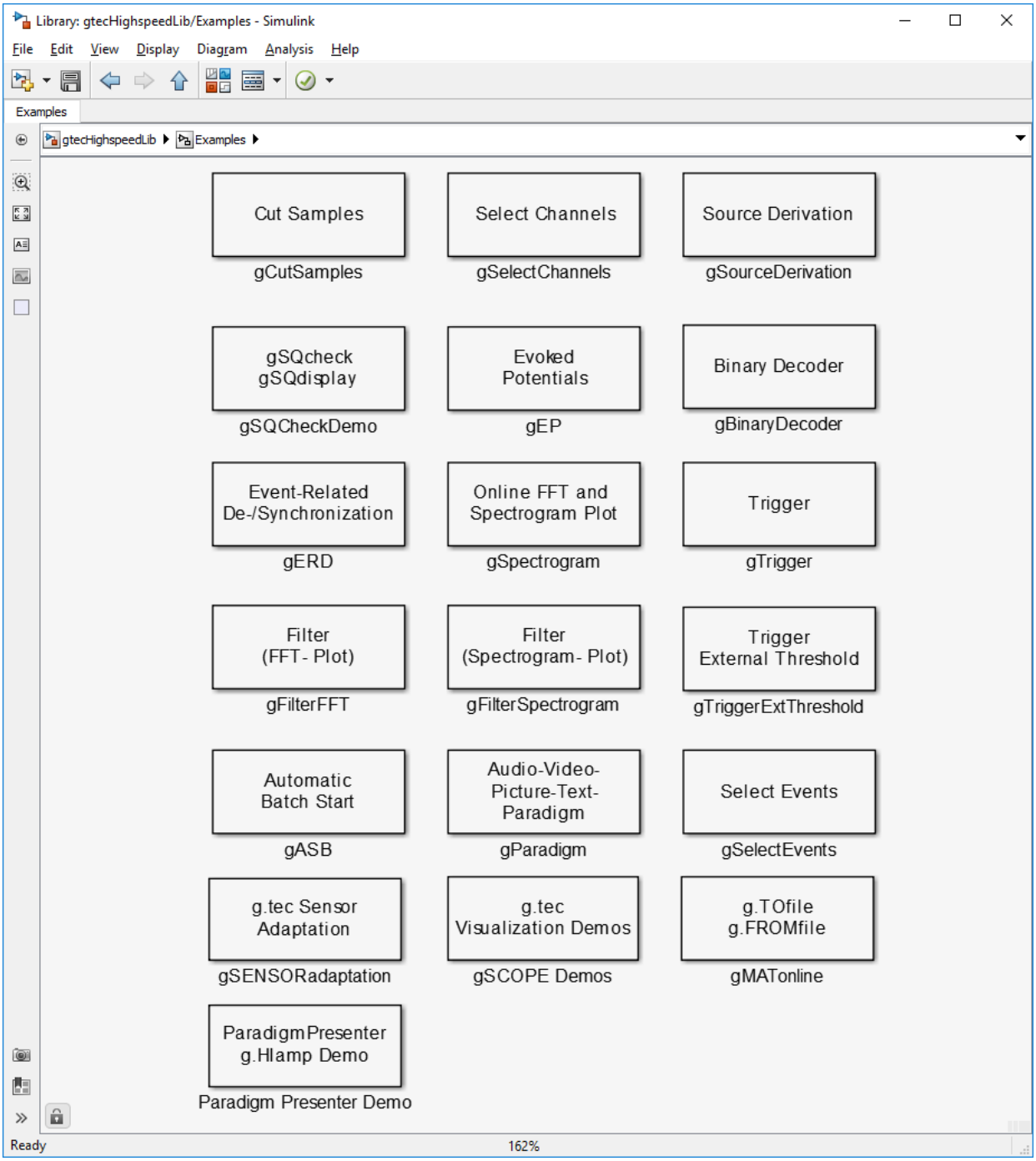
`..gtec\gtecHS\Examples`	Contains Simulink example models and data for the library blocks

g.tec Highspeed Library

The g.tec Highspeed Library contains the following blocks:



The Examples block contains links to all Simulink models referenced in this manual. The name of each block corresponds to the Simulink model, and the text displayed on the block names the blocks demonstrated by that model.



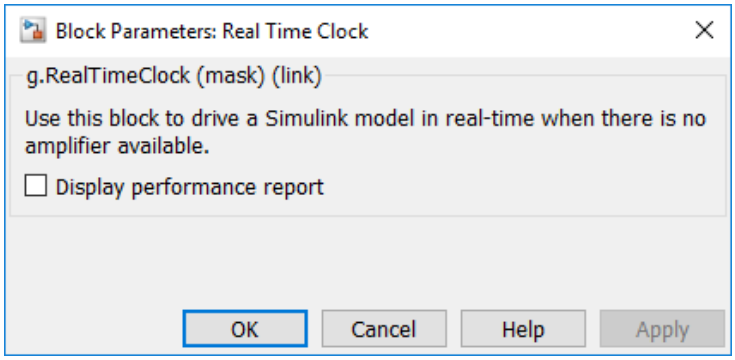
Real Time Clock

Description Real Time Clock can be used if no block is in a Simulink model that is able to drive it in real time (such as g.USBamp, g.HIamp, g.STIMbox in driving mode).

When the checkbox **Display performance report** is checked, the Real Time Clock will print a summary performance report on the MATLAB command line. This report includes the time simulated, the wall clock time elapsed, the difference between simulated and elapsed time and the time the Real Time Clock spend on idling while waiting for the next sample time to be reached.

NOTE: If there is a g.tec amplifier in the Simulink model, do **NOT** use Real Time Clock block.
Your model must include at least one source block explicitly defining a sample time > 0. Alternatively the fundamental sample time may be defined along with the solver settings. Solver must be fixed step type in this case.

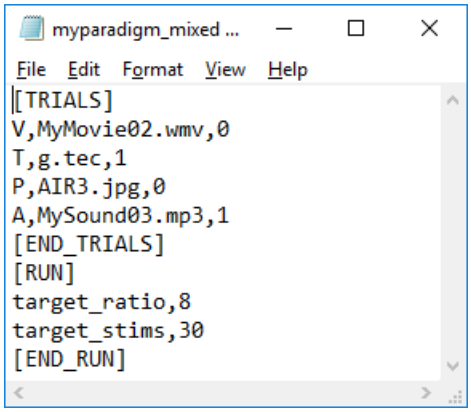
Dialog Box



Paradigm

Description The **Paradigm** block can be used for audio, picture, video and text paradigms. It uses the ActiveX plugin of the VLC media player, which must be installed to present audio, picture and video stimulations.
The block provides timing information of targets, non-targets and stimulation onset.
The paradigm is configured using a standard text editor.

Input: The paradigm block is configured with a *.txt file which specifies timing and stimulation constraints.



Paradigm File Description:

Trials- Section: Use tags [TRIAL] and [END_TRIALS] to specify the section of trial definition. Each trial is specified by stimulation type, stimulation context and stimulation processing (target or non-target).
<STIM_TYPE>, <STIM_CTX>, <STIM_PROCESS>

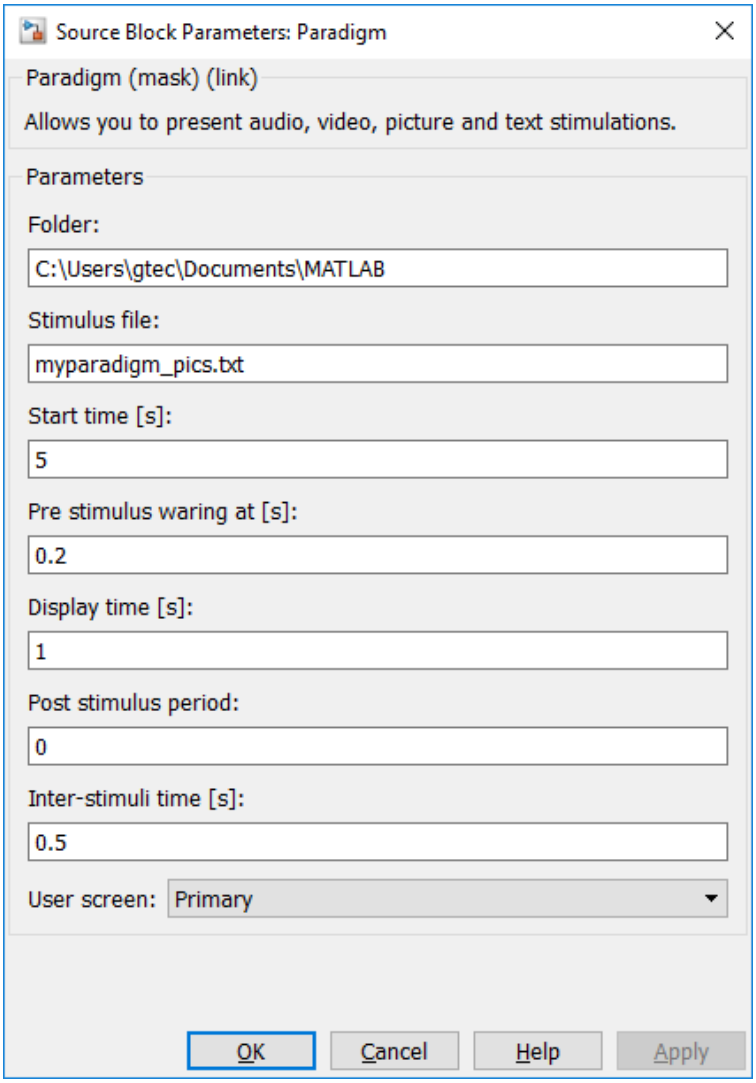
STIM_TYPE	V	Video stimulation identifier
	P	Picture stimulation identifier
	T	Text stimulation identifier
	A	Audio stimulation identifier
STIM_CTX	Video	Filename of Video to present (mp4, avi, mpeg, wmv supported)
	Picture	Filename of Picture to present (jpg, png, bmp, tif, tiff, jpeg supported)
	Text	Text to present
	Audio	Filename of Audio to present (mp3, wav supported)

Run- Section: Use tags [RUN] and [END_RUN] to specify the section of run definition. You can specify the target ratio (1 out of x stimulations is a target) and the number of targets to be presented.

target_ratio	Specify the target presentation ratio (1 out of target_ratio stimulations should be a target)
target_stim	Specify the number of target stimulations which should be presented

Dialog Box

Double click Paradigm block to modify the parameters.



Folder	Specify the path where paradigm files and stimulation context is located. The path can be specified using an absolute path or a relative path relative to the current MATLAB workspace path.

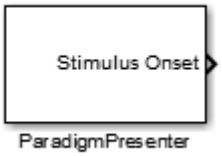
Stimulus file	Specify the filename of the paradigm file, which is used for stimulation
Start time	Specify the time in seconds before the paradigm creates the first stimulation
Pre stimulus warning at	Specify the time before a stimulation to create an attention warning
Display time	Specify the time the stimulation is presented (audio and video will be cropped)
Post stimulus period	Specify the time the stimulation trigger is active after the stimulation (indicated by a value of 1)
Inter-stimuli time	Specify the time between 2 stimulations
User screen	Specify the screen (primary or secondary) where the presentation window should be opened

Example Simulink Model	gParadigmDemo.slx

ATTENTION: This block requires VLC media player 64- bit version 2.2.1 to be installed on the computer.
In the paradigm description file make sure that there are no spaces or tabs at the end of the line.

ParadigmPresenter

Overview



Description The ParadigmPresenter block allows the playback of a paradigm defined in an XML file. It features audio, digital, picture, text, and video stimuli. See the respective manual "Paradigm Schema and Definition" for more details about paradigm definition.

Output The output signal "Stimulus Onset" is usually zero, and it delivers an impulse of length 1 sample each time a task switch occurs. The amplitude of this impulse evaluates to the numeric task ID of the respective task (see below for more information on numeric task IDs). The output is internally synchronized with the data of the amplifier block (g.HIamp, g.USBamp).

Note To ensure synchronous playback and data acquisition for time-critical paradigms, please observe the CPU load during model execution. If the CPU runs at high level, the task switches may jitter. However, the block is designed in a way such that the Stimulus Onset output signal reflects the *actual* task switch and not the nominal task switch. This allows a narrow specification of the stimulus arrival limits.

Dialog Box

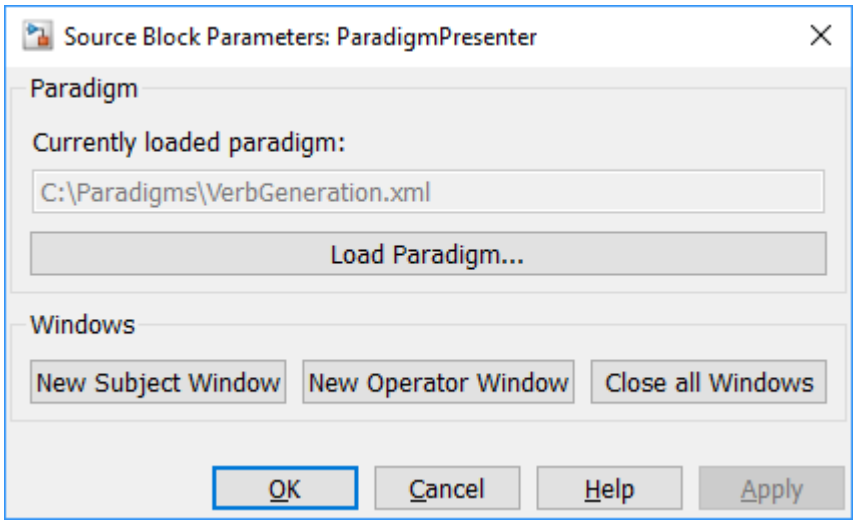


Figure 1. ParadigmPresenter Simulink mask.

Currently loaded paradigm	Shows the absolute path of the currently loaded paradigm XML file.
Load Paradigm…	Opens a file open dialog box to choose a valid XML paradigm file.

New Subject Window	Opens a new subject window (see below).
New Operator Window	Opens a new operator window (see below).
Close all Windows	Closes all windows.

Windows

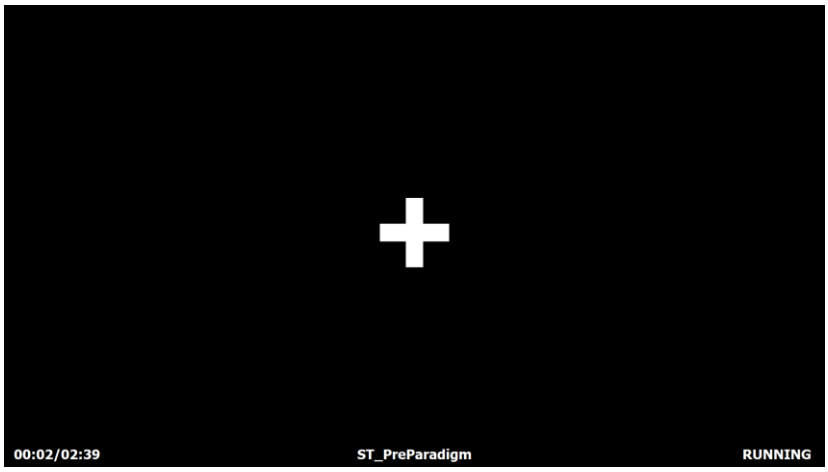


Figure 2. Exemplary operator window during playback.

Subject Window A subject window is completely empty and just displays the visual stimuli. Per default, on startup one subject window is opened.

Operator Window An operator window additionally shows information about the current time, total playback time, currently active task, and current ParadigmPresenter state (cf. Figure 2).

Interaction ParadigmPresenter windows feature the following user interaction:

- Move via mouse drag
- Resize via dragging the window corners
- Toggle fullscreen/window mode via double click
- Context menu via right mouse click:
  - Close Window (Hotkey: Ctrl+W)
  - New Subject Window (Hotkey: Ctrl+N)
  - New Operator Window (Hotkey: Ctrl+O)

Task Info Table

Each paradigm task is internally assigned a unique numerical ID. This ID is persistent over repetitive playbacks of the same paradigm. The task info table allows associating this somewhat arbitrary number with more meaningful task information assigned in the paradigm XML file (such as task ID, task label, group ID). This table is generated once a paradigm is successfully loaded. It is written to the MATLAB workspace (variable *taskInfo*) and printed to the console.

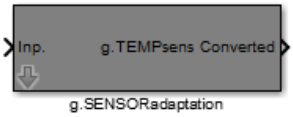
NumId	StringId	Label	Group
1	'ST_PreParadigm'	'Start'	0
2	'ST_Bsl1'	'Baseline'	1
3	'ST_Act1'	'Action'	2
4	'ST_Bsl2'	'Baseline'	1
5	'ST_Act2'	'Action'	2
6	'ST_Bsl3'	'Baseline'	1
7	'ST_Act3'	'Action'	2
8	'ST_Bsl4'	'Baseline'	1
9	'ST_Act4'	'Action'	2
10	'ST_PostParadigm'	'Finish'	0

Code Listing 1. Exemplary task info table.

For detailed information concerning paradigm definition, please read **ParadigmSchemaAndDefinition.pdf**.

ATTENTION: This block requires ASIO4ALL – Universal ASIO Driver For WDM Audio version 2.12 or later to be installed on the computer.

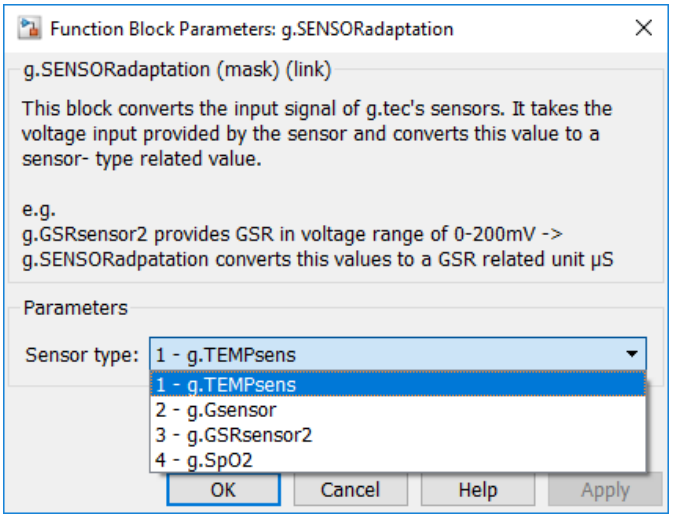
g.SENSORadaptation



The **g.SENSORadaptation** block is for converting input data from a g.tec sensor (such as a temperature sensor, G sensor, GSR sensor and SpO2 sensor) into meaningful outputs according to the sensor.

Description This block converts the input signal of g.tec's sensors. It takes the voltage input provided by the sensor and converts it to sensor- type related values.

Dialog Box

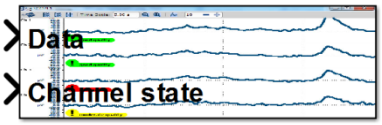


Double click on the **g.SENSORadaptation** block to edit the parameters

Sensor type — Select the sensor type the adaption should be used for from the provided list. Currently g.TEMPsensor, g.Gsensor, g.GSRsensor2 and g.SpO2 sensors are supported.

Example Simulink Model — gSENSORadaptation_Demo.slx

g.SCOPE



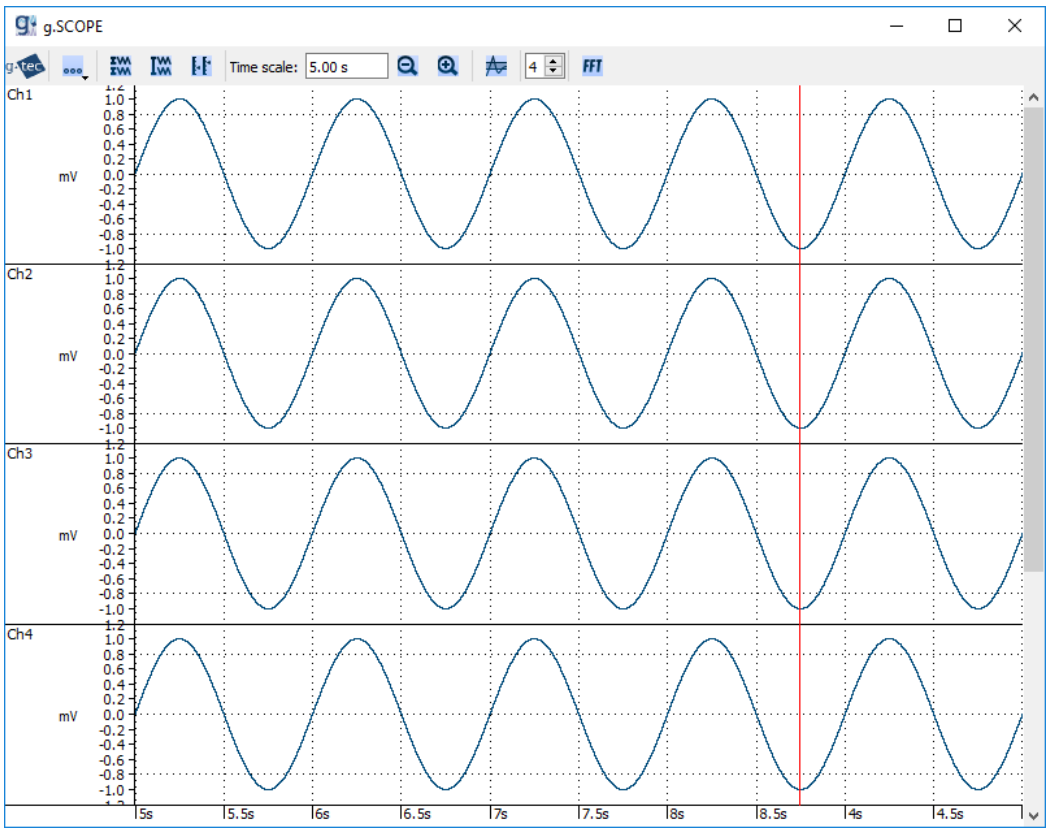
g.SCOPE

The **g.SCOPE** block provides a convenient way to display biosignal data. It was designed with a focus on performance and is able to visualize large amounts of data with high sampling rates.

Description g.SCOPE is a sample-based scope for high sampling rates and a large amount of channels, with easy-to-use scaling facilities. It offers a spectrum analysis panel which shows FFT of the incoming channels.

Input The data to be displayed is fed into the block using the **Data** input port. It can either receive sample-based data or frame-based data.
An additional input port named **Channel states** can be used to display additional information for each channel (see description of channel states below). This input accepts sample-based signals. If the Channel states input is used, its number of elements has to be equal to the number of channels of the Data input signal.
Make sure that the sample time of the input signals is the same.
The input format can be single (float32) or double. Use a Multiplexer block to input multiple channels.

g.SCOPE



1) Double click the **g.SCOPE** block to open the scope.

The toolbar of the **g.SCOPE** window provides access to the following actions.

g.SCOPE configuration menu, contains
- Edit channel states
- Import channel names
- Edit color settings
- Store screenshot
- Load / Save configuration

Auto-scales all the channels individually to their minimum and maximum data value (plus 10%).

Auto-scales all channels to the global minimum and maximum level of the incoming data (plus 10%).

Opens the channel settings dialog.

Time Scale

Sets the time interval displayed by each scope. Allowed units are h (hours), m/min (minutes), s/sec (seconds), ms (milliseconds) and smp (samples). If no unit is specified, the entered value is interpreted as seconds.

Button for increasing the displayed time interval (i.e. zooming out).

Button for decreasing the displayed time interval (i.e. zooming in).

Switch signal cropping on or off. If signal cropping is on, which is the default setting, then the signal will be cropped as soon as it exceeds the vertical limits of the display. If it is switched off, all data will be displayed overlapping the displays of the adjacent channels. Examples are given below.

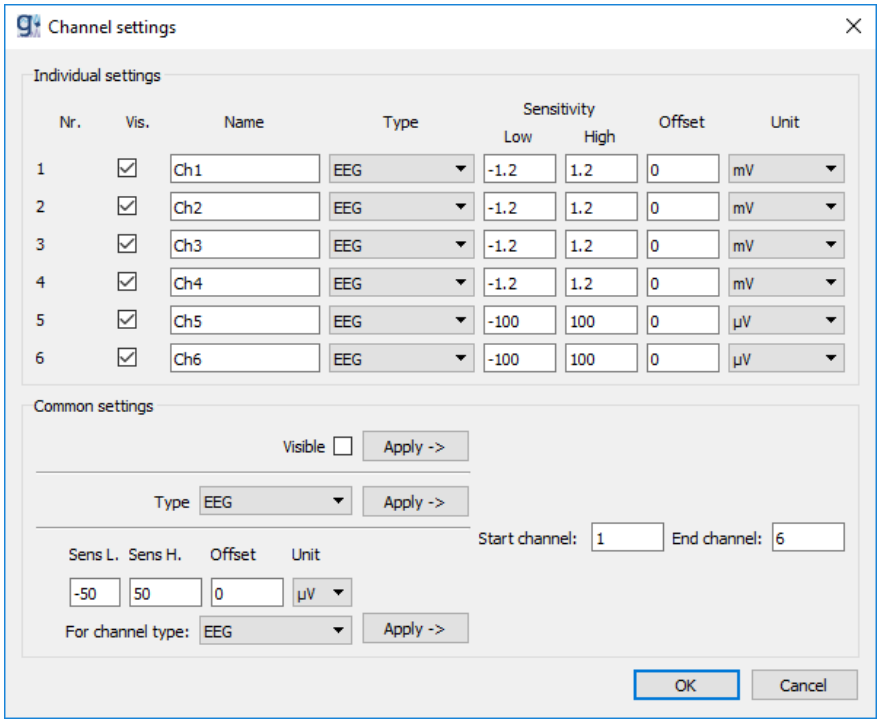
Opens the spectrum analysis panel.

 # visible Channels

Spin button to increase and decrease the number of visible channels.

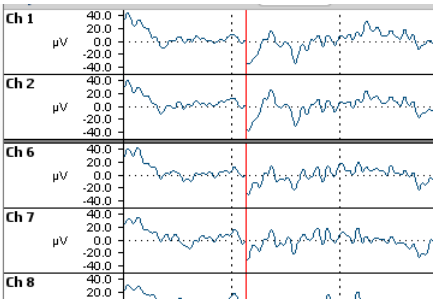
Make **Quick FFT** panel visible

2) Open the **Channel settings** dialog by clicking the  button to define the scaling and offset for each channel individually.



Nr The number of the channel.

Vis Sets the visibility of the channel. If not checked the channel is replaced by a grey bar.



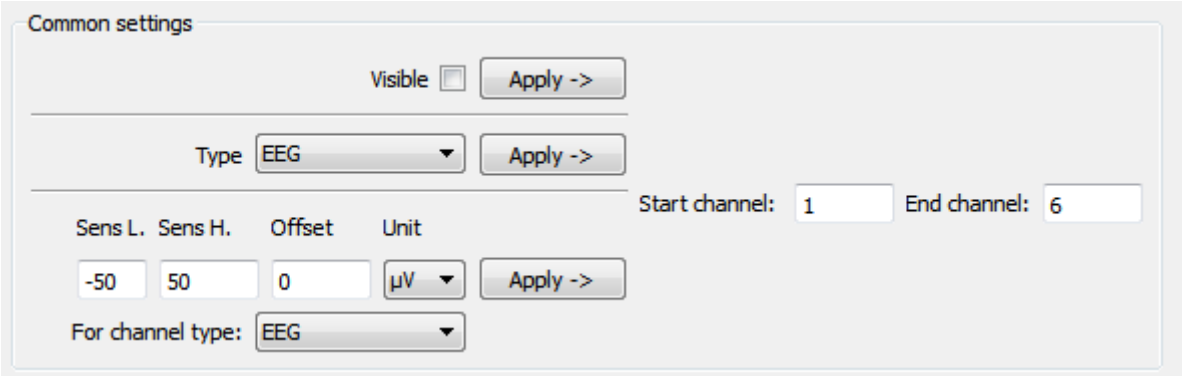
Name The channel name.

Type The signal type. The following channel types with appropriate default values are predefined:

Type	Low	High
EEG:	-50 µV	50 µV
ECG:	-1 mv,	1 mV
EOG:	-500 µV	500 µV
EMG:	-5 mV	5 mV
ECoG:	-500 µV	500 µV
Temperature	20 °C	45 °C
Acceleration	-3 G	3 G
Saturation	0 %	100 %
GSR / EDA	0 µS / µMho	30 µS / µMho
Resp.:	-500 µV	500 µV
TRIG:	0 V	5 V
OTHER:	-100 µV	100 µV

Low The lower limit of the signal display.
High The upper limit of the signal display.
Offset The vertical offset of the displayed signal.
Unit The unit of the selected **Low** and **High** limits .

3) In the **Common settings** frame set the **Start channel** to `1` and the **End channel** to `6`.



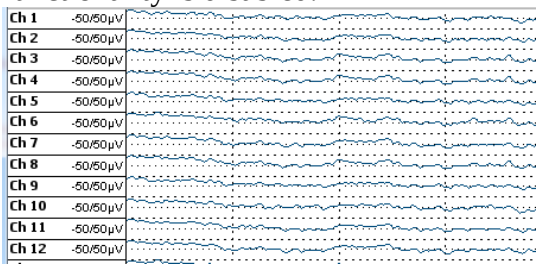
4) Check **Visible** in the **Common settings** frame and press **Apply**.

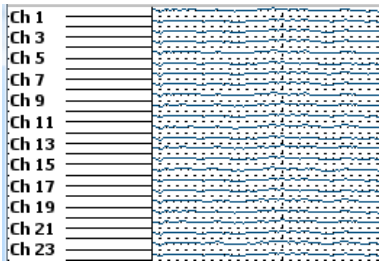
5) Select `EEG` as signal **Type** and press **Apply**.

6) Set the **Offset** to be applied to all channels to `0` and the **Unit** to `µV` and press **Apply**.

7) Press **OK** to apply the settings.

If a high number of channels are displayed in one scope and the height of the scope is limited, the layout of the axes is changed for better visibility. In the compact display mode, the dragging functionality is disabled.



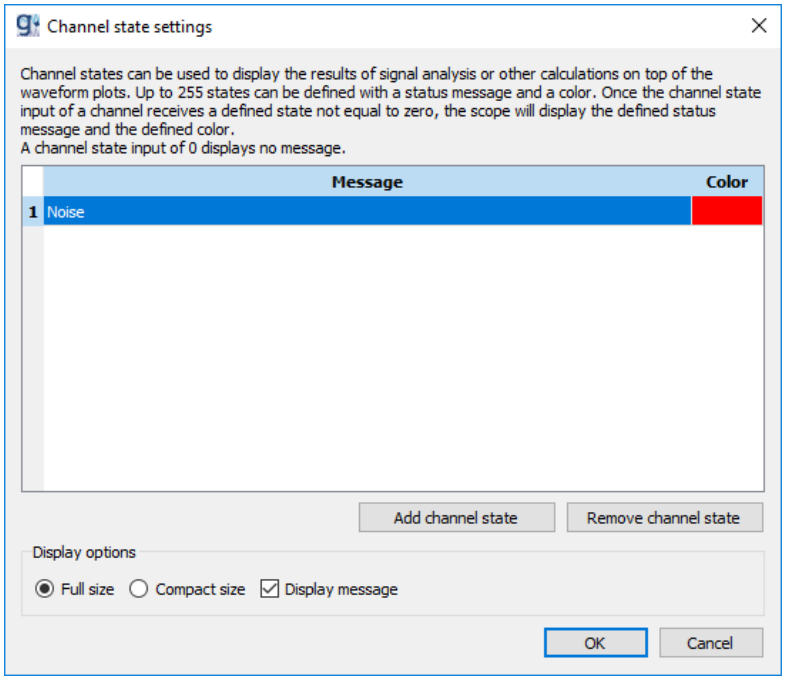


Channel states:

1) Attach the output of the **g.SQcheck** block to the **Channel states** port of the **g.SCOPE** block

2) Double-click the **g.SCOPE** block.

3) Open the g.SCOPE configuration menu and select **Edit channel states** to define the possible states



4) Click **Add channel state** to add the first state.

5) Set **Message** to `Noise` for the first state.

6) Click on the **Color** display left of **Message** and select the `red` color for this state.

7) Define a second state whenever a `High amplitude` is detected displayed in `yellow` **Color**.

8) Define a third state for the case that the signal is of `Good signal quality` use the `green` **Color** for this state.

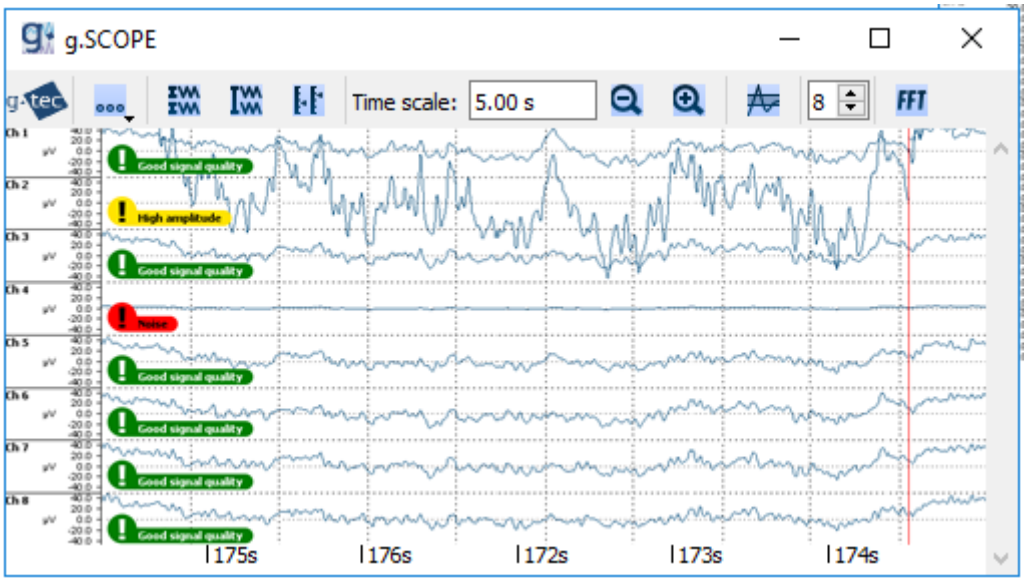
9) To remove a state, select it and press **Remove channel state.**

10) Define the outline of the channel state icon by choosing the Display option:
    Full size: show icon in channel state
    Compact size: do not show icon in channel state
    Display message: enable this option to show the message

11) Press **OK** to apply the new states.

12) Start the model.

13) Double click the **g.SCOPE** block.



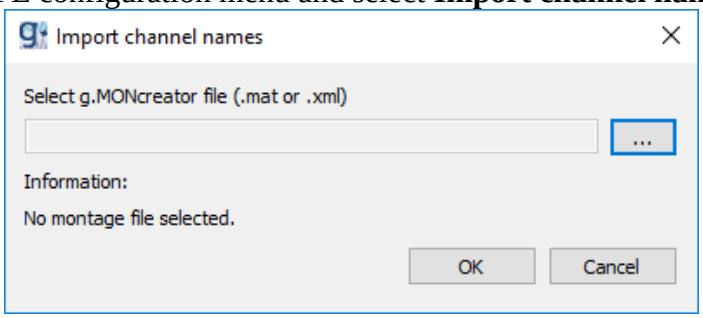
Channel **Ch 2** in the above example shows atypical high amplitude, and therefore is marked with the yellow message. Channel four has very low amplitude and is therefore marked red, while the amplitudes of all other signals are within range. Therefore, they are marked green.

Import channel names

g.SCOPE allows you to import channel names from montages created by g.MONcreator.

1) Double-click the **g.SCOPE** block.

2) Open the g.SCOPE configuration menu and select **Import channel names**.



3) Click  to browse to the montage file you want to import the channel names from.

4) Check the **Information** section to see whether the loaded montage is valid.

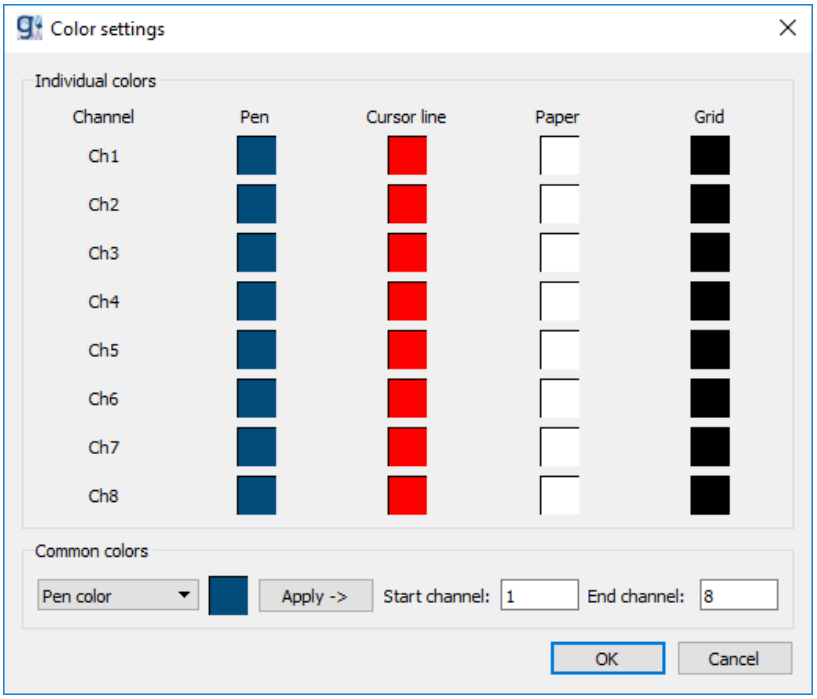
5) Click **OK** to apply channel names.

Color settings

For you convenience, g.SCOPE allows you to customize the colors of the paper, grid, pen and the cursor lines.

1) Double-click the **g.SCOPE** block.

2) Open the g.SCOPE configuration menu and select **Edit color settings**.



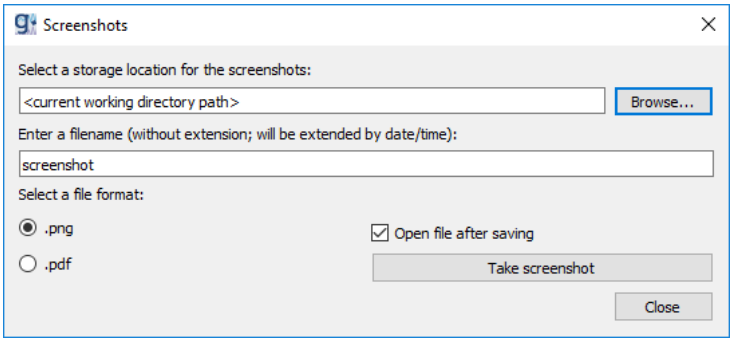
3) Click the color of one channel and item to open the color picker dialog. To specify the color for a range of channels, use the **Common colors** panel.

Store screenshot

g.SCOPE allows you to take a screenshot of the current state of the scope and export it as *.png or *.pdf file.

1) Double-click the **g.SCOPE** block.

2) Open the g.SCOPE configuration menu and select **Store screenshot**



3) You can specify a location where to store the screenshot via the **Browse** button or by editing the path manually. The default value is the current working directory of the Simulink model.

4) Enter a filename for the screenshot. The default value is *screenshot*

5) Select the output file format which can be **.png** or **.pdf**

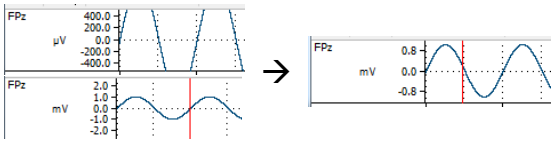
6) Specify whether the created screenshot should be opened (using the default program specified for the given file format) after the screenshot is taken or not by checking the **Open file after saving** check box.

7) Clicking the **Take screenshot** button will create a file with the selected settings. The filename will be extended by a post-fix string of the format '_dd_mm_yyyy_hh_mm_ss'. Each second you can produce one screenshot.

Mouse actions

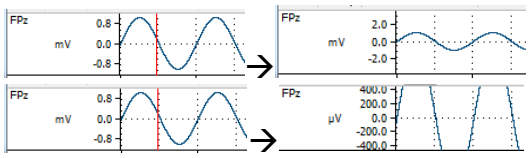
The axis configuration functions auto-scale, scaling and offset can be controlled using the mouse.

Auto-scale



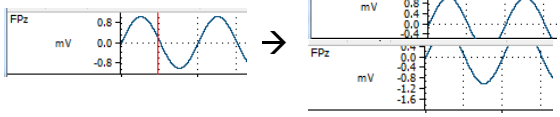
Double click the channel label area to auto-scale the channel to its minimum and maximum values.

Scaling



Right click the channel vertical (voltage) axis and drag towards 0 to zoom out drag away from 0 to zoom in.

Offset

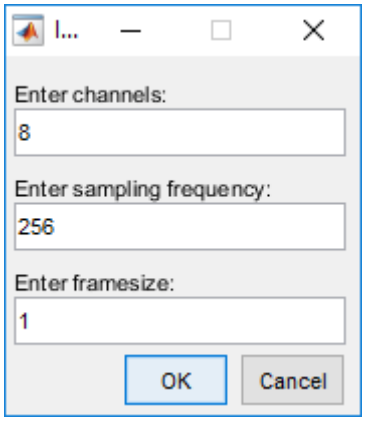


Left click the channel vertical (voltage) axis to shift the signal up or down.

Configuration of g.SCOPE

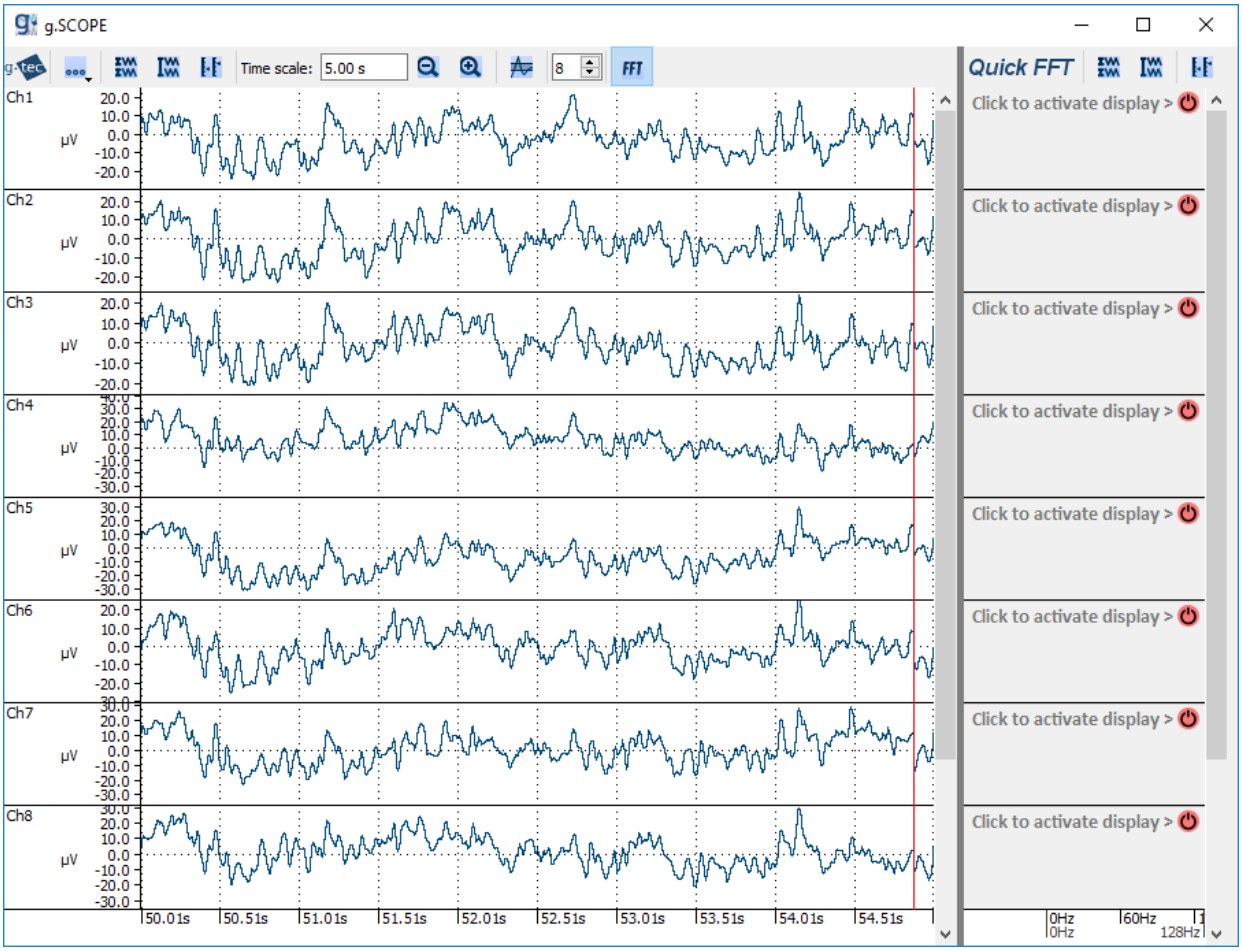
g.SCOPE allows to save and load the configuration files. To do so, open g.SCOPE and select **Load configuration** / **Save configuration** from the menu bar. A file dialog opens which allows to load or a save the g.SCOPE configuration file.

The configuration of g.SCOPE is only possible if the number of channels, sampling frequency and frame size is known. Before g.SCOPE opens, the block tries to compile the Simulink model. If the Simulink model can be compiled the needed information is retrieved automatically. If the model is not compiling, a dialog box opens, which prompts for the needed information (number of channels, frame size and sampling frequency).

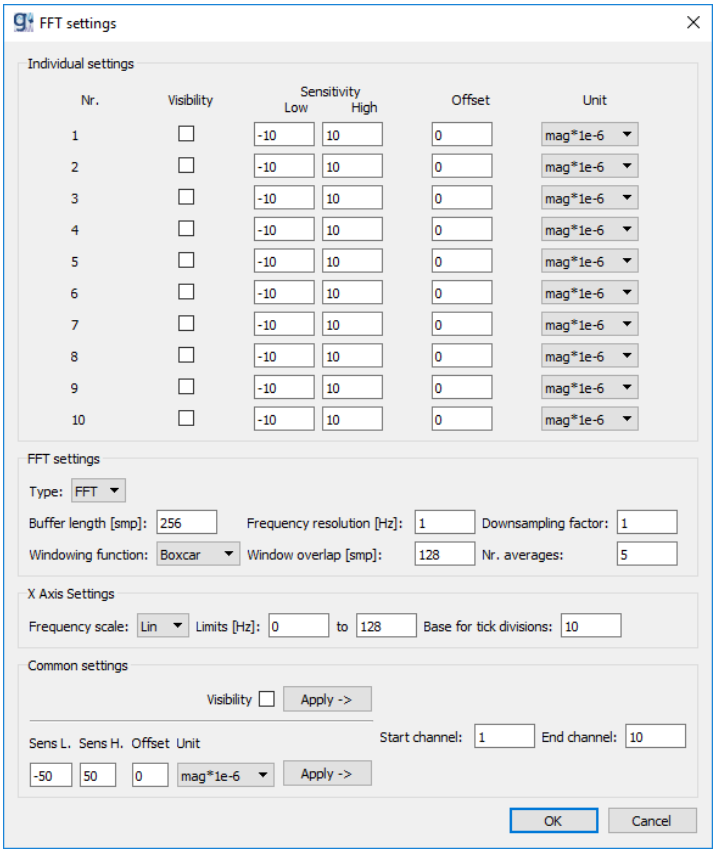


FFT panel of g.SCOPE

As described above, you can make the **Quick FFT** panel visible by clicking the FFT button in the toolbar.



The QuickFFT panel comes with 3 toolbar items. The first 2 items are for "Per Channel" and "Global" auto-scale. The 3rd item will open the FFT settings dialog:



This dialog allows you to configure the Quick FFT up to your needs.

Type 3 different FFT types are available:
- PSD (Power Spectral Density)
- PS (Power Spectrum)
- FFT (Fast- Fourier Transform)

Buffer length Specifies how many samples from the incoming data stream are used to create a FFT frame and are used to calculate the FFT. This value changes automatically, if the **Frequency resolution** parameter changes.

Frequency resolution Specifies the frequency resolution for each sample in the resulting FFT length (buffer length / 2).
This value is automatically updated when the **Buffer length** parameter changes.

Downsample factor Specifies for values > 1 that a down- sampling should be performed before calculating the FFT.

Windowing Function Specifies the windowing function which is applied to the FFT frame before calculating the FFT. Available functions are
- **Boxcar** (no window function)
- **Hanning**
- **Hamming**

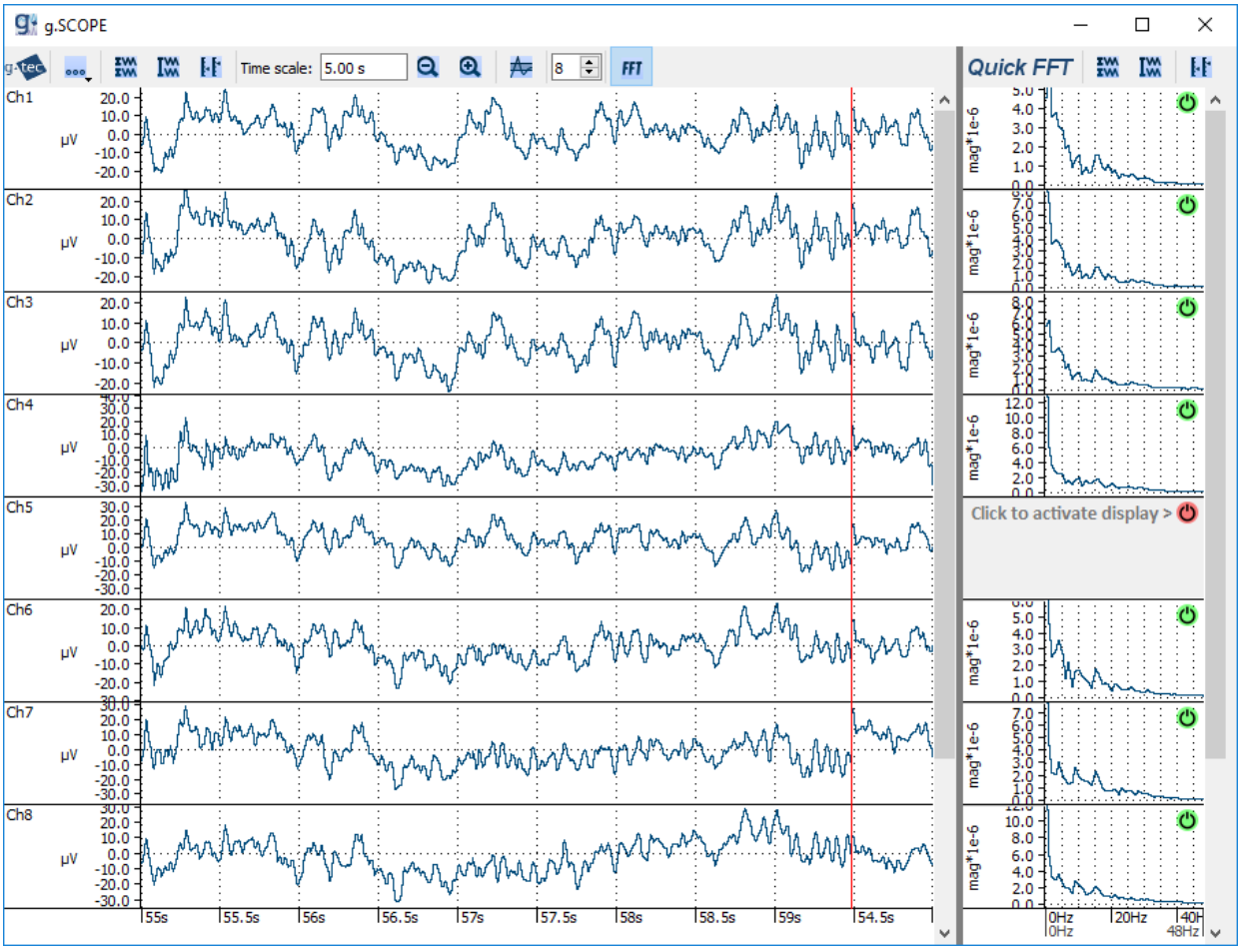
Window overlap Specifies the number of samples the FFT frames are overlapping.

Nr. Averages	Specifies how many FFT frames are averaged before calculating the FFT.
Frequency scale	Sets the type of the frequency scale - Lin → linear scaling - Log → logarithmic scaling -
Limits	Set the lower and upper limits of the frequency scale.
Base for tick division	Specifies the base of the frequency scale.

There are 3 ways to activate / deactivate a channel for FFT calculation

1. use the **Visibility** check box for each channel to set it active.
2. use the **Visibility** check box in the *Common Settings* to set a range of channels specified by **Start channel** and **End channel**
3. use the On / Off (/) buttons in the Quick FFT panel to activate / deactivate a channel.

Once you have activated a channel for calculation and a FFT frame was formed according to the FFT settings, the FFT result will be visualized in the Quick FFT panel.



As for g.VECTORscope and g.EPscope, you can zoom in to the resulting FFT result by dragging a range. Reset the zoom area by double clicking into the result FFT window.

You can adjust the size of the Quick FFT panel by hovering the border of Scope panel and Quick FFT panel. The mouse arrow will change to  indicating that the size is able to change. Drag the border to either direction (within minmum size constraints).

g.THRESHOLDscope

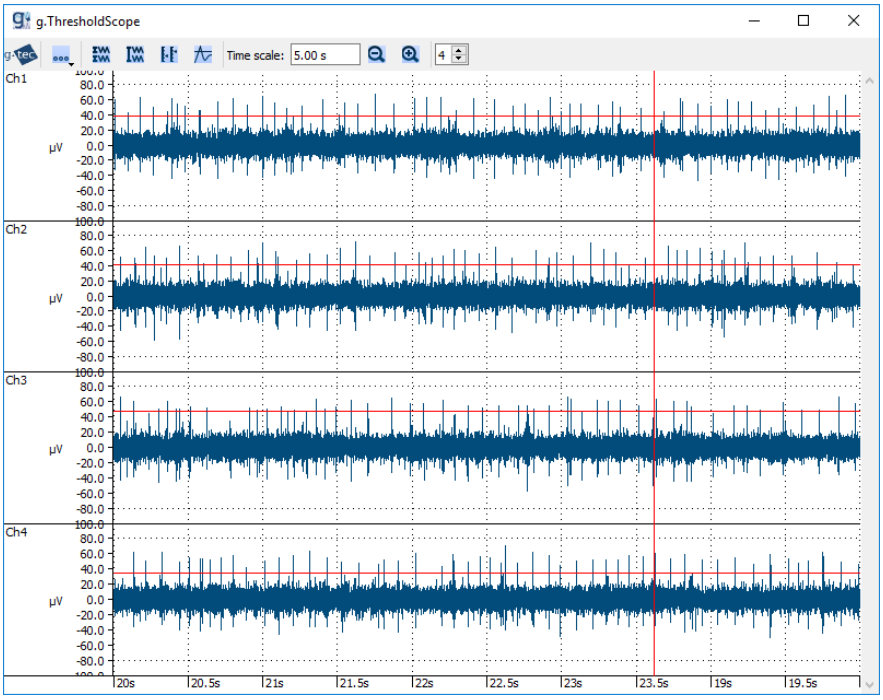
The **g.THRESHOLDscope** block provides a convenient way to display biosignal data and generate trigger information for the Trigger block.

Description g.THRESHOLDscope is a sample-based scope with easy-to-use threshold trigger facilities.

Input The data to be displayed is fed into the block using the **Data** input port. It can either receive sample-based data or frame-based data.
An additional input port named **Channel states** can be used to display additional information for each channel (see description of channel states in g.SCOPE description). This input accepts sample-based signals. If the Channel states input is used, its number of elements has to be equal to the number of channels of the Data input signal.
Make sure that the sample time of the input signals is the same.
The input format can be single (float32) or double. Use a Multiplexer block to input multiple channels.

Output The **Threshold** output port holds a threshold value for each channel. If the threshold is activated, a threshold value according to the horizontal bar location is generated. If one channel's threshold is not activated, *NaN* values will be output.

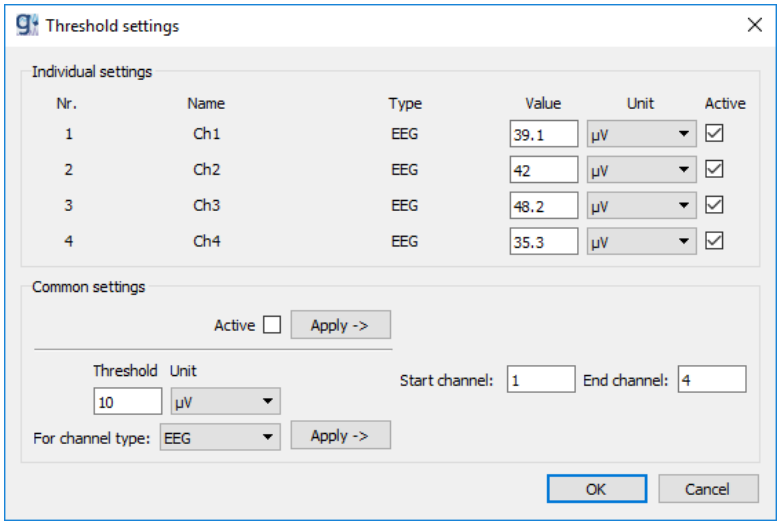
g.THRESHOLDscope

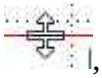


Double click the **g.THRESHOLDscope** block to open the scope.

The figure shows a g.THRESHOLDscope with 3 channels. The threshold feature is activated for channels 1 and 3. The threshold output holds a value of approx. 70 µV for channel 1 and a threshold value of approx. 40 µV for channel 3. Due to channel 2 is not activated for generating threshold information, the corresponding threshold channel holds *NaN* value. This threshold information can be used to trigger the data as soon as the data drops below or rise above the specified threshold. Use g.tec's Trigger block to trigger data.

Threshold- Scope configuration
Open the threshold settings by clicking the threshold icon in the toolbar. 



Once you have activated the threshold feature by checking the **Active** checkbox of the single channels there are 2 ways to set the threshold. You can either set the value of the threshold to an exact value within this configuration dialog by entering the threshold value in the edit box of the desired channel or you can drag the horizontal bar displayed in the channel up and down using the mouse. Once you hover the displayed threshold line the mouse changes to a move arrow  , indicating that the line can be dragged up and down now.
Use the **Common settings** in the Threshold settings dialog to apply settings of multiple channels.

NOTE: For detailed information on basic scope configuration, please read the g.SCOPE description (pg. 17ff)

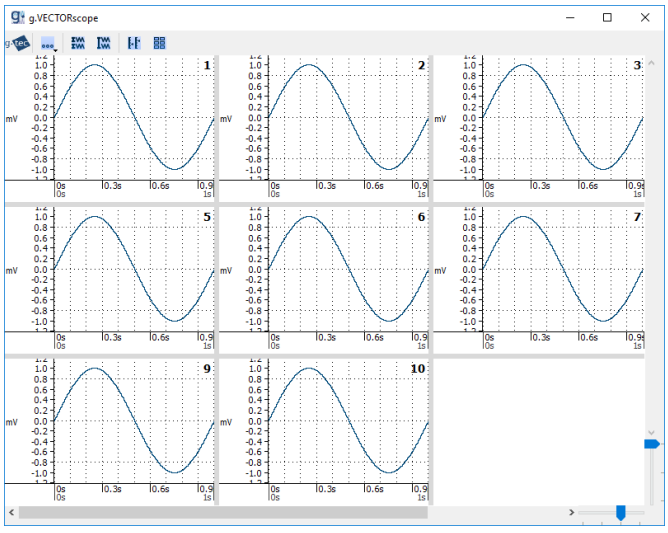
g.VECTORscope

The g.VECTORscope provides a convenient way to display frame-based data. It is designed with a focus on performance and is able to visualize large amounts of data with high sampling rates.

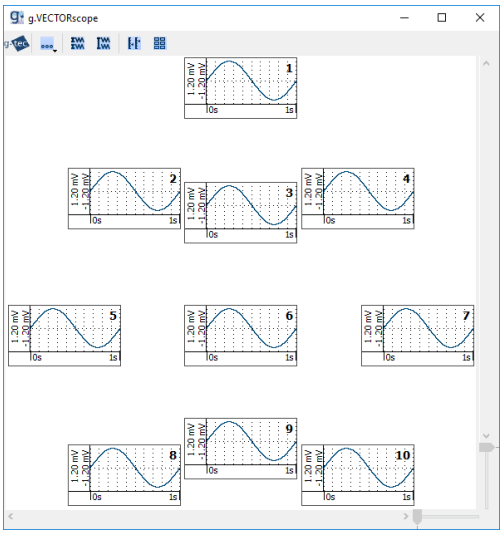
Description g.VECTORscope is a frame-based scope for high sampling rates and a large amount of channels, with easy-to-use scaling facilities.

Input The data to be displayed is fed into the block using the **Data and Validity** input ports. Data is frame-based while Validity is sample-based.
Beside frame based data for each channel g.VECTORscope can display a channel state value. Connect the channel state specific data to the **Channel States** input port and configure the channel states to activate the display.

g.VECTORscope

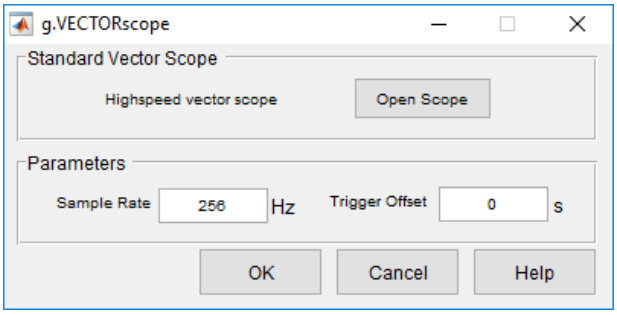


square channel arrangement



topographic channel arrangement

Parameters Double click on the **g.VECTORscope** block to open the scope parameters.



Sample Rate sample rate in Hz in which the sample based raw data is running.

Trigger Offset position of the vertical trigger occurrence line (in respect to the data frame which is visualized).

Open Scope click this button to open the scope.

Vector- Scope configuration

1) Double click the **g.VECTORscope** block. If model is running, then g.VECTORscope is opened. Otherwise click **Open Scope** to open g.VECTORscope.

The toolbar of the **g.VECTORscope** window provides access to the following actions.

g.VECTORscope configuration menu, contains
Edit channel states
Edit color settings

Auto-scales all the channels individually to their minimum and maximum data value (plus 10%).

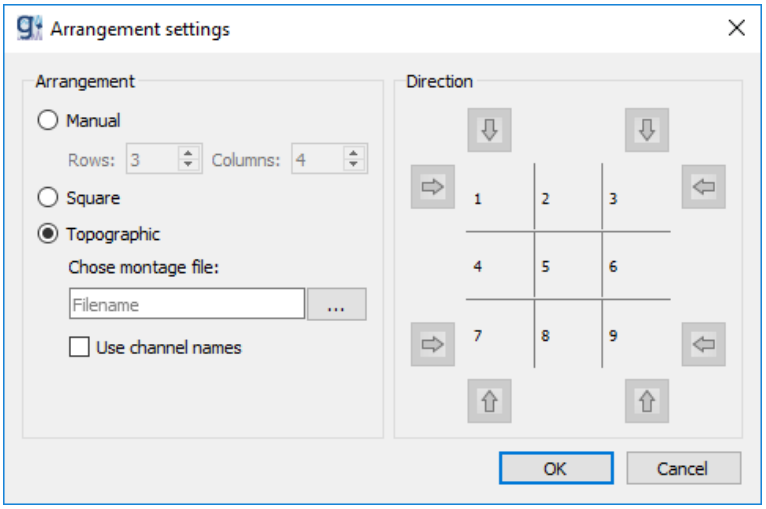
Auto-scales all channels to the global minimum and maximum level of the incoming data (plus 10%).

Opens the channel settings dialog.

Opens the channel **Arrangement settings** dialog.

2) Open the **Channel settings** dialog by clicking the  button to define the scaling and offset for each channel individually. For details, please see above.

3) Open the channel **Arrangement settings** dialog



Manual: select the number of rows and columns to arrangement the axes.

Square: auto-arrangement in square shape.

Topographic: use the g.tec montage file created by g.MONcreator to arrange channels topographically. Optionally the channel names can be imported.

Direction: use the arrow to change the order of the axes.

Zooming

g.VECTORscope allows you to zoom in to interesting data regions.

1) Press the left mouse button on an displayed axis.
2) Keep the mouse button pressed and select the range of the zooming by moving the mouse along the time axis → a zoom bar is shown.
3) Releasing the mouse button when you reached the end of the region will zoom all channels to the specified range.
4) Double click one axis will zoom out all channels to the 100 % view.

NOTE: For detailed information on color and channel state settings and mouse actions please read the g.SCOPE description (pg. 17ff)

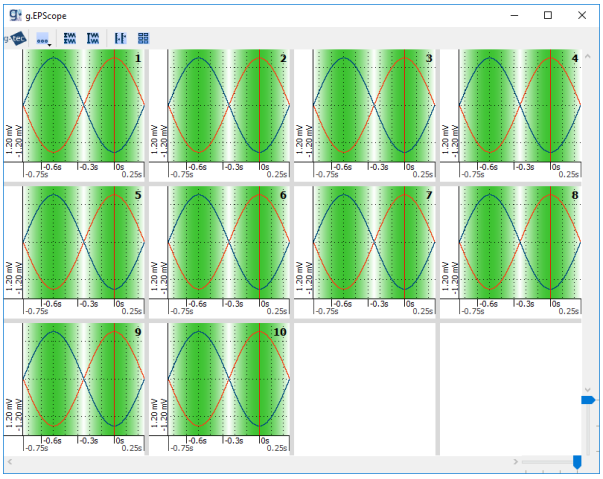
g.EPscope

The **g.EPscope** block provides a convenient way to display evoked potentials. It is designed with a focus on performance and is able to visualize large amounts of data with high sampling rates.

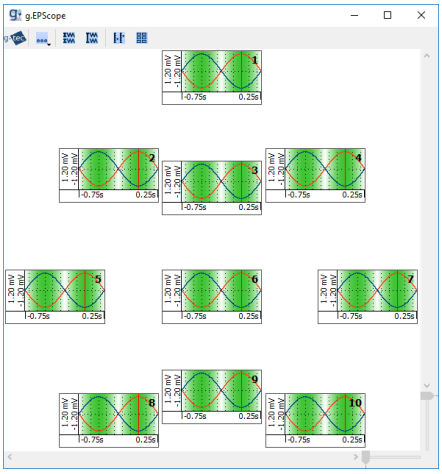
Description g.EPscope is a frame-based scope for high sampling rates and a large amount of channels, with easy-to-use scaling facilities.

Input The data to be displayed is fed into the block using the **Target Avg., Non-Target Avg., Significance and Validity** input ports. Data on Target Avg., Non-Target Avg. and Significance is frame-based while Validity is sample-based.
Target and Non-Target Averages for one channel are shown in the same channel axis that forms the EP. The result of the statistical analysis (see Statistical Analysis, Pg. 62) can also be shown in g.EPscope. Areas where the averages of target and non-target show statistically significant differences are indicated by a green background.
Beside target, non-target and significance each channel can display a channel state value. Connect the channel state specific data to the **Channel States** input port and configure the channel states to activate the display.

g.EPscope

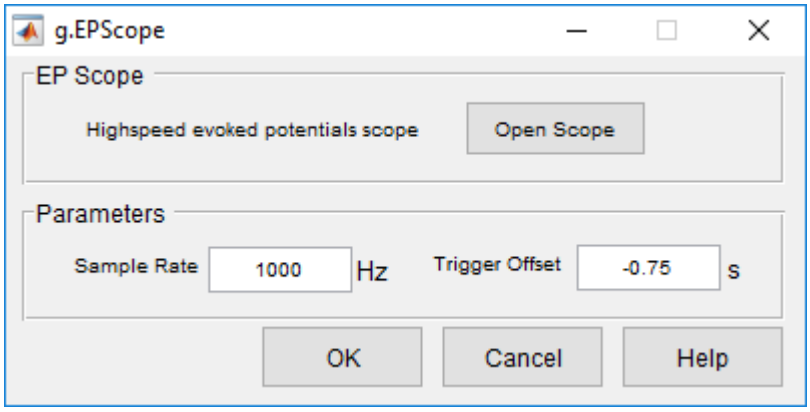


square channel arrangement



topographic channel arrangement

Parameters Double click on the **g.EPscope** block to open the scope parameters.



Sample Rate sample rate in Hz in which the sample based raw data is running

Trigger Offset position of the vertical trigger occurrence line (in respect to the data frame which is visualized)

Open Scope click this button to open the scope

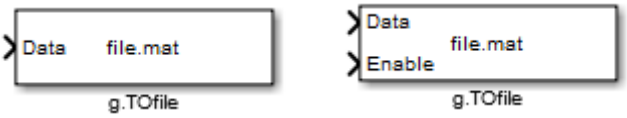
EP-scope configuration

Please check g.VECTORscope description (pg**.** 29ff)

NOTE: For detailed information on color and channel state settings and mouse actions please read the g.SCOPE description (pg. 17ff) for information on zooming please read the g.VECTORscope description (pg. 29ff).

g.MATfile

g.TOfile Block

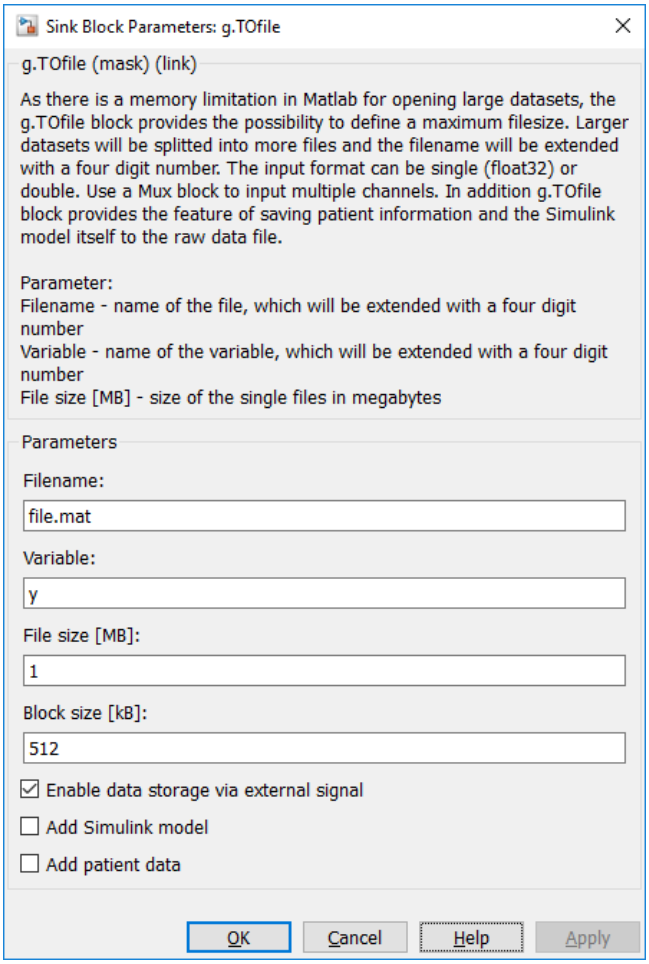


The **g.TOfile** block is for saving large datasets in real-time to files.

Description

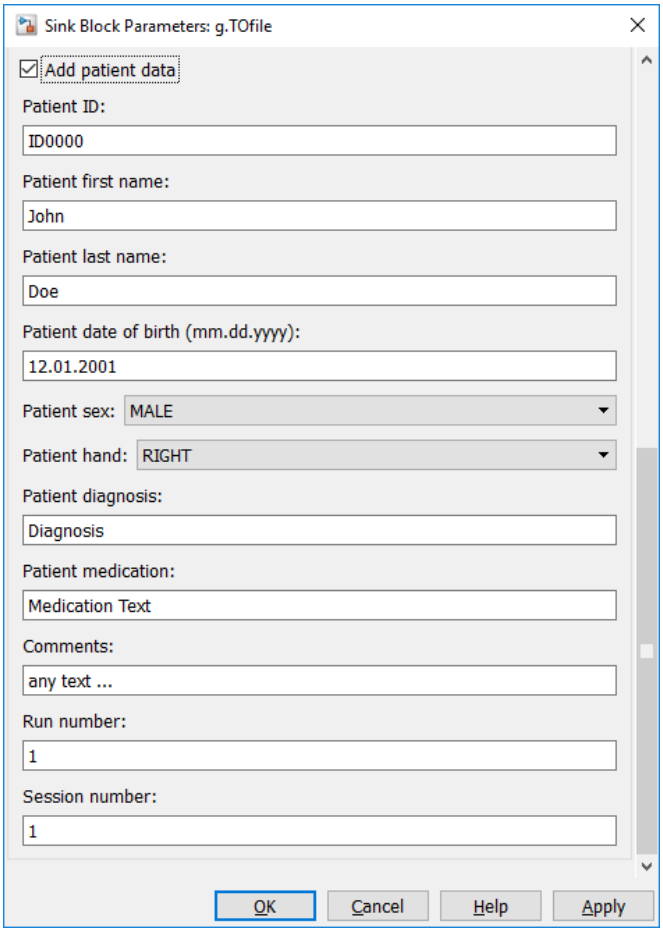
As there is a memory limitation in MATLAB for opening large datasets, the **g.TOfile** block provides the possibility to define a maximum file size. Larger datasets will be split into more files and the filename will be extended with a four digit number. The input data format is double. Use a **Mux** block to input multiple channels. The data storage can be controlled via the **Enable** input port. The file will be created as soon as there is data available and data storage is enabled. In addition to the raw data, the g.TOfile block can save both patient information and the Simulink model itself to the data file.

Dialog Box



Double click on the **g.TOfile** block to edit the parameters.

Filename	The filename will be extended automatically with a timestamp representing the start time of the recording in local system time and a four digit number, starting with 0000. The number is incremented for each file of the recording – the timestamp is updated every time a new recording is started. The filename has the following format: `<filename>_DD_MM_YYYY_HH_mm_SS_NNNN` DD day of month MM month of year YYYY year HH hour (24 hour format) mm minute SS seconds NNNN filenumber in sequence
Variable	data matrix name
File size [MB]	maximum size of one file in megabyte data exceeding this limit will be split up in more files *NOTE:* Use the **gBSmerge** available through the **Transform/Merge** from the **g.BSanalyze** menu to concatenate these files into a single data set
Block size [kB]	minimum size of data that is written to the hard disk
Enable data storage via external signal	Activate this feature to control data storage via an external input, only a value higher 0 at the added *Enable* input port will activate data storage
Add Simulink model	Activate this feature to store the Simulink model along with the raw data. The Simulink model is saved when the model finishes, and the model is stored in all files created by the g.TOfile block.
Add patient data	Activate this feature to store patient information along with the raw data. The patient information will be stored to all files please check available patient info fields in the next figure



NOTE: Patient information can also be retrieved from g.BSanalyze.

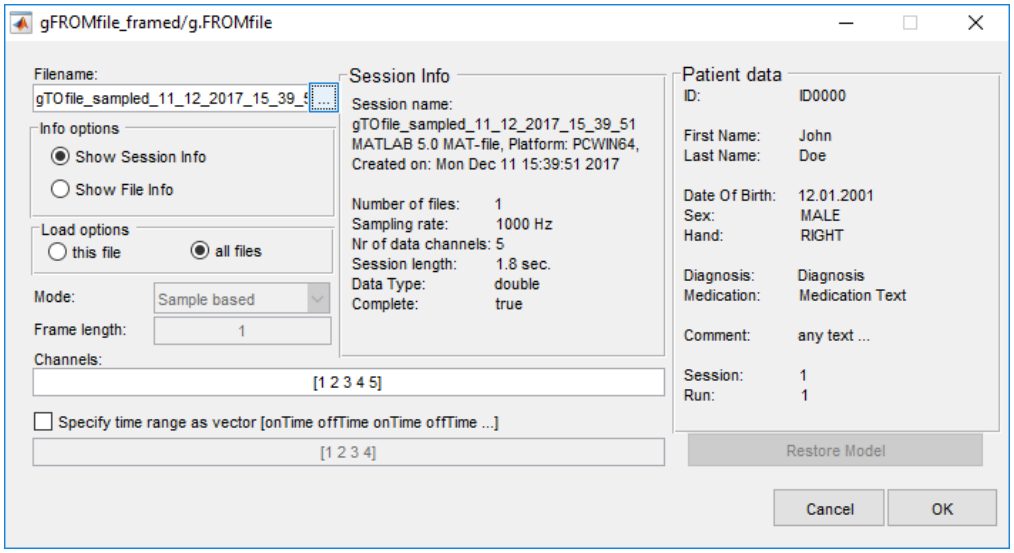
g.FROMfile Block

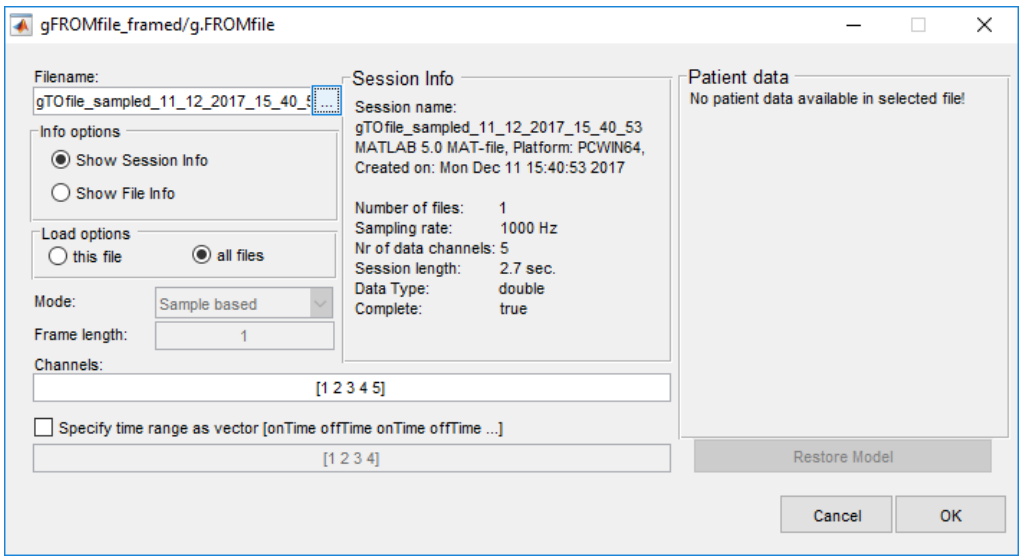
The **g.FROMfile** block is for streaming data from a g.MATfile session in real-time.

Description

The **g.FROMfile** block output provides the recorded data stored with **g.TOfile**. The data format can be single (float32) or double. If the selected file is frame-based, an additional block output is created, which outputs the activation of a frame. Use a **Demux** block to de-multiplex multiple channels. The file to load must be on the MATLAB path.

Dialog Box





Double click on the **g.FROMfile** block to edit the parameters. Click **OK** to apply or **Cancel** to discard parameter changes.

Filename	enter the filename of the session to read the data into Simulink or use the browse button (…) to select the file	
Info options	Show Session Info	Show stored recording session information
	Show File Info	Show file information
Load options	this file	only load this file
	all files	load all files of the session
Mode	indicates whether the loaded data is sample or frame based	
Frame length	indicates the frame length of loaded data 1 for sample based data number of samples per frame for frame based data	
Channels	shows the number of channels and allows selecting channels for output	
Specify time range	this function is only available for sample based data, and provides the functionality of cutting specified time ranges from the signal e.g. [0 10 30] → output first 10 seconds and from second 30 to the end	
Restore Model	if this button is enabled, a Simulink model was stored to the loaded file click this button to restore the model **NOTE:** the model will be restored in the file restored.mdl or restored.slx depending on the saved file format **ATTENTION:** restored.mdl / .slx model in the path will be replaced!	

gMATrevise Function

```
gMATrevise(sessionname, newsessionname, filesize, timechannel,
channels, begintime, endtime, datatype)
```

Description Convert data files recorded with the **g.TOfile** block to a defined size

Parameters

sessionname	string	data filename without four digit number
newsessionname	string	new data file
filesize	double	size in megabytes of the new files

Optional

timechannel	int	convert the time stamp (0 … no, 1 … yes)
channels	int or char	vector with the channels, that should be saved in the new session files or `all` for all channels
begintime	double	start time of the data
endtime	double	end time of the data
datatype	string	can be ‘single’ (float32) or ‘double’

gMATparam Function

```
[file, session] = gMATparam(fname)
```

Description Get information about a g.MATfile and the associated session.

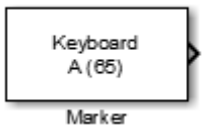
Parameter

fname	string	name of the g.MATfile

Output

file	name (string)	name of the file
	info (string)	information about the version and recording date/time
	nr (int)	number of the file within the session
	startt (double)	start time of the file within the session
	endt (double)	end time of the file within the session
	channels (int)	number of channels in the file (excluding time stamp channel)
	lastFile (string)	indicates the last file (`true` if last file, `false` otherwise)
	varname (string)	name of the variable data is saved in the file
	datatype (string)	can be `single` (float32) or `double`
	sampleRate (int)	sample rate of recorded data
session	name (string)	name of the session (extended by the file number)
	complete (string)	`true` if all files are available in the current folder, `false` otherwise
	IsFrameBased (double)	`1` if session is frame based, `0` otherwise
	IsSampleBase (double)	`1` if session is sample based, `0` otherwise
	FrameLength (double)	number of samples per frame (1 for sample based data)
	length (double)	total recording time
	nrOfFiles (int)	total number of files

Marker Block

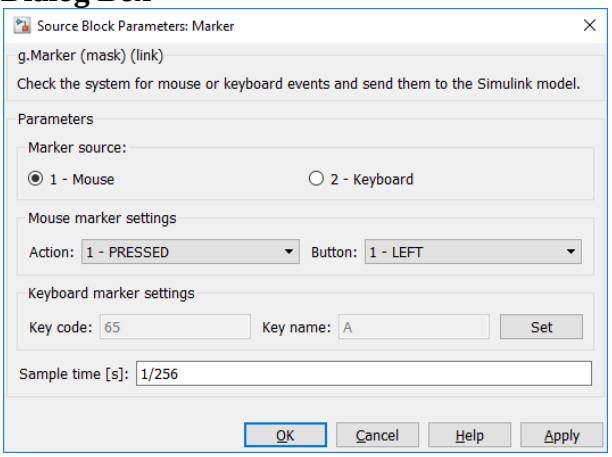


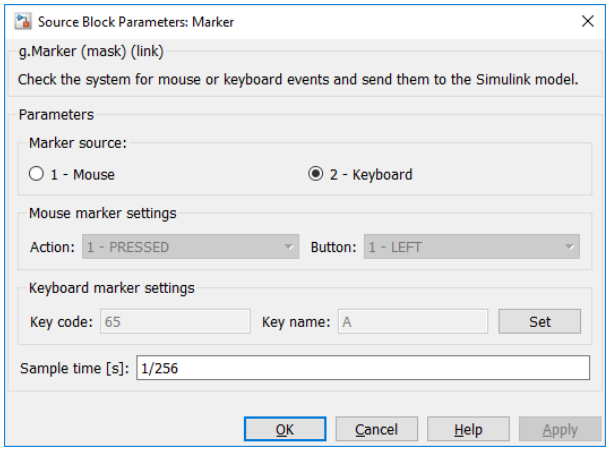
The **Marker** block is for generating keyboard and mouse markers in Simulink.

Description

The **Marker** checks the system for keyboard and mouse events and generates markers in Simulink. To generate multiple markers in Simulink use one **Marker** block for each marker.

Dialog Box

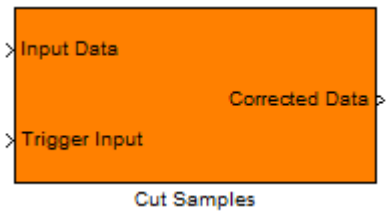




Double click on the **Marker** block to edit the parameters.

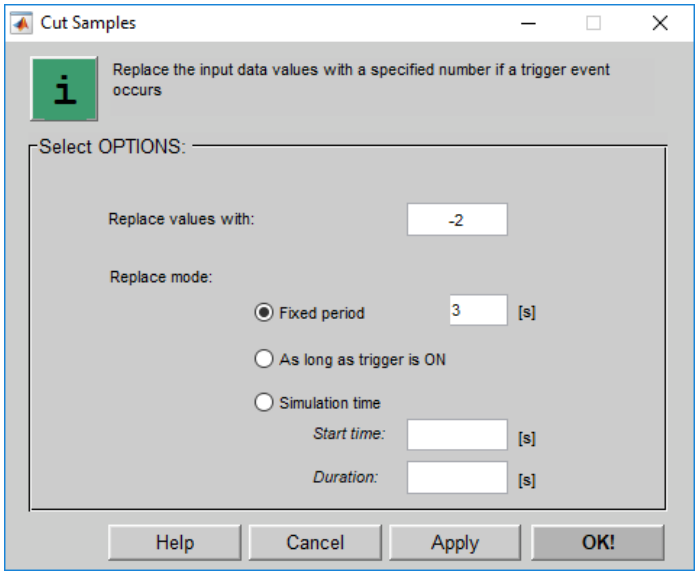
Marker source		Specify if the block should check for keyboard or mouse events
Mouse marker settings	Action	Pressed: generate marker when the button is pressed RELEASED: generate marker when the button is released ANY: generate marker when the button is pressed or released
	Button	LEFT: generate markerwhen the left button of the mouse is used RIGHT: generate marker when the left button of the mouse is used
Keyboard marker setting	Key code	the key code the block is listening to
	Key name	the key name of the corresponding key code
Set		Opens a window to specify key code and key name by keyboard hit
Example		Marker_Demo.slx

Cut Samples



Description The **Cut Samples** block replaces input data values with a specified number if a trigger event occurs.

Dialog Box



Double click on the **Cut Samples** block to edit the parameters

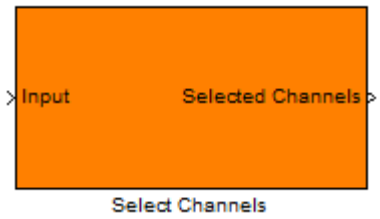
Replace values with — enter number that is used to correct the input data

Replace mode

- Fixed period [s] — all input data values in this period are replaced
- As long as trigger is ON — all input data values are replaced as long as trigger is on
- Simulation time — all input data values are replaced for a specific duration starting after the trigger onset
  - Start time [s] — starting time
  - Duration [s] — length

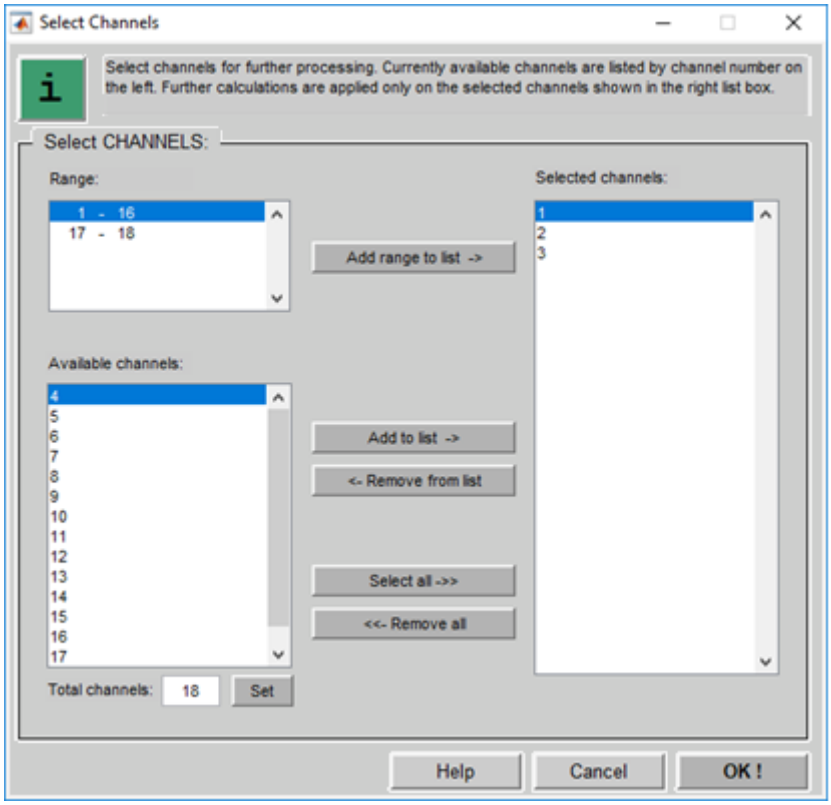
Example Simulink model — `gCutSamples.mdl`

Select Channels



Description The **Select Channels** block specifies channels for further processing. Available channels are listed by the channel number on the left. Further calculations are only applied to selected channels listed in the right box.

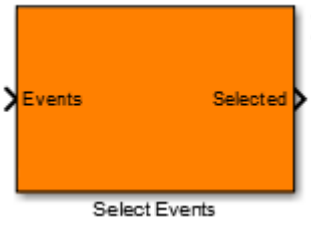
Dialog Box



Double click on the **Select Channels** block to edit the parameters

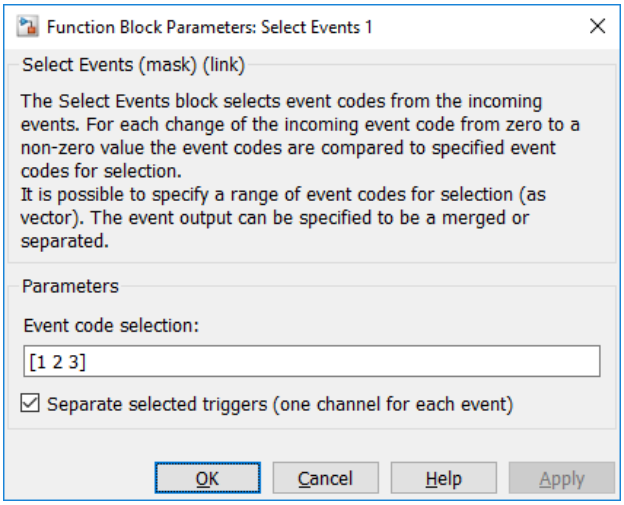
Range	specify a range of available channels
Available channels	shows all available channels
Total channels	specify the number of available channels
Selected channels	shows the specified channels
Example Simulink model	`gSelectChannels.mdl`

Select Events



Description The Select Events block selects event codes from the incoming events. For each change of the incoming event code (from zero to a non-zero value), the event codes are compared to specified event codes for selection.
It is possible to specify a range of event codes for selection (as a vector). The event output can be specified to be a merged or separated.

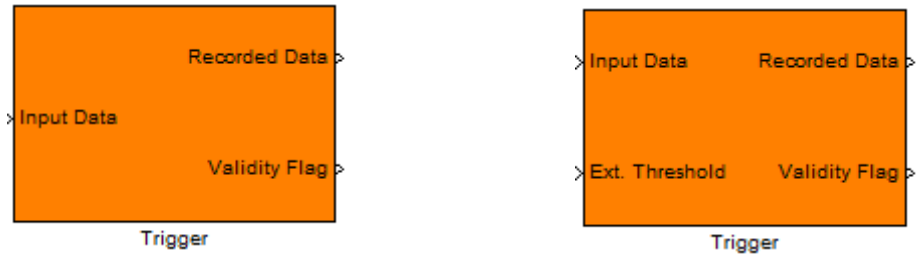
Dialog Box



Double click on the **Select Events** block to edit the parameters

Event Code Selection	specify the event codes to be selected from the incoming events
Separate selected events	specifies whether the selected event codes are output in a single channel or in separate channels for each event code checked → one channel for each selected event code unchecked → one channel for all selected event codes
Example Simulink model	gSelectEvents

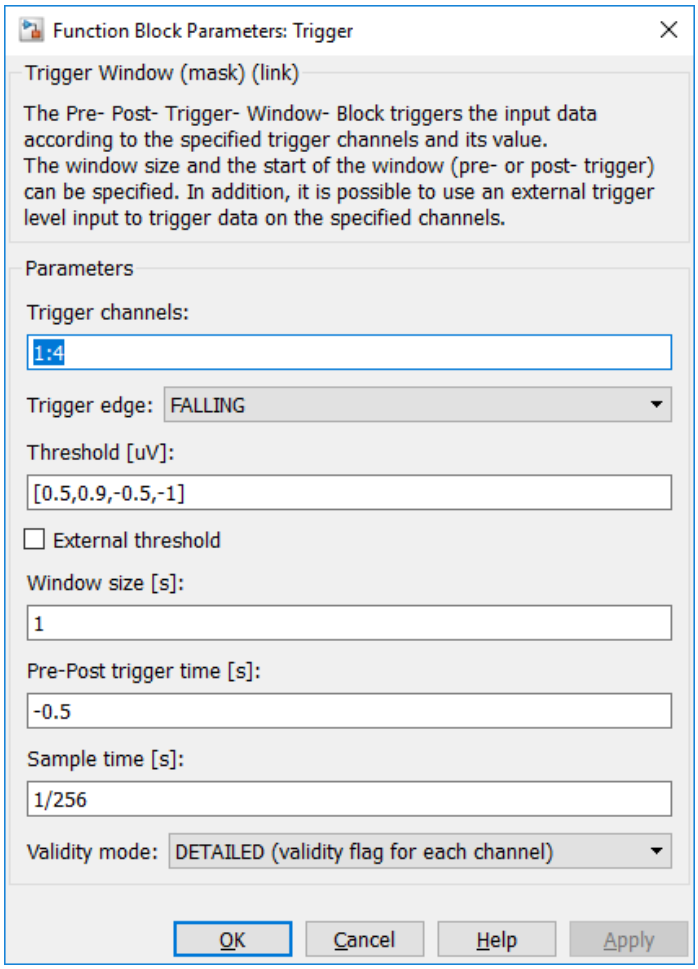
Trigger



Description The Trigger- Block triggers the input data according to the specified trigger channels, including value and edge. The window size and the start of the window (prior or post to trigger) can be specified. In addition, it is possible to use an external trigger level input to trigger data on the specified trigger channels.
To use the Trigger- Block in combination with g.THRESHOLDscope, select all channels in the Trigger channels parameter.

Note Since this block uses the Sample time- Parameter for trigger timing, it is not allowed to use DC sampling frequency.
The data type of the **Input Data** input port is double.

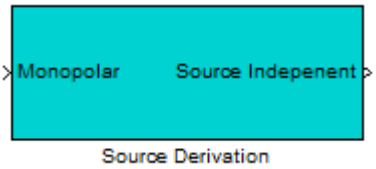
Dialog Box



Double click on the **Trigger** block to edit the parameters.

Trigger channel	select the used trigger channel it is possible to enter a vector of channels
Threshold	specify level for triggering it is also possible to enter a vector of thresholds one threshold for each channel
Window size [s]	specify the interval for triggering (total length of window)
Trigger edge	specify whether a rising or falling edge should be detected
External threshold	check the box to use an external signal as ther trigger level for all channels an addition input appears for the block
Pre-Post trigger time [s]	time before or after triggering starts use < 0 for pre- and > 0 for post- trigger timepoint
Sample time [s]	specify the sampling time
Validity mode	specify the mode of the `Validity Flag` output port STANDARD: merged triggers to a single channel DETAILED: one channel for each trigger
Example Simulink model	gTriggerDemo.mdl, gTriggerExternalThreshold.mdl

Source Derivations

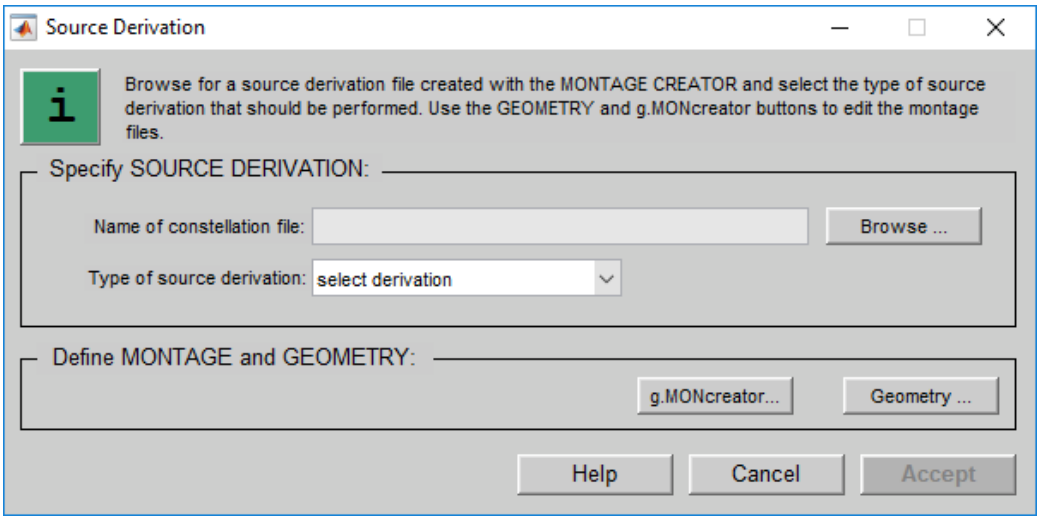


Description

The **Source Derivation** block performs different types of source derivations (CAR, Laplacian, bipolar, …) of the input data. The source derivation file can be created with the Montage Creator.

NOTE: Source Derivation block requires g.BSanalyze to be installed on your system.

Dialog Box

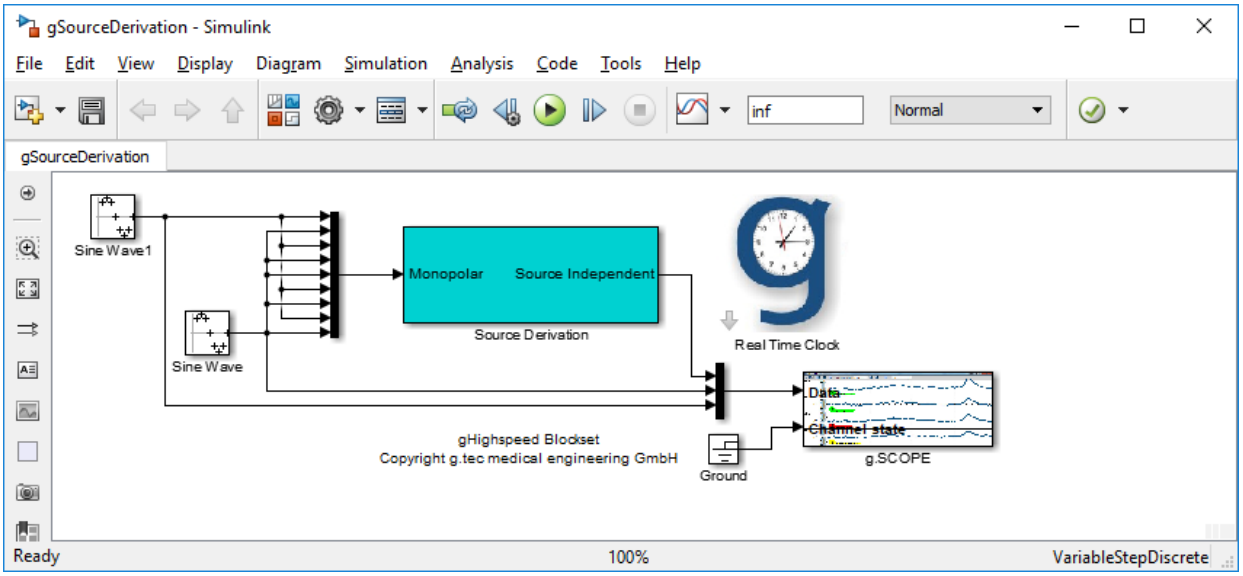


Double click on the **Source Derivation** block to edit the parameters

Name of constellation file	name of the source derivation file
Browse	open the explorer window to search for the file
Type of source derivation	specify the type of source derivation that should be performed
g.MONcreator	press button to create new constellation or inspect a constellation file with the Montage Creator
Geometry	edit the electrode positions of the montage
Example Simulink model	`gSourceDerivation.mdl`

Example:

1.) Open the Simulink model gSourceDerivation by double clicking the corresponding icon on the **g.tec Highspeed Library/Examples** section of the **Simulink Library browser** or enter gSourceDerivation.mdl at the Matlab command line.

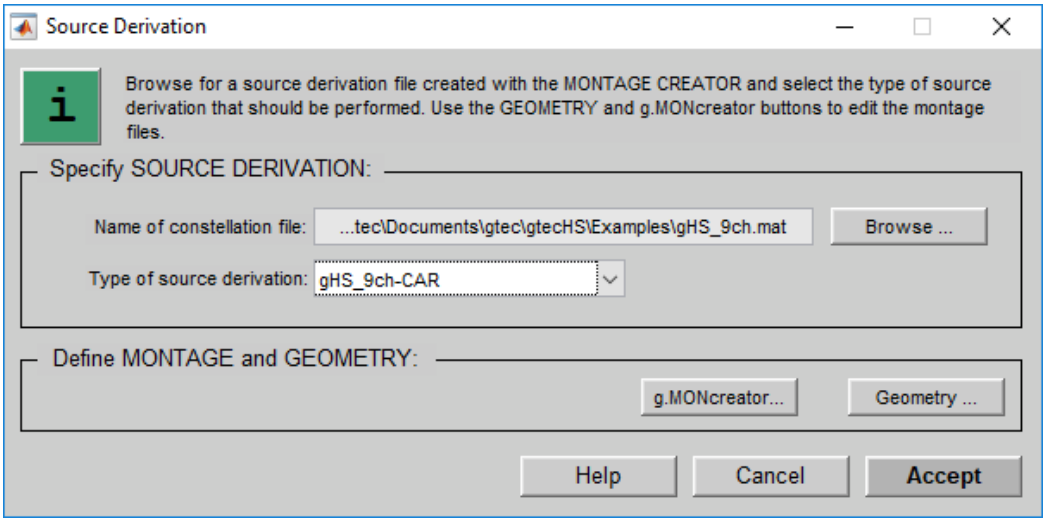


2.) Double click on the **Source Derivation** block.

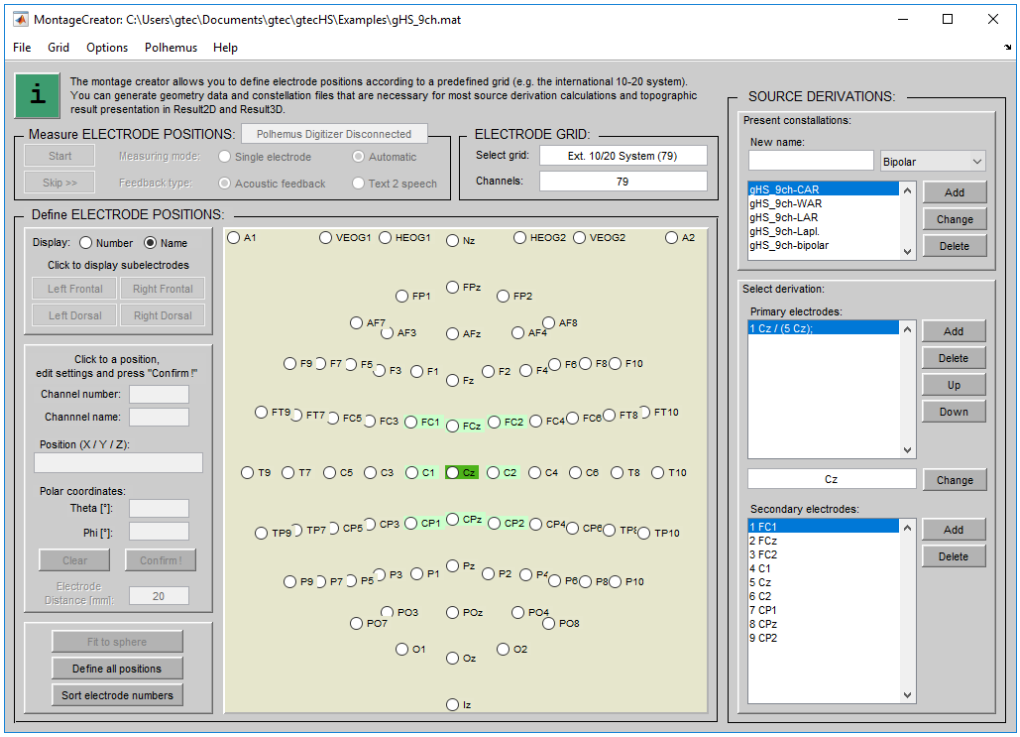
3.) Click on the **Browse…** button and open the file `gHS_9ch.mat` from the folder

`C:\Program Files\gtec\gtecHS\Examples`

4.) For the **Type of source derivation** select `gHS_9ch-bipolar`



5.) Click on the **g.MONcreator…** button to open the constellation file in the Montage Creator.

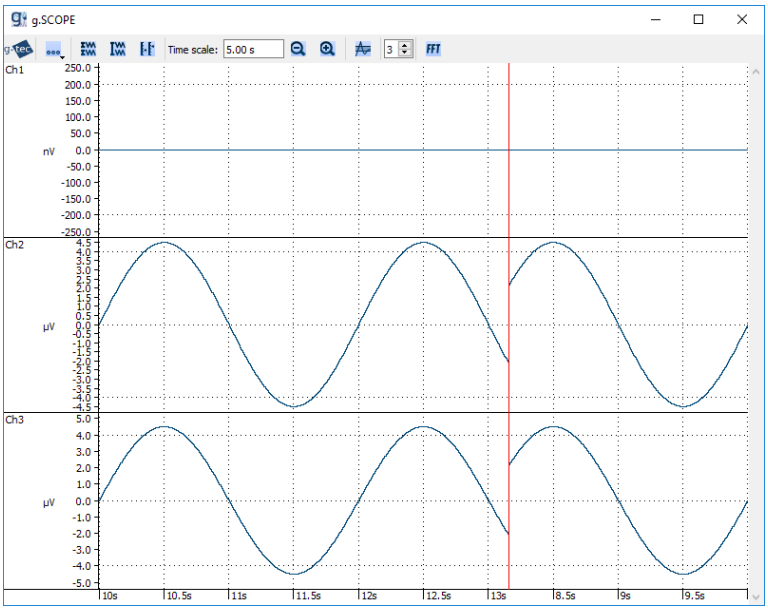


In the bipolar constellation the electrode `CP2` will be subtracted from the electrode `Cz`. Then close the Montage Creator.

6.) Press the button **Accept** button in the **Source Derivation** dialog box

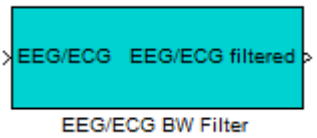
7.) To start the simulation, click on **Start** under the **Simulation** menu

8.) Double click the **g.Scope** block to view the following screen:



The first channel shows the bipolar derivation of the two sine-waves, which results in a zero line.

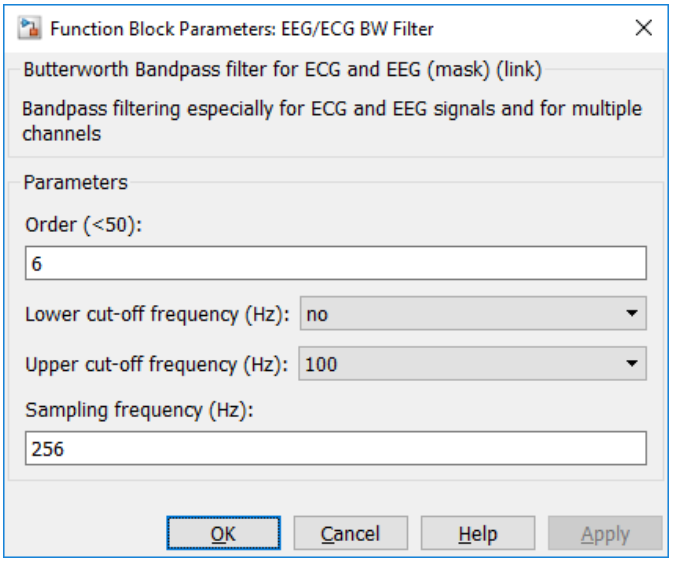
EEG/ECG BW Filter



Description

The **EEG/ECG BW Filter** block can be used for Butterworth bandpass filtering. The filters were designed especially for ECG and EEG signals, and work for multiple channels with optimized speed. Use the block to filter the incoming biosignal data to extract activity in specific frequency bands. The filter is realized using a Butterworth band-pass filter with twice the specified order.

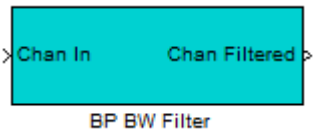
Dialog Box



Double click on the **EEG/ECG BW Filter** block to edit the parameters

Order (<50)	filter order of the Butterworth filter
Lower cut-off frequency (Hz)	lower cut-off frequency
Upper cut-off frequency (Hz)	upper cut-off frequency
Sampling frequency (Hz)	sampling rate of the used amplifier
Example Simulink model	gFilterDemo_FFTPlot.slx, gFilterDemo_SpectrumPlot.slx

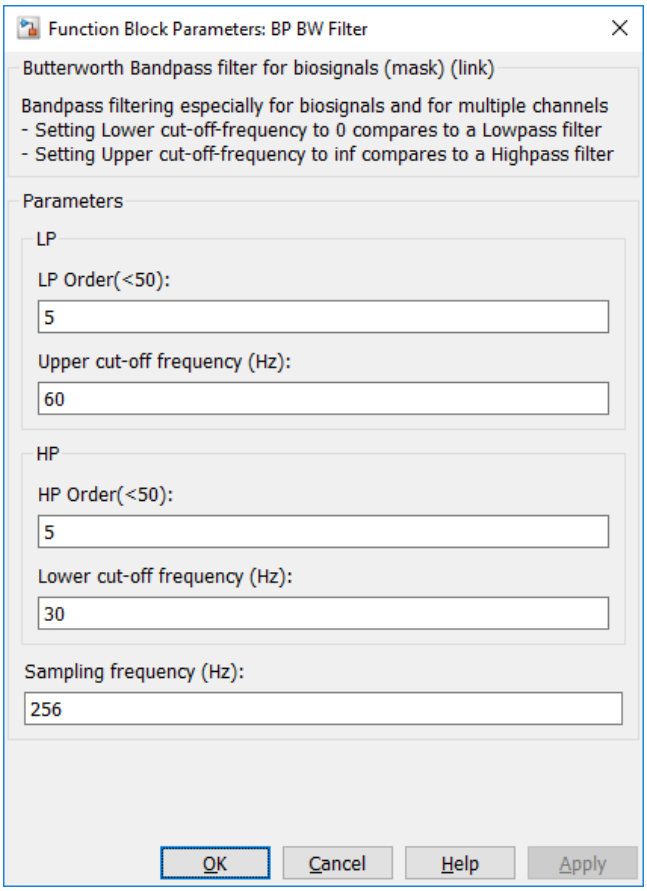
BP BW Filter



Description

The **BP BW Filter** block can be used for Butterworth bandpass filtering, especially for biosignals and for multiple channels. Use the block to filter the incoming biosignal data to extract activity in specific frequency regions. The filter is realized using a cascaded Butterworth high-pass and low-pass filter with the specified filter order.

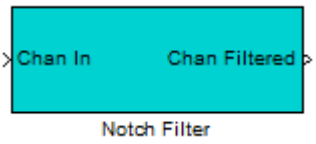
Dialog Box



Double click on the **BP BW Filter** block to edit the parameters

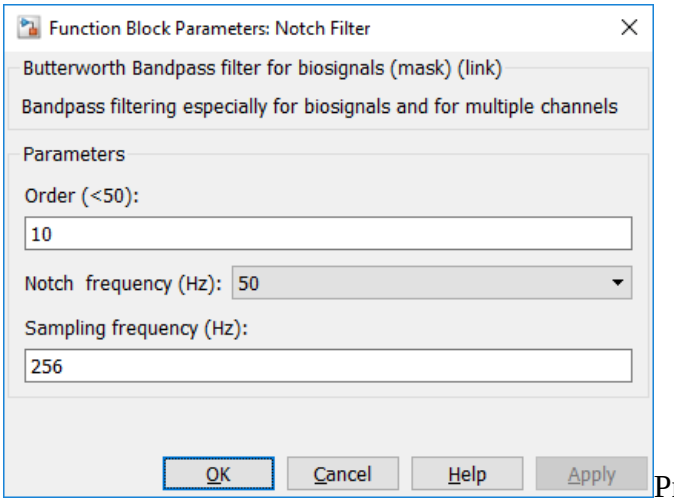
Order LP (<50)	filter order of the Butterworth low-pass filter
Upper cut-off frequency (Hz)	upper cut-off frequency (inf → no filter)
Order HP (<50)	filter order of the Butterworth high-pass filter
Lower cut-off frequency (Hz)	lower cut-off frequency (0 → no filter)
Sampling frequency (Hz)	sampling rate of the used amplifier
Example Simulink model	gFilterDemo_FFTPlot.slx, gFilterDemo_SpectrumPlot.slx

Notch Filter



Description The **Notch Filter** block is used to suppress power line interference with a frequency of 50 or 60 Hz. It works for multiple channels and is optimized for speed.

Dialog Box

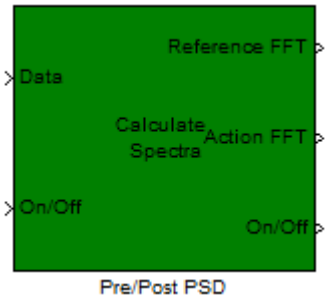


Pre

Double click on the **Notch Filter** block to edit the parameters

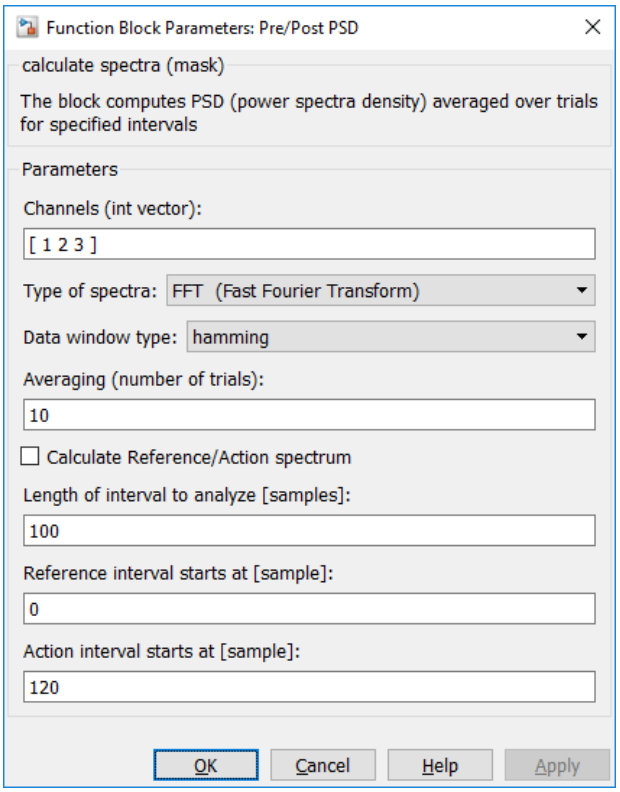
Order (<50)	filter order of the band-stop filter
Notch frequency (Hz)	band-stop frequency
Sampling frequency (Hz)	sampling rate of the used amplifier
Example Simulink model	gFilterDemo_FFTPlot.slx, gFilterDemo_SpectrumPlot.slx

Pre/Post PSD



Description The **Pre/Post PSD** block computes PSD (power spectra density) averaged over trials for specified intervals. If a reference interval and an action interval are analysed, a significance test is applied to identify reactive frequency bands.

Dialog Box

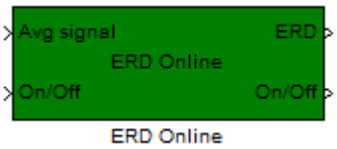


Double click on the **Pre/Post PSD** block to edit the parameters

Channels (int vector)	select the channels
Type of spectra	select type of spectrum **FFT (Fast Fourier Transform)** **PSD (Power Spectrum Density)**
Data window type	select the data window type for action and reference interval. Window type can be

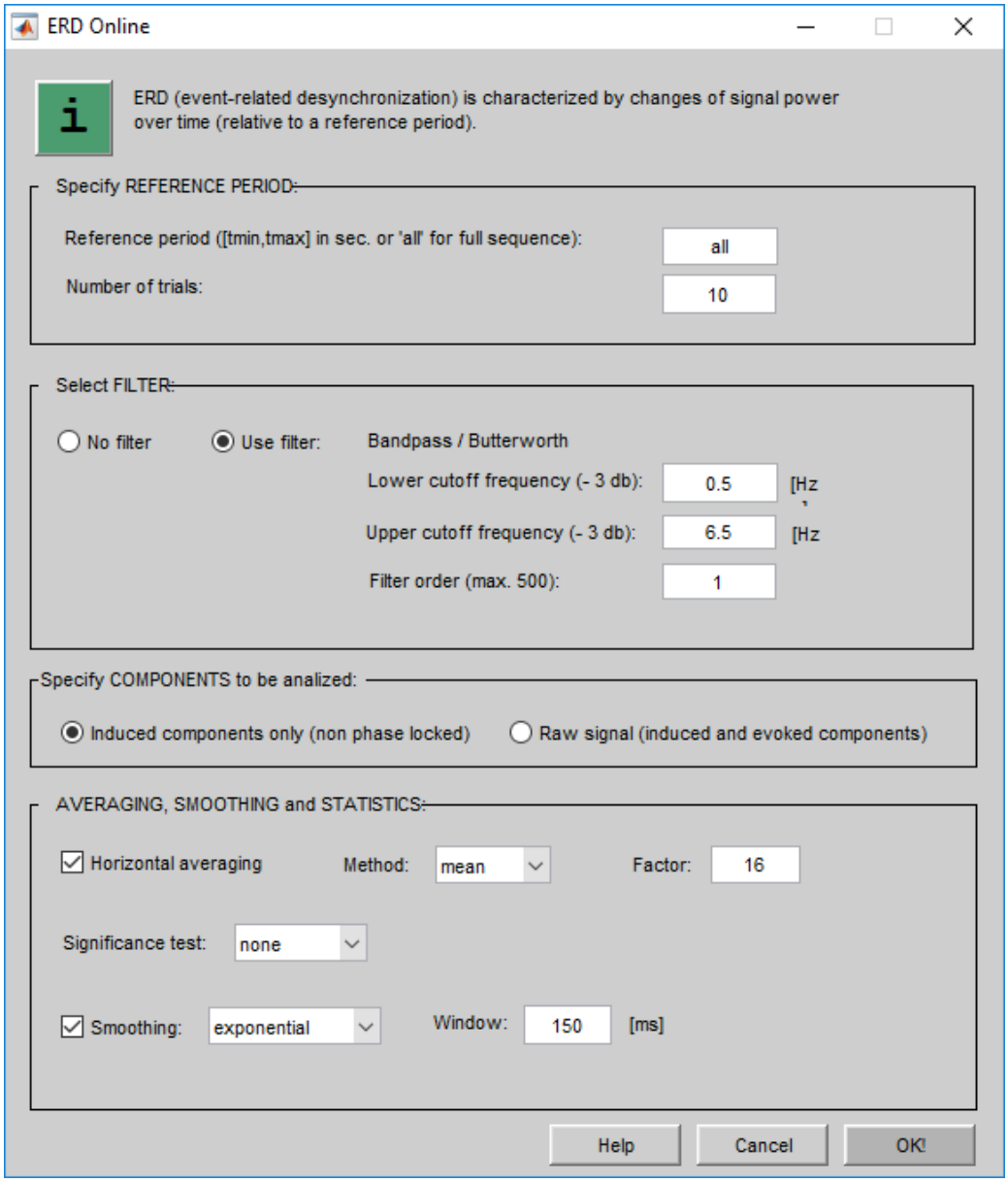
	boxcar, **hamming** or **hanning**.
Averaging (number of trials)	specify the number of trials to average
Calculate Reference/Action spectrum	check the box to calculate the spectrum of the reference interval and of the action interval and compare the results
Length of interval to analyze [samples]	specify the length of the reference and action interval in samples
Reference interval starts at [sample]	insert the start point of the reference interval in samples
Action interval starts at [sample]	insert the start point of the action interval in samples

ERD Online



Description The **ERD Online** block calculates the ERD (event-related desynchronization). ERD is characterized by changes in signal power over the time (relative to a reference period).

Dialog Box



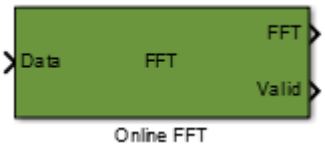
Double click on the **ERD Online** block to edit the parameters

Reference period ([tmin,tmax] in sec. or 'all' for full sequence)	select a specific interval `[tmin, tmax]` in sec. as reference period or use the entire trial `all`
Number of trials	number of used trials for the ERD calculation
Use filter	Check this box to use a bandpass filter.
Lower cutoff frequency (-3 db) [Hz]	lower cut-off frequency
Upper cutoff frequency (-3 db) [Hz]	upper cut-off frequency
Filter order (max. 500)	filter order of the bandpass filter
Specify COMPONENTS to be analyzed	select **Induced components only (non phase locked)** where phase locked components will be removed or **Raw signal,** where non-phase locked and phase locked components will be analyzed
Horizontal averaging	average over consecutive samples
Method	can be **mean** or **median**
Factor	number of samples to average
Significance test	perform the significance test
Smoothing	smooth the result **average** - average over the specified window length in ms **exponential** - exponential window with window length in ms **cosine** - cosine window with a window length in ms
Window [ms]	window length for smoothing
Example Simulink model	`gERD.mdl`

Load the file `erd_data.mat` from path `C:\Program Files\gtec\gtecHS\Examples` to run the example `gERD.mdl`.

In the **Buffer** block, the **Output buffer size (per channel)** must be set to `512`.

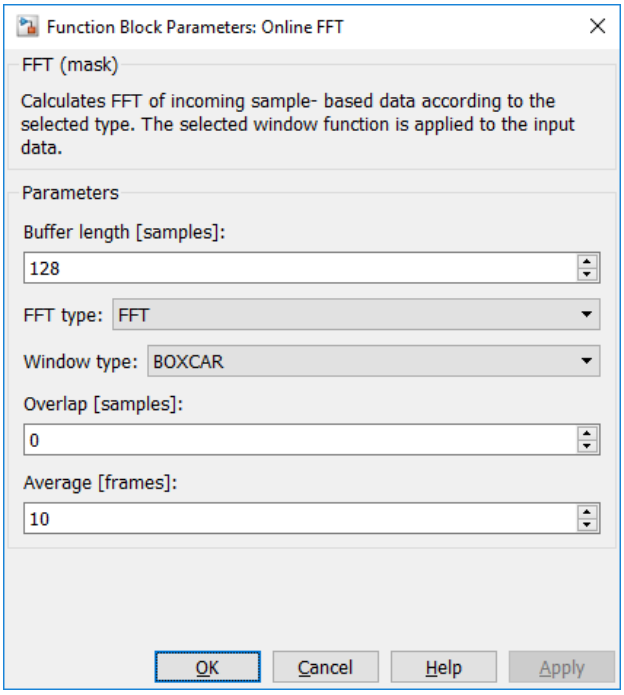
Online FFT



Description

The **Online FFT** block calculates the spectrum for all channels passed to the block. Different FFT and window types can be selected as well as frame overlaps and averaging.

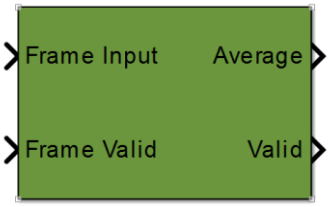
Dialog Box

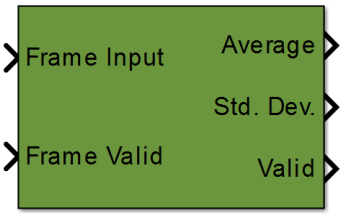


Double click on the **Online FFT** block to edit the parameters.

FFT type	select type of spectrum **FFT (Fast Fourier Transform)** **PS (Power Spectrum)** **PSD (Power Spectrum Density)**
Window type	select the data window type. **BOXCAR**, **HAMMING** or **HANNING**.
Averaging (number of frames)	specify the number of frames to average
Buffer length (# of samples)	specify the length of the interval in samples
Overlap (# of samples)	define the overlap of the interval in samples
Example Simulink model	`gOnlineSpecs.mdl`

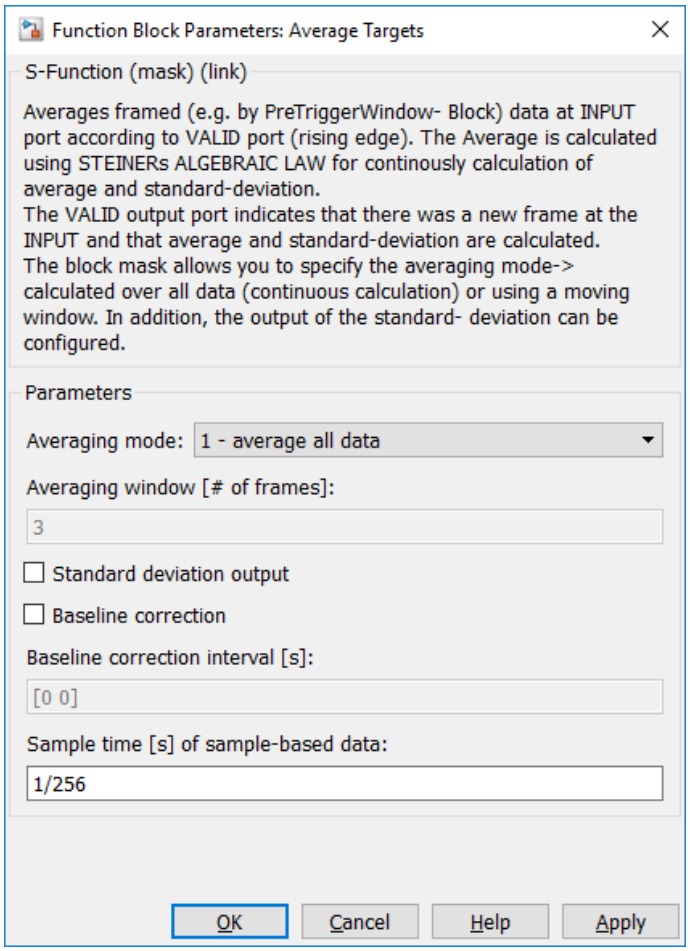
Online Averaging





Description Averages framed (see TriggerBlock, Pg. 46) data according to the 'Frame Valid' port. The Average is calculated using STEINERs ALGEBRAIC LAW for continuous calculation of average and standard-deviation.
The 'Valid' output port indicates that there was a new frame and average and standard-deviation are calculated and updated.
The block mask allows you to specify the averaging mode-> calculated over all frames or using a moving window. The output of the standard-deviation can also be configured.

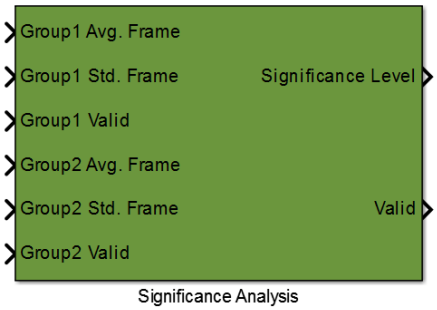
Dialog Box



Double click on the **Online Averaging** block to edit the parameters.

Averaging Mode	`1-average all data:` running average will use all incoming trials `2-moving window averaging:` specify the number of trials to average over the last N trials only
Averaging Window	only enabled in moving window averaging mode. Specifies the number of trials of the window
Standard-Deviation Output	If this box is checked/selected, an additional output is presented that holds the standard-deviation of the incoming data frames
Sample Time	Sample time of the incoming frame data (original sample time) in seconds
Example Simulink model	`gEP.mdl`

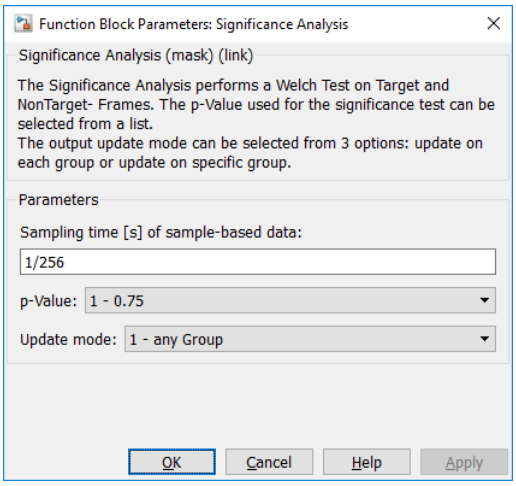
Significance Analysis



Description

The Significance Analysis block performs a Welch test of the 2 populations represented by average and standard deviation, Group1 and Group2. The p-value used for the significance of the Welch test can be selected from a list.
The output update mode can be selected from 3 options: update on each group change or update on group1 or group2 changes only.

Dialog Box



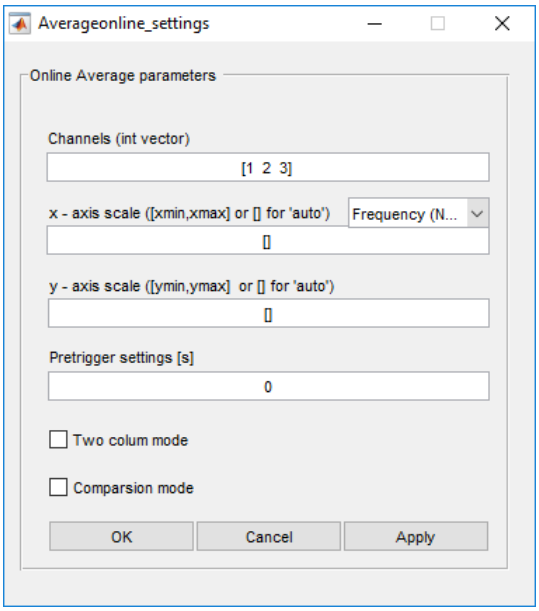
Double click on the **Statistical Analysis** block to edit the parameters.

Sampling time	Sample time in seconds the block is running
p-Value	Set the p-Value for significance calculation. Available p-Values: 0.75, 0.875, 0.9, 0.95, 0.975, 0.99, 0.995, 0.999
Update Mode	`1-any Group` update output with changed input of any Group `2-Group1 only` update output only if Group1 input changed `3-Group2 only` update output only if Group2 input changed
Example Simulink model	`gEP.mdl`

Online Plot

Description The **Online Plot** block plots the forwarded input data. This block can also be run in a comparison mode with two input data.

Dialog Box



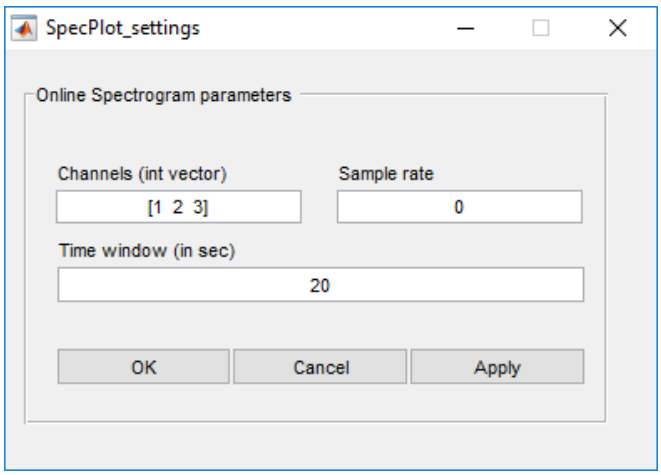
Double click on the **Online Plot** block and click on **Parameters** in the menu bar to edit the parameters

Channels (int vector)	select the channels
x-scale ([xmin, xmax] or [] for ‘auto’)	scale x - axis
x-label	the label of the x-axis could be **Normalized Frequency** or **Time in sec**
y-scale ([ymin, ymax] or [] for ‘auto’)	scale y axis
Pretrigger [s]	specify a pretrigger line point
Two columns mode	plot windows are arranged in two columns
Comparison mode	whether selected comparison mode is used
Example Simulink model	`gOnlineSpecs.mdl`

Online Spectrogram Plot

Description The **Online Spectrogram Plot** block plots the results of the **Calculate Spectrum** block.

Dialog Box



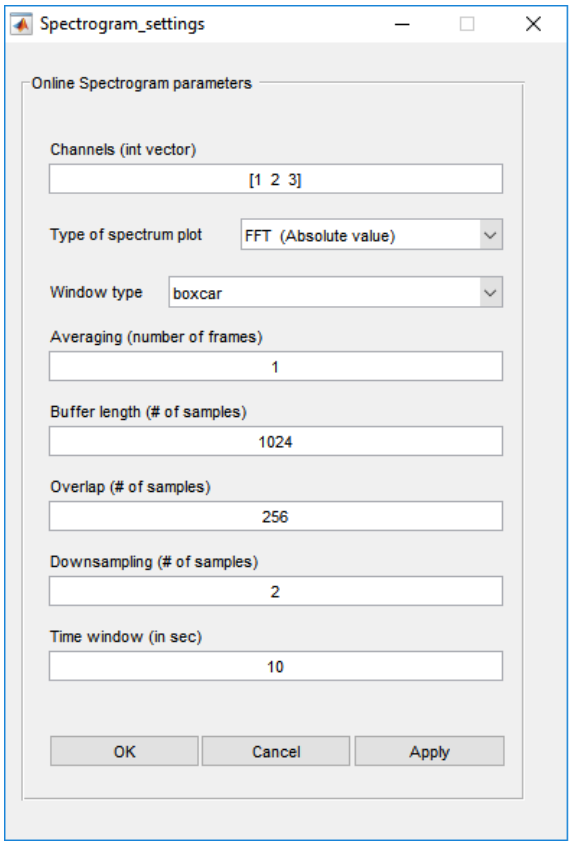
Double click on the **Online Spectrogram Plot** block and click on **Parameters** in the menu bar to edit the parameters

Channels (int vector)	select the channels
Time window (in sec.)	displayed time window in the plot
Example Simulink model	`gOnlineSpecs.mdl`

Online Spectrogram Plot + Calc

Description The **Online Spectrogram Plot + Calc** block calculates and plots the forwarded input data.

Dialog Box

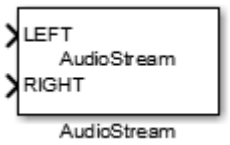


Double click on the **Online Plot** block and click on **Parameters** in the menu bar to edit the parameters

Channels (int vector)	select the channels
Type of spectrum plot	select the type from the following options: - FFT (Absolute values) - PS (Power Spectrum) - PSD (Power Spectral Density)
Window type	select the from the following options: - boxcar - hanning - hamming
Buffer length (# of samples)	Number of samples used for spectrum

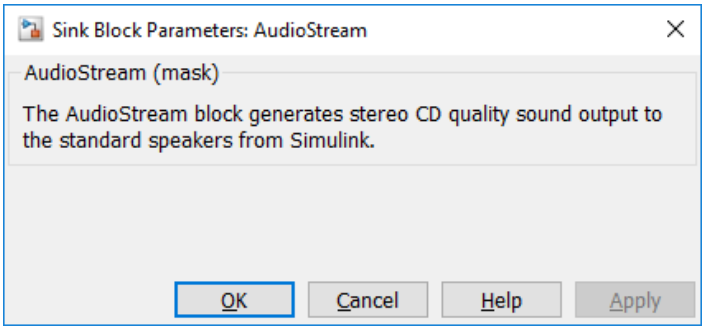
	calculcation
Overlap (# of samples)	specify how many samples the frames for calculation should overlap
Downsampling (# of samples)	specify how many samples should be combined for downsampling
Time window (s)	length of the plot

AudioStream



Description The **AudioStream** block generates stereo CD quality sound output to the standard speakers from Simulink.

Dialog box

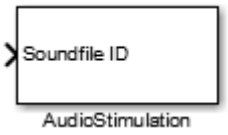


Double click the AudioStream block to edit the parameter.

Example Simulink model `AudioStream_Demo.slx`

ATTENTION: This block requires ASIO4ALL – Universal ASIO Driver For WDM Audio version 2.12 or later to be installed on the computer.

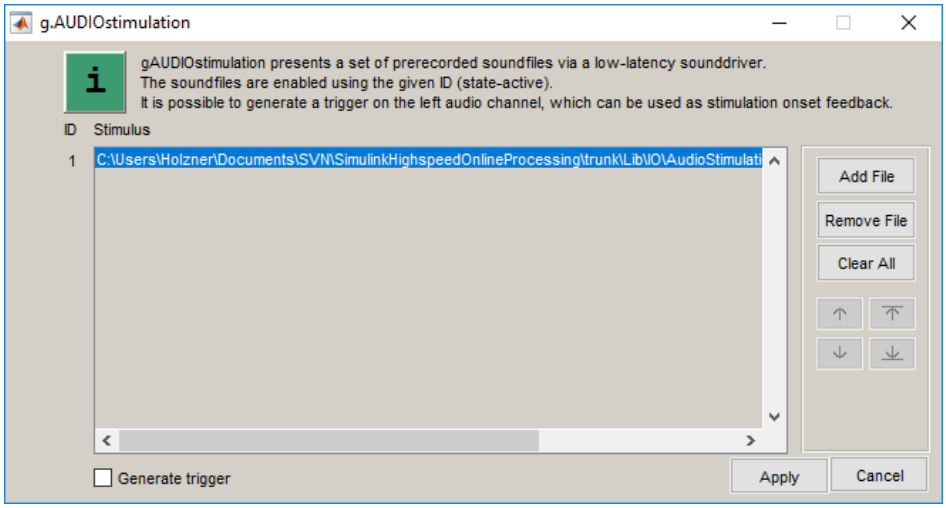
AudioStimulation



Description

The **AudioStimulation** block presents a set of pre-recorded sound files via a low-latency sound driver. The sound files are enabled using the given sound ID (see column ID in AudioStimulation GUI). As long as the sound ID input holds an ID of a sound files, this file is replayed. Setting the ID to 0 or to another ID will stop the file replay or start the other file replay respectively. It is possible to generate a trigger on the left audio channel, which can be used as stimulation onset feedback. Use mindBEAGLE audio trigger adapter box to split the output signal into a trigger and a stimulation line.

Dialog box



Double click the AudioStimulation block to edit the parameter. Use **Add File**, **Remove File** and **Clear All** to modify the sound file list. Specify the ID of one sound file by using the arrow buttons to move the sound files up and down in the list.

Example Simulink model `AudioStimulation_Demo.slx`

ATTENTION: This block requires ASIO4ALL – Universal ASIO Driver For WDM Audio version 2.12 or later to be installed on the computer.

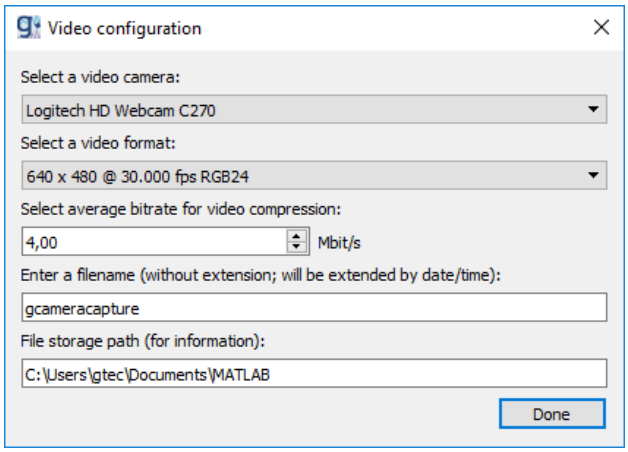
g.CAMERAcapture

Description

The **g.CAMERAcapture** block allows you to record a video from a webcam and synchronize the video with biosignal data.
The synchronization is done using the video frame number which is output from the g.CAMERAcapture block. Save this output along with the biosignal data for synchronization.

Double click **g.CAMERAcapture** block to start the video recording tool.

The toolbar allows you to configure the webcam, the camera settings and to control the video recording.
To configure the webcam click **Config** to open the configuration dialog:



1) Pick the camera to use from the **Select a video camera** list.

2) Depending on the used camera and depending on the needed video quality select a video format from the **Select a video format** list

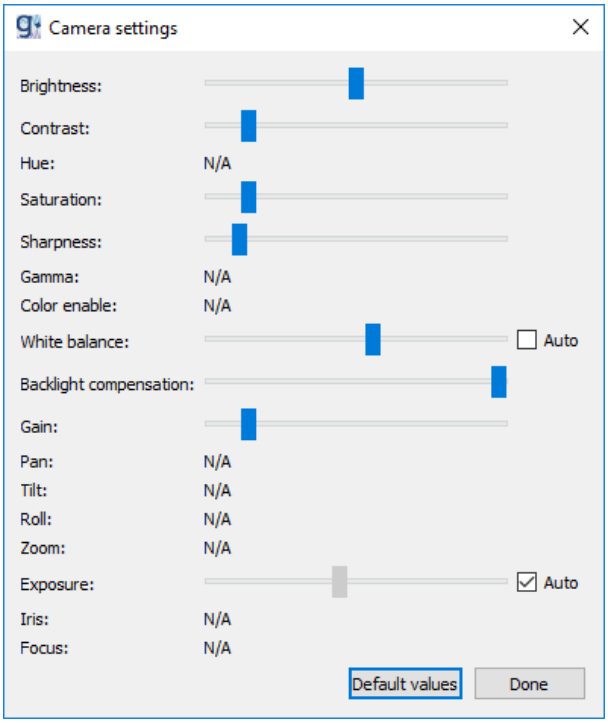
3) Use the **Select average bitrate for video compression** parameter to specify the final size of your video (see table below). Select a bitrate in the range of 0.1 and 10 Mbit/s.

Length of video [min]	Bitrate [Mbit/s]	Size of video [MB]
1	0.1	0.75
1	4	30
1	10	75
15	0.1	11.25
15	4	450
15	10	1125
60	0.1	45
60	4	1800
60	10	4500

4) The filename specified in the **Enter a filename** parameter will be extended by a post-fix string with the format '_dd_mm_yyyy_hh_mm_ss'.

5) The **File storage path** parameter is set automatically to the MATLAB workspace path of the Simulink model and cannot be modified.

To set the camera settings click **Camera settings** in the menu bar to open the settings dialog box:



Change the camera settings up to your needs. The changes will take effect immidiately so you can check the settings with the resulting video output. The parameters **White balance** and **Exposure** provides the option for auto-calibartion leading to adjustment of the white balance and exposure during your video recording. This could have an impact to the frame rate.
To restore the camera default parameters click **Default values** button.

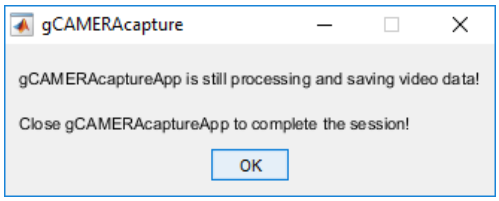
Control the video recording via the record and pause button in the toolbar.

g.CAMERAcapture is ready to record a video. Click the red record button to start recording.

NOTE: **You can only record one video file per run of the Simulink model.**

g.CAMERAcapture is currently recording a video. Stop recording of the video by clicking the blue pause button

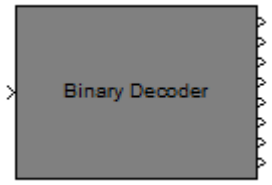
Best practise is to start and stop the video recording while the Simulink model is running. If the Simulink model ends and g.CAMERAcapture is still recording video a warning message will pop up.



Example Simulink model `gCAMERAcapture_Demo.slx`

INFO: We tested g.CAMERAcapture with Logitech c270 HD Webcam.

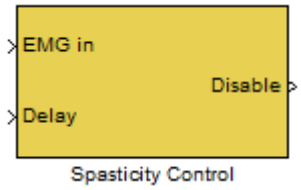
Binary Decoder



Description The **Binary Decoder** block converts an incoming double or single data into a binary value.

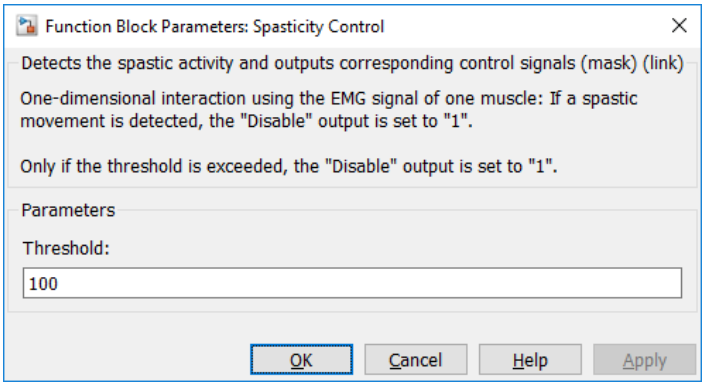
Example Simulink model `gBinaryDecoder.mdl`

Spasticity Control



Description The **Spasticity Control** block detects the muscle spastic activity in muscles and outputs corresponding control signals.

Dialog box

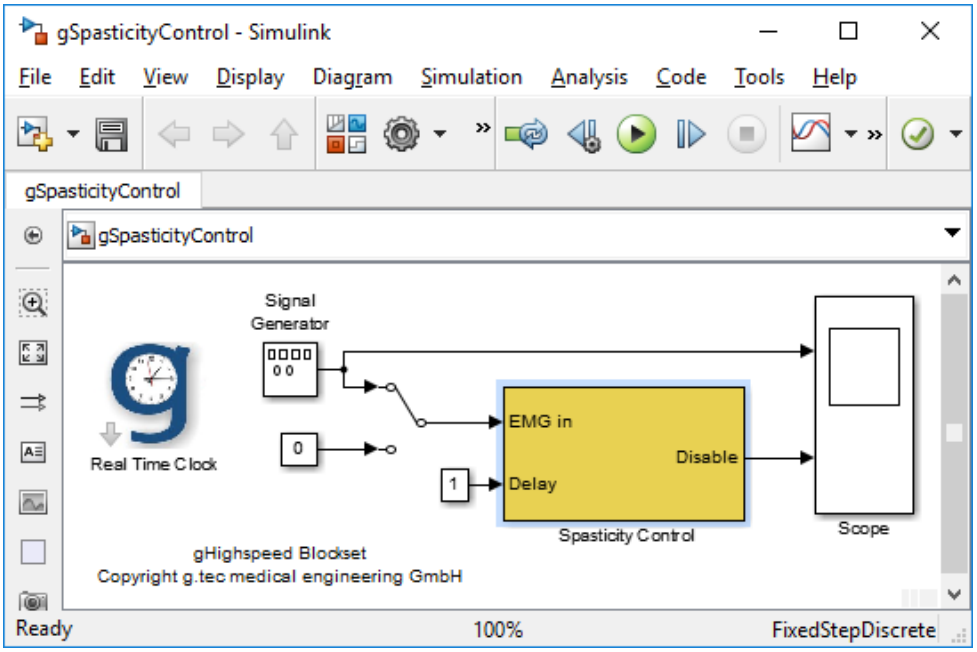


Double click on the **Spasticity Control** block to edit the parameters

Threshold	threshold value for determining the spastic movements. If this amplitude threshold is exceeded, the **Disable** output of the block will be set to 1
Example Simulink model	`gSpasticityControl.mdl`

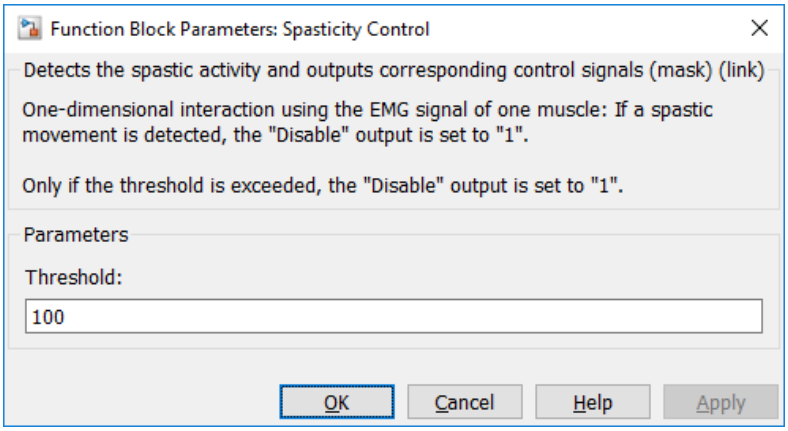
Example:

1.) Open the Simulink model `gSpasticityControl.mdl`



2.) Double click on the **Spasticity Control** block.

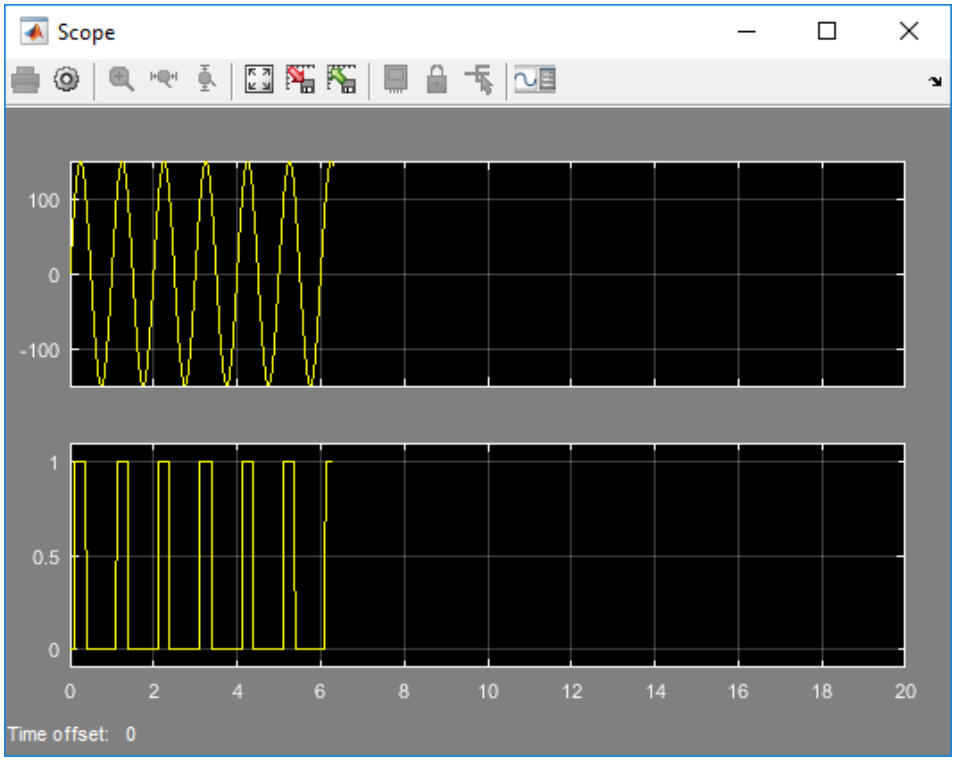
3.) For the **Threshold** select `100` mV



4.) Press the button **OK**

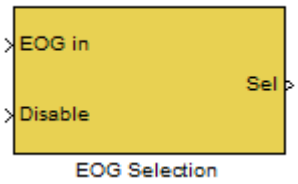
5.) To start the simulation, click on **Start**

6.) Double click on the **Scope** block to view the following screen:



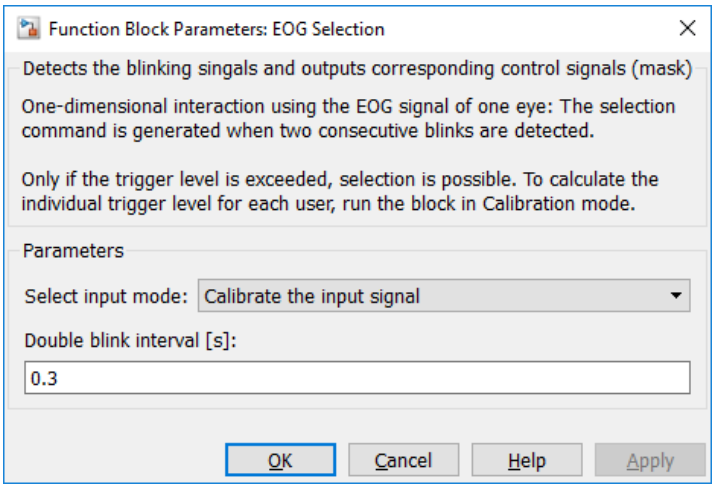
The first channel shows the input signal and the second channel displays the Disable signal which is 1 whenever the amplitude of the sine wave is higher than 100 mV.

EOG Selection



Description The **EOG Selection** block detects the blinking signals and outputs corresponding control signals.

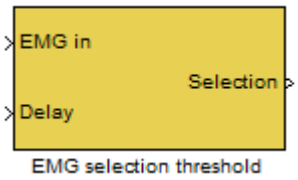
Dialog box



Double click on the **EOG Selection** block to edit the parameters

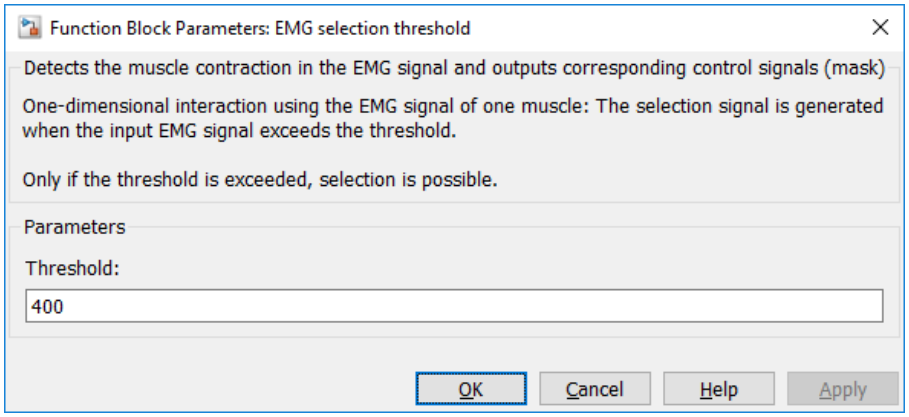
Select input mode	select `Calibrate the input signal` to calculate the trigger level for the user, `Use the input sensor` to activate the selection mode or `Use input sensor & blink detector` to distinguish between voluntary eye movements and blinks.
Double blink interval [s]	time window within which the second eye blink must occur for a valid selection
Output:	During calibration 0, when **Select input mode** is set to `Use the input sensor` is selected than a positive peak is generated when upper limit is crossed. **Select input mode** `Use input sensor & blink detector` generates a second peak which is positive for eye movements and negative for blinks.
Example Simulink model	`g.EMGEOGcontrol/g.BCI_EOG_gUSBamp`

EMG Selection threshold



Description The **EMG Selection threshold** block detects the muscle contraction in the EMG signal and outputs corresponding control signal. The control signal is thereby compared to a manually configurable threshold.

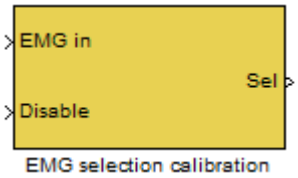
Dialog box



Double click on the **EMG Selection threshold** block to edit the parameters

Threshold threshold value the selection command

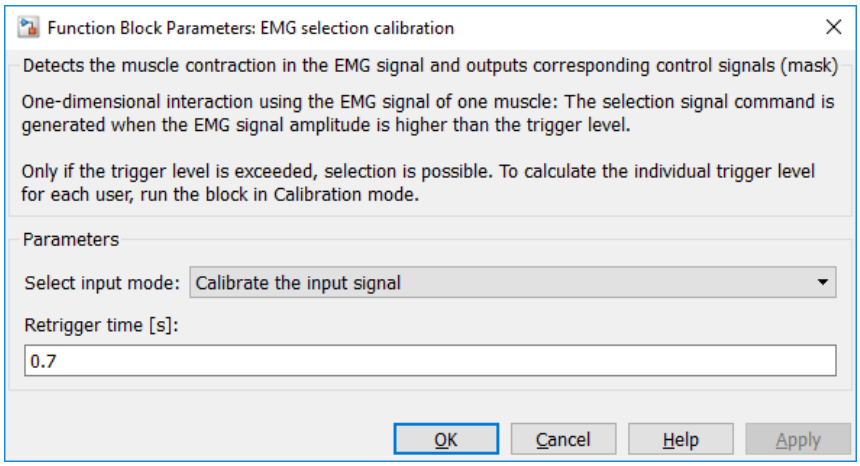
EMG Selection calibration



Description

The **EMG Selection calibration** block converts the EMG signal from one muscle to selection commands. Unlike the **EMG selection block**, the control signals are generated by comparing the input signal to a threshold that has been determined during an initial calibration phase.

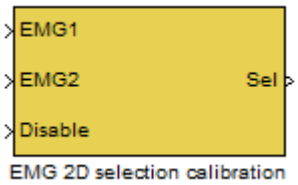
Dialog box



Double click on the **EMG Selection 2** block to edit the parameters

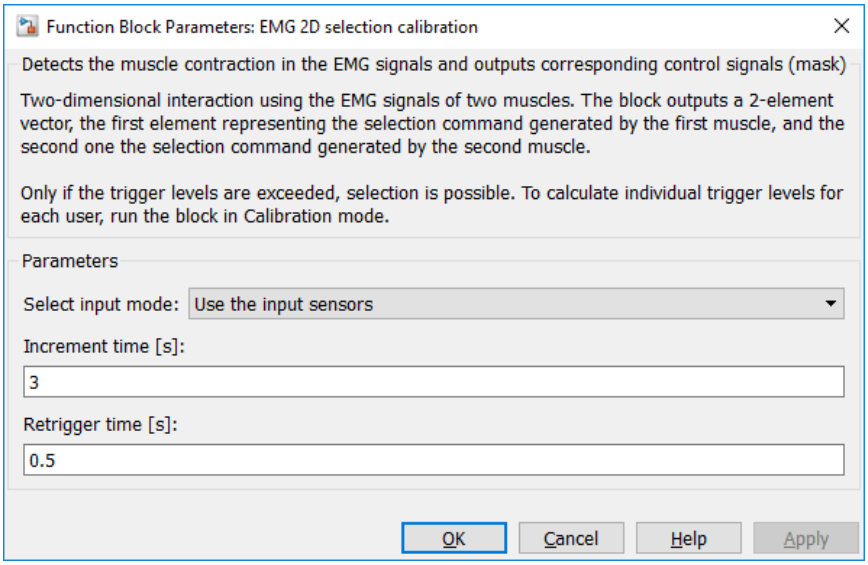
Select input mode	select `Calibrate the input signal` to calculate the trigger level for the user or `Use the input sensor` to activate the selection mode
Retrigger time [s]	minimum time between two selections
Example Simulink model	`g.EMGEOGcontrol/gBCI_EMG_gUSBamp.mdl`

EMG 2D selection calibration



Description The **EMG 2D selection calibration** detects the muscle contraction in the EMG signals and outputs corresponding control signals.

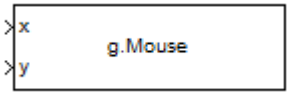
Dialog box



Double click on the **EMG 2D selection calibration** block to edit the parameters

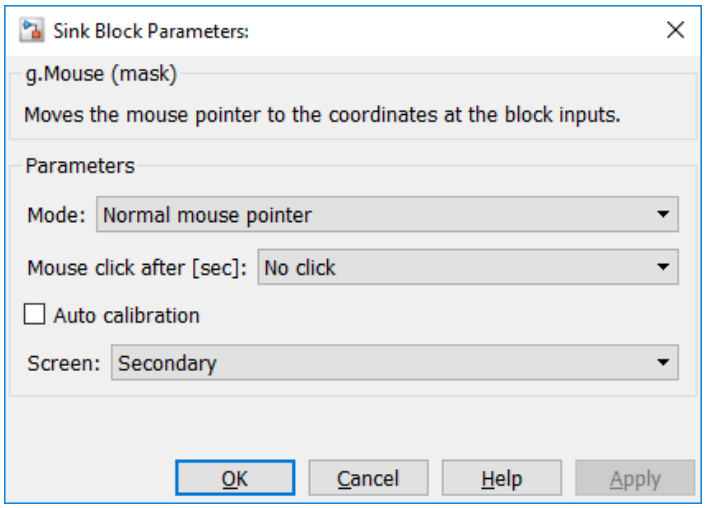
Select input mode	select `Calibrate the input signal` to calculate the trigger level for the user or `Use the input sensor` to activate the selection mode
Dwell time [s]	time window between two selections
Retrigger time [s]	minimum time between two triggers
Example Simulink model	`g.EMGEOGcontrol/g.BCI_EMG_2D_gUSBamp`

g.Mouse



Description The **g.Mouse** block controls the mouse pointer with respect to the input coordinates **x** and **y**.

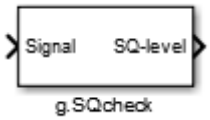
Dialog box



Double click on the **g.Mouse** block to edit the parameters

Mode	the pointer control mode could be: **Normal mouse pointer**, **Incremental mouse pointer** or **Scroll mouse pointer.**
Mouse click after [sec]	time interval between the moment when the cursor stopped on a point and the moment when the automatic click command is given. Select **No click** to control only the pointer movement.
Auto calibration	calibrate the pointer position based on the minimum and maximum values reached by the input coordinates **x** and **y** during a pre-defined **Calibration time [sec]**.
Screen	select **Primary** to control the mouse pointer on the primary screen of the PC, or select **Secondary** to move the pointer on the secondary attached screen.

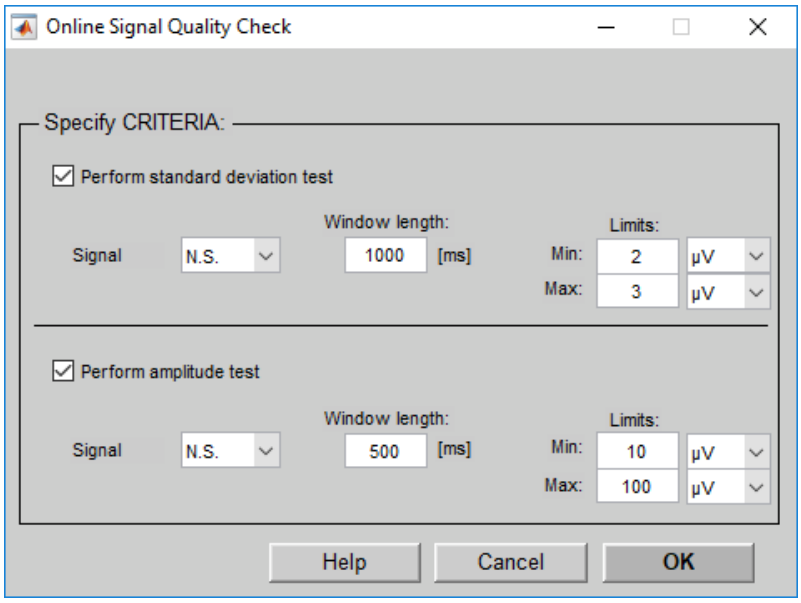
g.SQcheck



Description

The **g.SQcheck** block analyzes the incoming signal with respect to amplitude range and variability and outputs the quality rating 1 (bad, low limit), 2 (poor/high limit) 3 (good/within limits) for each input channel. You can configure the window length and the limits for the amplitude and the standard deviation criteria, or you can select the signal type from the provided list, which will set the default configuration. It is possible to activate and deactivate the test separately.

Dialog box



Double click on the **g.SQcheck** block to edit the parameters

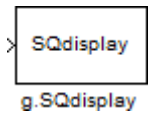
Window length [ms]	Specifies the length of the window used for the criteria.

Signal Quality Levels

Standard deviation test	1 (bad) – standard deviation exceeds MAX limit 2 (poor) – standard deviation falls below MIN limit 3 (good) – standard deviation satisfies MIN and MAX limits
Amplitude test	1 (bad) – < 17 % of evaluation window samples satisfies limits 2 (poor) – < 50 % of evaluation window samples satisfies limits 3 (good) – > 50 % of evaluation window samples satisfies limits

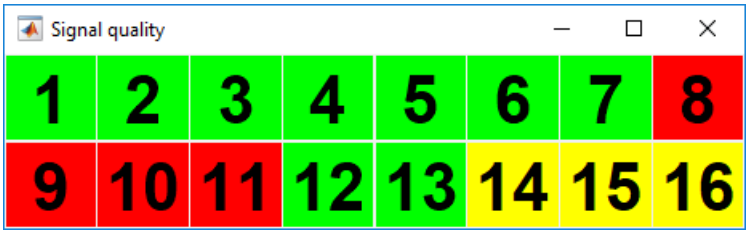
If tests are combined, each test creates its quality level and voting for a quality level. The quality level with the highest number of votes will be the overall result. If there are quality levels with the same vote, the poorer quality of both is chosen.

g.SQdisplay



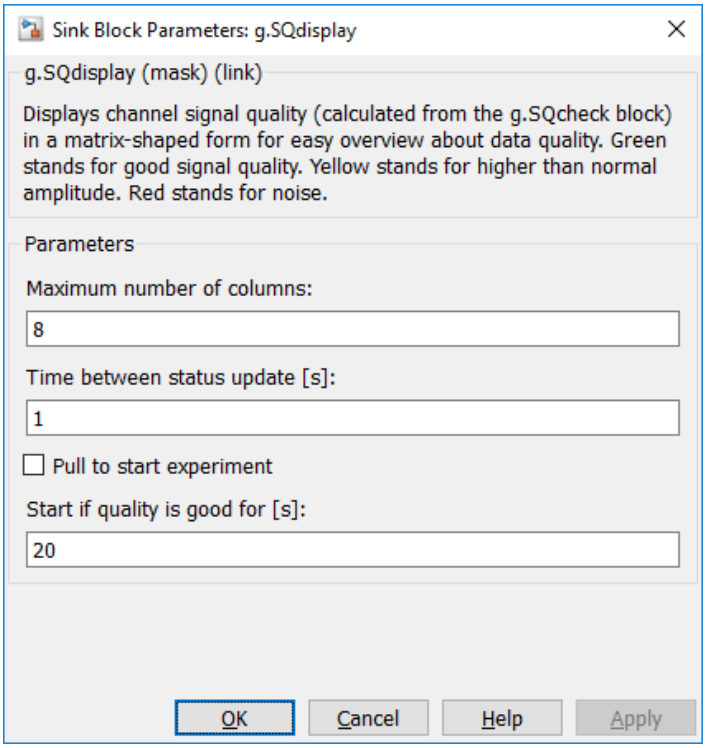
Description

The **g.SQDisplay** displays a matrix showing the signal quality of each input channel. The colors correspond to the quality ratings generated by the **gSQcheck** block.



A red or yellow color indicates poor signal quality, or low and high limits exceeded (respectively) and good quality is indicated in green.

Dialog box

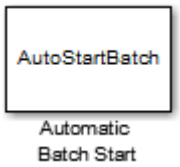


Double click on the **g.SQdisplay** block to edit the parameters

Maximum number of columns:	Defines the number of columns of the display matrix
Time between status update [s]	The interval defining how often the status display is updated.
Start trigger	Output a start trigger. 0 wait, 1 start. A Start button is displayed. When pressed, trigger is changed to 1.

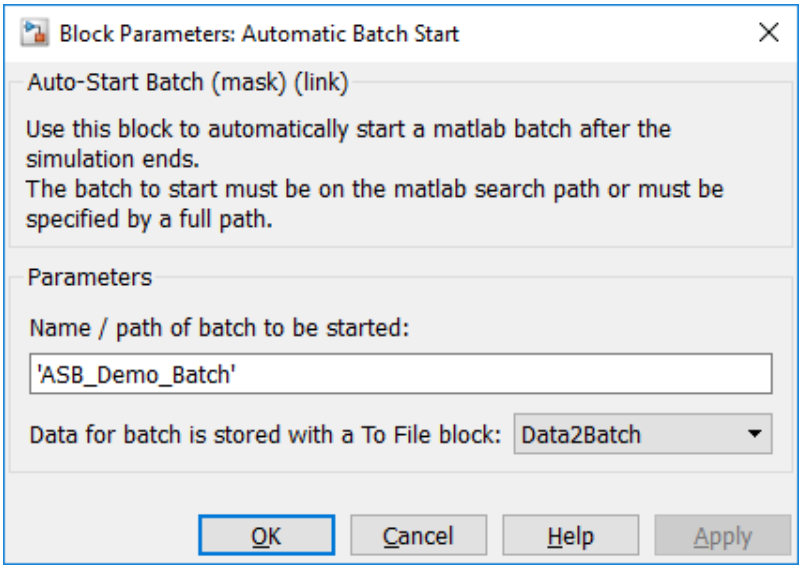
Start if quality is good for [s]	Automatically change trigger to 1 when all signals are good for the specified amount of seconds.
Example Simulink model	gSQCheckDemo.mdl

AutoStartBatch



Description The **AutoStartBatch** block allows you to start a MATLAB function for data processing (Batch) when the Simulink model ends. The **AutoStartBatch** block requires a MATLAB **To File** block in the Simulink model, which is used as intermediate data storage.
Once the Simulink model ends, the specified MATLAB function will be started and a workspace variable *AutoStartBatchData* will be created, which holds the acquired data. Use this variable to access your data.

Dialog box

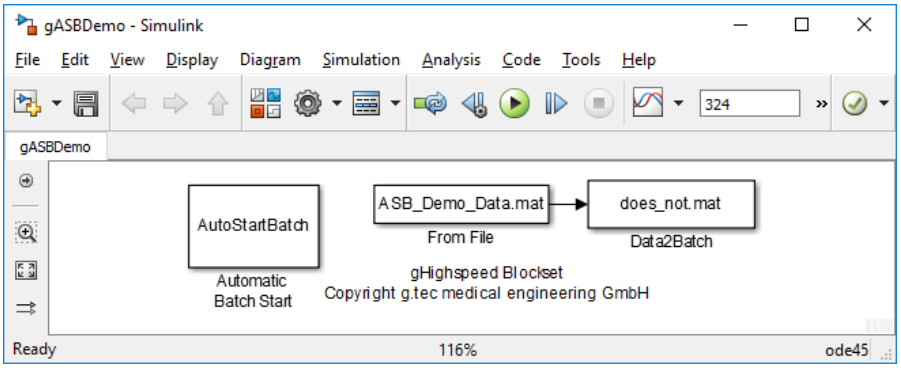


Double click on the **AutoStartBatch** block to edit the parameters

Name / Path of batch to be started	Define the batch file name with this parameter. For batch files not located on the MATLAB search path use the full file path.
Data for batch is stored in To File block	Define the **To File** block in the Simulink model from which data should be passed to the MATLAB function
Example Simulink model	gASB_Demo.slx

Example

1. Enter `gASBDemo` in the MATLAB command window to open the **gASBDemo.slx** Simulink model.
2. The following Simulink model which demonstrates the basic functionality of the **AutoStartBatch** opens:



3. Double click the **AutoStartBatch** block and check that the right To File block is selected.
4. Open the demonstration batch file by typing `edit ASB_Demo_Batch` to the MATLAB command window.
5. Click  to run the Simulink model. Due to the fact that there is no **Real Time Clock** block in the model, the simulation only lasts a few seconds. After the simulation has ended, the ASB_Demo_Batch batch should be called and result in an evoked potential analysis plot.

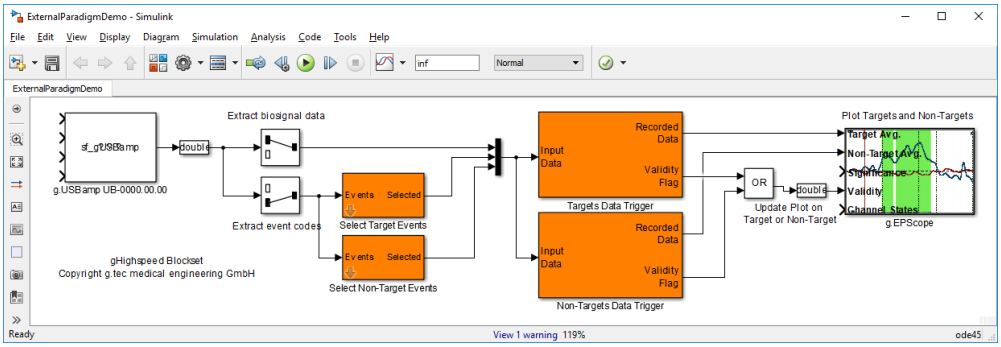
Interface to external stimulation units

g.HIsys interfaces external stimulation units like *Presentation* and *E-Prime* using event codes sent via parallel port. The parallel port is connected to the digital IO module of the g.tec amplifier (available for g.USBamp, g.HIamp, g.Nautilus) using an amplifier specific adapter cable (see product catalogue). The amplifier driver acquires the parallel port data as well as the biosignal raw data synchronously and streams it to MATLAB Simulink.
The **Select Events** block (pg. 46ff) can be used to select specific event codes only and use these event codes for triggering the biosignal raw data.

Example E-Prime

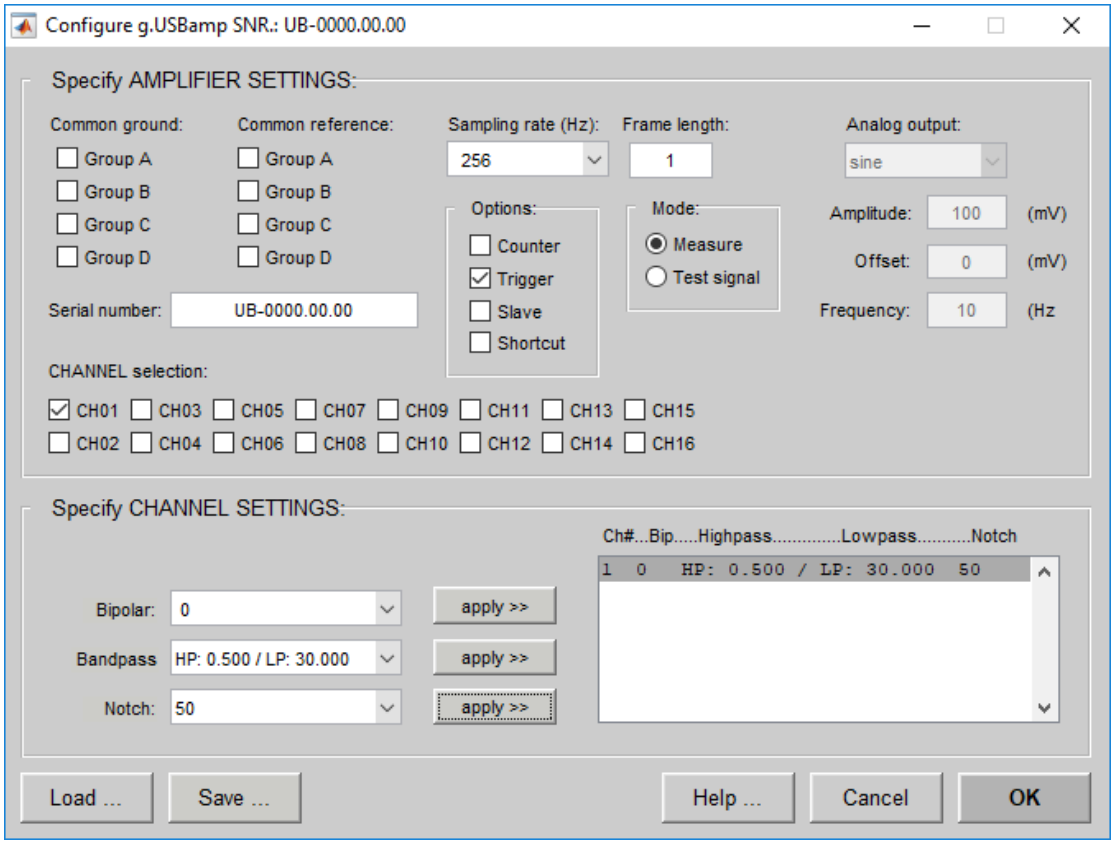
1. Enter `ExternalParadigmDemo` in the MATLAB command window to open the **ExternalParadigmDemo.slx** Simulink model.

2. The following Simulink model opens:



3. Open your g.tec Highspeed Library installation folder and browse to **Examples – External Paradigms** folder. Double click **ParallelPortDemo_E_Prime.es** to open the E-Prime demonstration project (expecting that E- Prime is installed on your system). If you use a newer version of E- Prime, follow the instructions on the screen to upgrade the project file.

4. Double click **g.USBamp** block and configure it according to the settings below:



5. Start the experiment in the E-Prime GUI. The paradigm will generate packets of event codes. Each packet contains the event codes 1 to 5 which are sent consecutively with a time interval of 400 milliseconds. The event code packets are sent with an interval of 3 seconds.

6. Click  to start the Simulink model.

The model splits the incoming data into raw data and event code data. The target event codes 1, 3 and 5 as well as the non-target event codes 2 and 4 are selected using the **Select Events** block. The incoming raw data of channel 1 is triggered according to the selected target and non-target event codes and the trigger result is presented using **g.EPscope**.

EEGlab data import

Installation

To install the g.tec data importer for eeglab, copy the folder `gtecimport` located in

`C:\Program Files\gtec\gtecHS\EEGlab Importer`

to

`eeglab14_1_0b\plugins` folder of your eeglab installation.

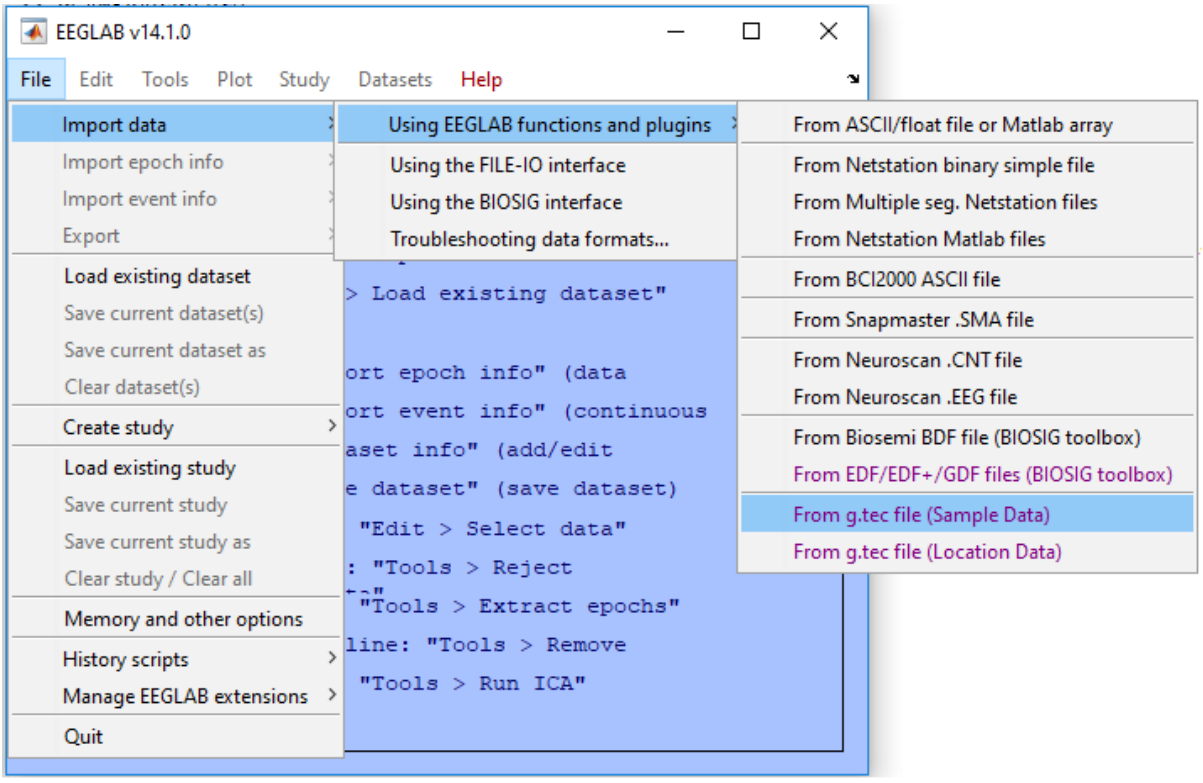
How to load *.mat files to eeglab

To open eeglab, enter `eeglab` to the MATLAB command window. While eeglab is loading a message

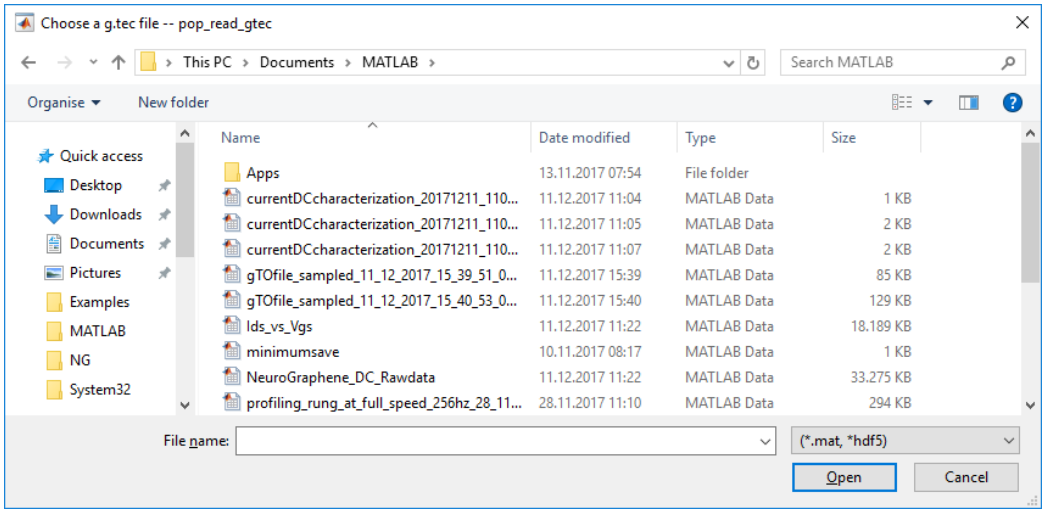
```
EEGLAB: adding "gtecimport" v3.16.01.bbb (see >> help
eegplugin_gtecimport)
```

should appear in the MATLAB command window. This message indicates that the g.tec data import plugin was loaded successfully to eeglab.

To load data to eeglab, use the **From g.tec file (Sample Data)** function which is located in the menu **File** – **Import data** – **Using EEGLAB functions and plugins**.

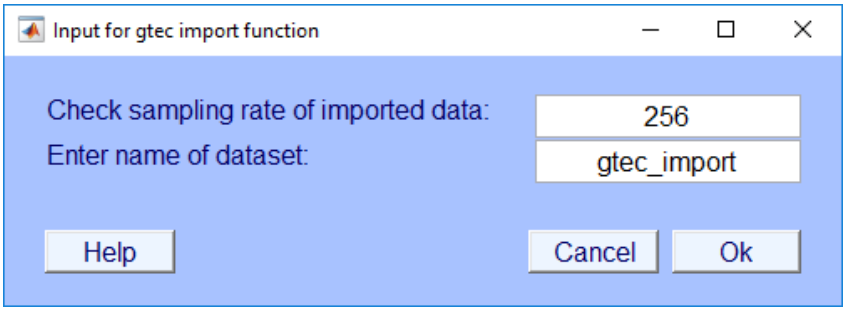


Clicking the function will bring up a file browser:



The file browser accepts only *.mat files, generated by MATLAB or Simulink and *.hdf5 files, generated by g.Recorder.

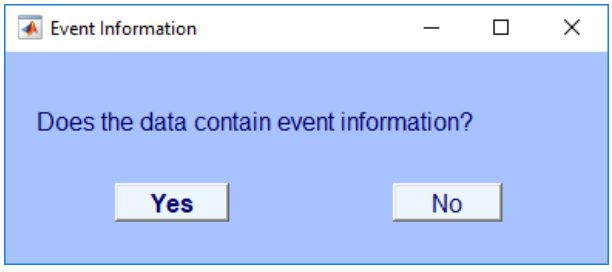
Choose a *.mat file and click **Open**. The following input dialog opens:



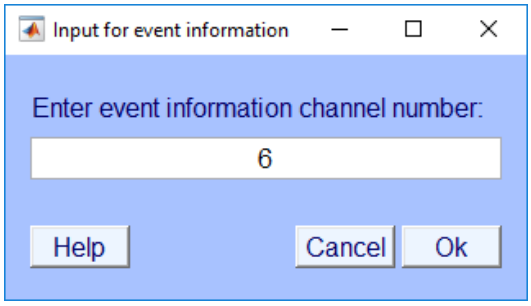
The dialog asks for the sampling rate of the acquired data and the name of the dataset. By default, the function reads the sampling rate of the data file and shows the value. The default name for the dataset is `gtec_import` which can be modified. Click **Ok** to continue.

Load event information from channels

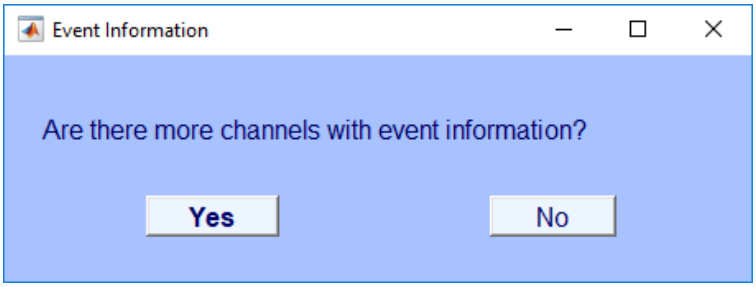
Due to the fact that *.mat files do not contain markers or events in the dataset (as *.hdf5 files do), event information is often provided in data channels which represent special conditions (trigger, events, …). The import function prompts if the loaded data contain such event information and if you want to import it.



Clicking **Yes** will bring up the next screen for the configuration of the event channels. Enter the channel number of the channel which contains event information and click **Ok**.



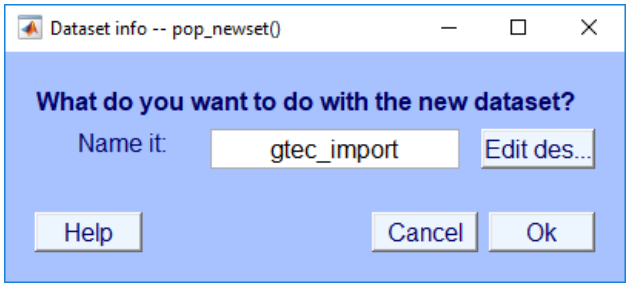
Due to the fact that the loaded data file consist of more channels containing event information, the importer asks if there are more channels with event information.



By clicking **Yes** the input dialog for the event channel is opened again.
All event information for all channels are combined, e.g. if you have 2 channels and 50 events the final event information will consist of 100 events.
This procedure goes on as long as there are channels with event information and the dialog above is not canceled with **No**.

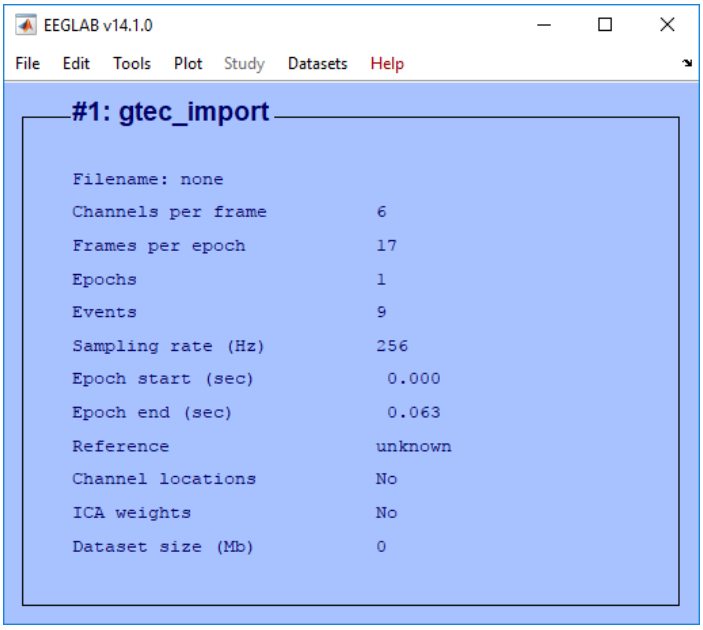
For information on how the events are named please refer to eeglab help on function `pop_chanevent`.

The dataset is added to eeglab, which prompts the following window:



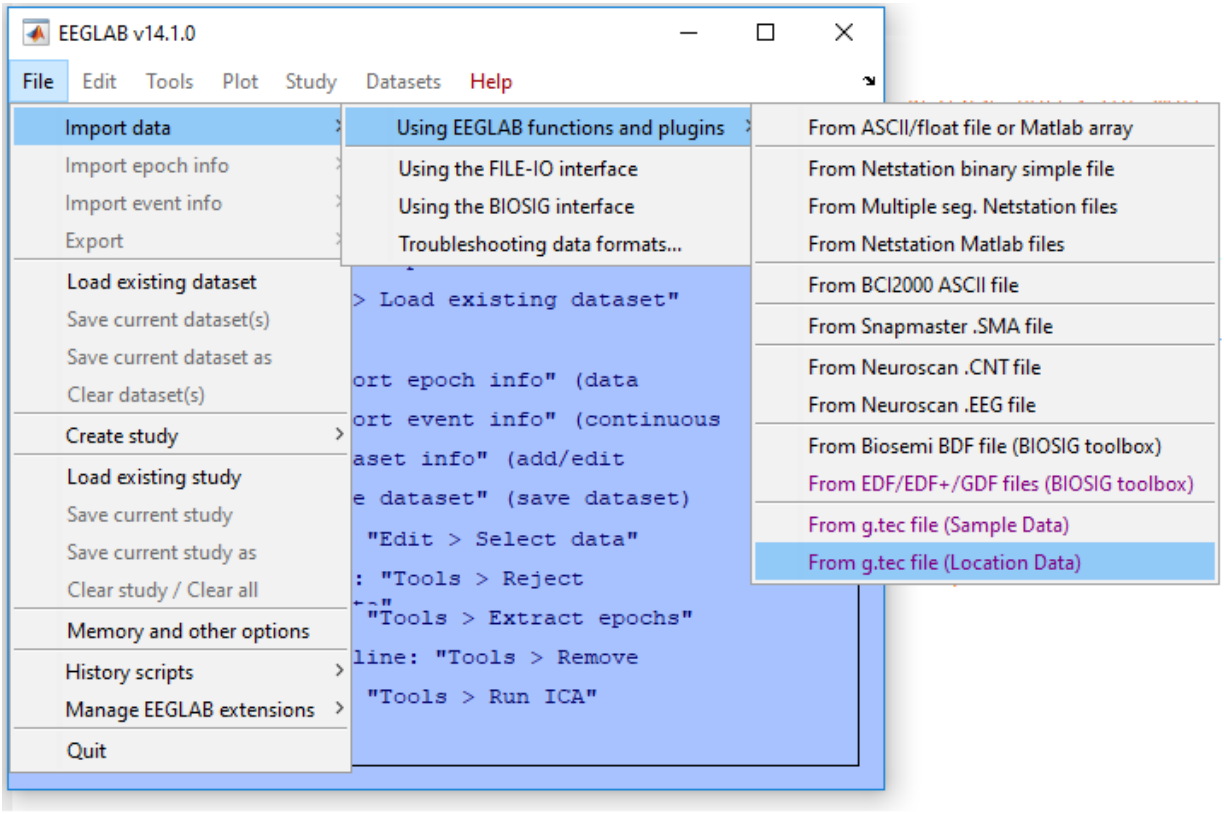
Click **Ok** to load the dataset to eeglab.

The data is now loaded to eeglab. The importer created a new dataset called **gtec_import** which consists of 3 data channels, a sampling rate of 250 Hz and a dataset length of 60.88 seconds.

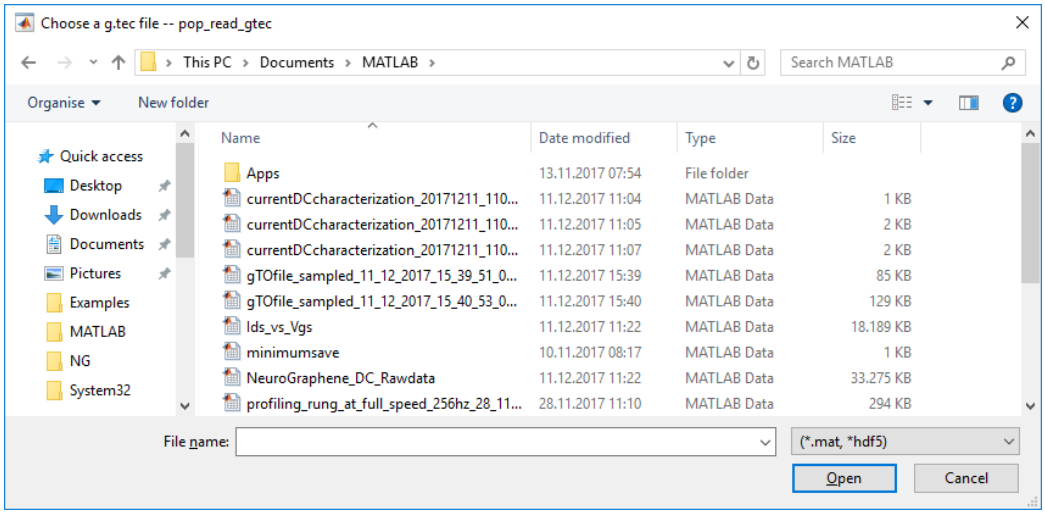


How to load electrode positions to eeglab
Once you have loaded your data correctly to eeglab one may want to add electrode position information to eeglab.

To load electrode position information to eeglab, use the **From g.tec file (Location Data)** function which is located in the menu **File** – **Import data** – **Using EEGLAB functions and plugins**.



Clicking the function will bring up a file browser:



The file browser only accepts MATLAB *.xyz files.
The MATLAB *.xyz file for an 8 channel electrode montage looks like this:

```
1 0.0000 1.0000 0.0000 FPz
2 0.0000 0.7100 0.7100 Fz
3 0.0000 0.0000 1.0000 Cz
4 -0.0000 -0.7100 0.7100 Pz
5 0.6900 0.3800 0.6200 FC4
6 -0.6900 0.3800 0.6200 FC3
7 0.6900 -0.3800 0.6200 CP4
8 -0.6900 -0.3800 0.6200 CP3
```

The basic format of this file is
`<CH Nr.> <X-Coord> <Y-Coord> <Z-Coord> <CH Name>`

Choose a file and click **Open**. If the import succeeded, the eeglab panel should show **Yes** next to **Channel locations**.

NOTE

If the channel numbers does not match, the import is ignored by eeglab. Check the MATLAB command window for the error message `Wrong channel structure size, changes ignored`*. If this message occurs try loading an electrode position file with the correct number of channels.*

Create *.xyz files from g.tec montage files

In the g.tec g.BSanalyze environment montage files are specified using g.MONcreator. g.MONcreator is a g.tec tool which allows you to specify electrode location information. If you have g.BSanalyze installed you can start g.MONcreator by typing `gMONcreator` to the MATLAB command line (see help file of g.MONcreator for detailed help).

After you have created and saved your g.tec montage, this montage can be converted to the MATLAB *.xyz format by using the montage2EEGlab function which is located in the **gtecimport** folder located in the **plugins** folder of your eeglab installation.

Load the created montage to the MATLAB workspace (variable Mon is created), make sure that the function is on the MATLAB path or you are in the same folder, and call the function

```
montage2EEGlab(Mon, 'converted_montage.xyz');
```

which creates the *.xyz file loadable to eeglab.

Help

g.tec Highspeed provides printable documentation.

The printable documentation is stored under

```
C:\Program Files\gtec\gtecHS\help
```

as

```
gHIsysLibraryDescription.pdf
```

Use Acrobat Reader to view the documentation.

Product Page

Please visit our homepage www.gtec.at for

- Update announcements
- Downloads
- Troubleshooting
- Additional demonstrations


g.tec
GUGER
TECHNOLOGIES
contact information
g.tec medical engineering GmbH
Sierningstrasse 14
4521 Schiedlberg
Austria
tel. +43 7251 22240
fax. +43 7251 22240 39
web: www.gtec.at
e-mail: office@gtec.at