Stimulation

ContStim – CONTinousSTUMulation

ContStim is used to generate visual stimuli (e.g., drifting bars, drifting/rotating square grating) on a monitor screen, to generate auditory stimuli via a sound card or to produce TTL sequences to control other auxiliary devices. Also, it generates synchronization signals, which are constantly transmitted through DIO or UDP channels to the acquisition computer and recorded by **ContImage**.

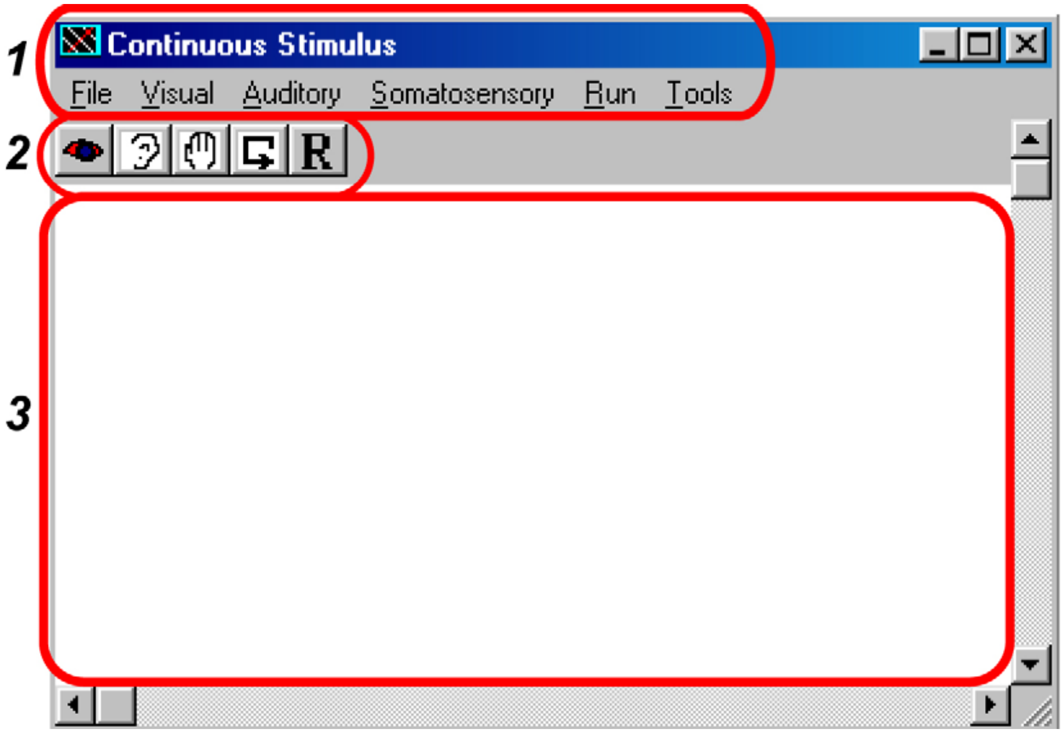

Figure 1. *Components of ContStim.*

Components of ContStim

1 Menu bar. Function: selection of modalities, and stimuli.
2 Main tool bar. Function: reinitialization of the stimuli, starting stimulation.
3 Log window. Function: error (red) and warning (blue) logging. Other types of logged info are colored black for normal events, green for debugging messages. The latter ones can be ignored.

Selection of the necessary modality and stimulus is made trough the Menu. Modifications of the particular stimulus parameters are made by editing the initialization file. The initialization file can be placed anywhere in the system, its full path should be referenced in the master initialization file: `C:\WINNT\ContStimMaster.ini`. The simplest version of `ContStimMaster.ini` has a reference to the actual initialization file only:

```
[contstim]
IniFile=C:\OpticalImaging\contstim.ini
```

The `ContStimMaster.ini` may contain default initialization values of stimuli parameters as well. These values are used if the reference "`IniFile`" file is not specified or cannot be found. It is recommended to have a few initialization file prepared for different modalities, e.g. "`ContStimVisual.ini`", "`ContStimAuditory.ini`", and so on, and/or different stimuli and then modify the link "`IniFile`" in the master file `C:\WINNT\ContStimMaster.ini` only.

The initialization file may contain information about many kinds of stimuli. The particular stimulus to be generated is selected through the Menu (Figure 2). Some parameters may be repeated a few times in the initialization sections of different stimuli. The user should comment out the values that do not belong to the desired stimulus. It is safer to have individual initialization files prepared for each stimulus and to change the link in the master initialization file only.

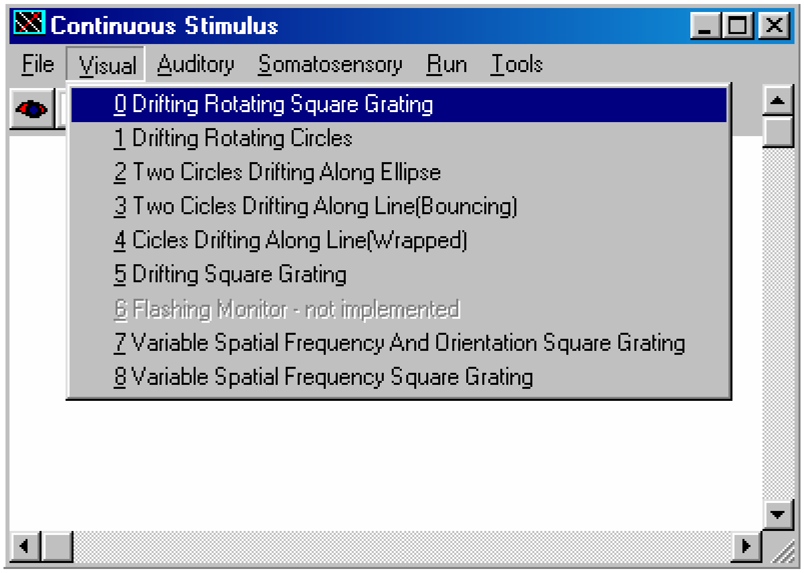
Continuous Stimulus
File
Visual
Auditory
Somatosensory
Run
Tools
0 Drifting Rotating Square Grating
1 Drifting Rotating Circles
2 Two Circles Drifting Along Ellipse
3 Two Cicles Drifting Along Line(Bouncing)
4 Cicles Drifting Along Line(Wrapped)
5 Drifting Square Grating
6 Flashing Monitor - not implemented
7 Variable Spatial Frequency And Orientation Square Grating
8 Variable Spatial Frequency Square Grating

***Figure 2**. Selection of the visual stimulus. (Some rarely used stimuli may not be present in your version of ContStim).*

Pressing Main tool bar "R" button or "Run" button in the menu or the "Space button" starts stimulus generation. Keyboard keystrokes control **ContStim** while it is generating visual stimuli (the program window is not seen on the screen). Pressing "a" (Abort) stops stimulation and pressing "p" (Pause) pauses stimulation. Pressing "r" (Run) continues stimulation.

Visual Stimuli

Useful visual stimuli provided with **ContStim** are:

1 Drifting rotating square grating (0).
2 Drifting rotating circles (1).
3 Circles drifting along line (4).
4 Drifting square grating (5).

Other stimuli are experimental (of no particular use).

Description of the initialization file: contstim.ini

contstim.ini has a structure of the Windows initialization files (.ini files). Section names are surrounded with square brackets, e.g. `[contstim]`. Other names are variables and values they are set to, e.g. `monitorwidthcm="40.0"`. **contstim.ini** has 3 sections: `[Sync]` for sunchronization options, `[contstim]` for visual stimuli, `[audio]` for auditory stimuli, and `[TTL]` for other types of stimulation, e.g. control of shutters. The variable names are not cASe-Sensitive. Any special character (e.g. “;”) inserted before the variable comments out the whole line.

Stimulus independent parameters

Synchronization section

The synchronization section `[Synch]` defines what channels are to be used for synchronization. There are two types of synchronization channels: DIO (direct wire connect between computers) and UDP (connection via Ethernet IP/TCP). The following is a typical `[Sync]` section

```
[Sync]

SyncDIOYes="0"              // Do not use DIO channel
SyncUDPYes="1"              // Use UDP channel

SyncUDPHost="grabber"       // The host to receive UDP synch
SyncUDPPort="8936"          // The port on the host to receive the synch
```

Note that the acquisition computer “listens” to preset port “8936”; this port value should not be modified. The `[sync]` section is rarely edited.

Visual stimuli section

The visual section [`contstim`] starts from description of the monitor parameters and settings. These are stimulus independent parameters.

```
[contstim]
MonitorResolutionWidth="800"
MonitorResolutionHeight="600"
refreshratehz="120.0"
monitorwidthcm="40.0"
monitorheightcm="30.0"
screendistancecm="40.0"
```

`MonitorResolutionWidth` and `MonitorResolutionHeight` – resolution of the monitor XVGA mode in pixels. Standard values are 640x480, 800x600, 1024x768, 1280x1024, and higher. It is not recommended to choose high-resolution modes since it puts a lot of load on to the CPU and as a result some frames will be skipped leading to a jerky motion of the stimuli. A reasonable choice is 800x600.

`RefreshRateHz` – refresh rate of the current mode measured in Hertz. **ContStim** does not set the refresh rate. The program uses this value to set proper timing for frame display. The user through the ControlPanel should set the actual refresh rate. Standard values depend on the monitor make and model. They usually vary between 60 and 120Hz or even higher. It is not recommended to use very low refresh rates such as 60Hz. It will lead to flickering. It has been reported that some neurons in the cat visual system can follow this flickering. The values from 80Hz and up are reasonable. Using 120Hz insures stability of the image on the screen and is recommended. Combination of high-resolution with high refresh rate can be very CPU time consuming. If motion of the stimuli is not uniform or jerky either resolution or refresh rate or both should be lowered.

`MonitorWidthCm` and `MonitorHeightCm` – physical dimensions of the monitor screen in centimeters. They should be measured with a ruler for a monitor in use, written to the initialization file and never modified. A typical 22`` monitor has dimensions 40x30cm. Nonetheless the actual dimensions of the visible part should always be measured manually.

ScreenDistanceCm – physical distance from the screen to the subject (subject's nose) in centimeters. This parameter is usually set to 40-50cm for ferrets and cats, and to 25cm for mice.

Next group of parameters describes the size and position of the bounding box. Stimuli consisting of bars will be displayed inside this rectangular box only.

```
BoundBoxCenterXDeg="10"
BoundBoxCenterYDeg="10"
BoundBoxCenterWidthDeg="20.0"
BoundBoxCenterHeightDeg="20.0"
```

BoundBoxCenterXDeg and BoundBoxCenterYDeg – specify position of the center of the bounding box on the screen measured in degrees. The reference point is located in the center of the screen. The example above puts the center of the box 10 degrees to the right and 10 degrees up from the center of the screen.

BoundBoxCenterWidthDeg and BoundBoxCenterHeightDeg – specify the size of the box in degrees. The example above makes a box of 20x20 degrees. The stimulation will be limited to right-top quadrant of the screen. Setting value of either of these variables to –1 will unbind stimulation in the corresponding dimension. If both are set to –1 the whole screen is used.

The final stimulus independent group of parameters defines foreground and background measured in percent.

```
contrastpct="50"
backgroundpct="50"
```

The formulae used to calculate actual foreground and background gray-scale color are

```
    Foreground=(backgroundpct+contrastpct/2)*2.55
    Background=(backgroundpct-contrastpct/2)*2.55
```

The example above sets the foreground to 191 and the background to 64, where 255 corresponds to lightest screen and 0 to darkest. The positive values of give light stimulus elements on a dark screen (ON-OFF type). Use of negative contrastpct is permitted; it gives bright background and dark foreground (OFF-ON type). Setting contrastpct to 100 and backgroundpct to 50 gives highest contrast.

Stimulus dependent parameters

The first two variables of every stimulus define number of cycles this stimulus will be displayed for:

```
iniangle="-1.0"
finangle="11.0"
```

Originally the program was written to generate rotating square gratings only, that is why the variables are called `iniangle` (INItial ANGLE) and `finangle` (FINal ANGLE). The number of cycles is the difference between `finangle` and `iniangle`. The number of cycles to display will be set to 12 by the example above. The display time depends on the number of cycles and on the length of one cycle, which is by another (stimulus dependent) variable.

Parameters for drifting-rotating square grating

This stimulus is used to evaluate orientation and direction selectivity. The stimulus is the full field square grating with two degrees of freedom: (1) slow rotation of the grating and (2) fast drift of the grating. The following parameters describe the rotation:

```
anglecycle_min="1.0"
clockwise="-1"
```

`AngleCycle_Min` – specifies rotation speed in cycles per minute. The example above sets it to 1 cycle per minute. This variable should take only positive values. Direction of rotation is controlled by another variable `clockwise`. Setting `clockwise` to 1 will rotate the grating clockwise and to –1 will rotate it counterclockwise. The example above will generate a grating rotating counterclockwise.

The pattern and the fast drift are described by the following three parameters:

```
tfcycle_sec="2.0"
sfcycle_deg="0.2"
striperatio="0.5"
```

`tfcycle_sec` – temporal frequency measured in cycles per second. Typically used value for cats and ferrets is 2.0.

`sfcycle_deg` – spatial frequency measured in cycles per second. Typical value for cats and ferrets is 0.2.

`striperatio` – this parameter specifies the duty cycles. It is a ratio of the width of the bright bar to the whole spatial cycle. The examples sets `striperatio` to 0.5, which means that duty cycle is 50% or that the bright and dark bars have equal width. This is the value used in cat experiments.

Parameters for drifting square grating

This stimulus is used to evaluate topography of the visual cortex (retinotopy). The stimulus is a simplified version of the previous stimulus (drifting-rotating grating); it does not have the rotation parameters. Parameters specific to this stimulus are

```
stripesizedeg="2.0"
direction_deg="90.0"
```

`StripeSizeDeg` – carries function similar to `striperatio`. It defines width of the bars in degrees. Both can be used interchangeably. `StripeSizeDeg` is more convenient since it explicitly defines the width.

`Direction_Deg` – specifies direction of the drift in degrees. 0 – vertical bars moving to the right, 90 – horizontal bars moving upward, 180 – vertical bars moving to the left, and 270 – horizontal bars moving downward. All other values (oblique) are acceptable as well.

Typical values for temporal and spatial frequencies are

```
tfcycle_sec="0.125"
sfcycle_sec="0.01"
```

These values are used for mouse imaging.

Auditory stimuli

The modality should be set to auditory in the Menu (Figure 3) for generation of sound sequences.

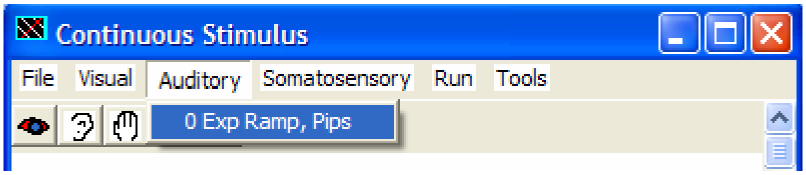

Figure 3. *Selection of the auditory stimulus.*

Common parameters for WAV file generation

The following three entries define parameters for WAV file generation. Values of these parameters are hardware dependent.

```
bits=3
sample_rate=5
channel=0
```

`bits` defines the number of bits per sample by formula

bits_per_sample=(bits+1)*8

for a regular sound card `bits` should set to 1, thus providing resolution of 16bits; for a Hi-Fi card, such as Lynx2, it should be set to 3, providing 32 bit resolution. Only 24 bits are actually used by the driver.

`sample_rate` defines sampling rate. It may take values ranging from 0 to 7 with the following translation to the actual sampling rates measured in Hertz

0 - 11025, 1 - 22050, 2 - 44100, 3 - 88200,
4 - 96000, 5 - 192000, 6 - 200000, 7 - 250000

`sample_rate` should be set to 5 (192000Hz sampling rate) for Lynx2 sound board.

The parameter `channel` controls how many channels to be used. It should be set to 0 for mono recording and to 1 for stereo. The stereo channels are identical in the current realization of the program.

Parameters controlling pip envelope

Parameter `pattern` should be set to 1 to generate tone pips. Setting it to 0 will make a flat envelope, basically no envelope, which is used to generate FM sweeps.

The following entries define parameters pip envelope (in seconds).

```
timeinterval_s="0.25"
timeduration_s="0.03"
timerise_s="0.005"
```

`timeinterval_s` - onset-to-onset time (250ms here).
`timeduration_s` - duration of the tone pip excluding rise and fall (30ms here).
`timerise_s` - rise and fall time (5ms here).

An extra parameter `rise` controls the shape of the rise and fall.
`rise=0` - half-sinosoidal rise/fall
`rise=1` - exponential rise/fall
Typically it is set to 0.

Parameters controlling pip sequence generation

```
timetotal_s="4.0"
frequencylow_hz="32000"
frequencyhigh_hz="2000"
```

`timetotal_s` – length of the staircase (4sec here). This parameter divided by `timeinterval_s` defines how many tone pips are in the sequence.

`frequencylow_hz` – frequency of the first tone in the sequence (32000Hz here).

`frequencyhigh_hz` – frequency of the last tone in the sequence (2000Hz here). This example generates descending staircase.

If needed the staircase can be padded with zeros. The following two parameters specify how many seconds of silence (blank) should be added before and after the sequence respectively

```
timepresilence_s="0.0"
timepostsilence_s="0.0"
```

Note that the total length of a sequence is

```
timepresilence_s+ timetotal_s+ timepostsilence_s
```

Typically these parameters are not used, that is set to zero.

Parameter `oscillator=1` controls the carrier. It should be set to 1 which sets the carrier to a sinusoid.

Parameter `scheme` controls the kind of frequency gradient from pip to pip. It is linear if `scheme=0` and exponential if `scheme=1`. Typically it is set to 1.

Parameter `randomize` specifies whether the sequence should be randomized. Setting it to zero generates and ordered sequence of tone pips. Setting it to any other value, e.g. 1, generates a random sequence of tone. Note that the frequencies used in both randomized and ordered sequences are the same. To set the random number generator to a specific value and, thus, generate the same random sequence every time one should use parameter `seed`, for example

```
seed=1357
```

Parameters controlling sound intensity

Parameter `intensity_db` controls attenuation of the generated tones. The sounds of highest intensity are generated if this parameter is set to zero. Values higher than zero attenuate generated tones. This parameter controls level of attenuation relative to the loudest state; e.g `intensity_db="20.0"`.

To generate tones of specific intensity an adjustment file should be used. Parameter `intensity_adjustment` should be set to 1 to use the adjustment procedure. Another variable `adjustmentfile` should specify a correct full-path of the adjustment file, e.g.

```
adjustmentfile="C:/Temp/speaker_calibration.txt"
```

Parameter `intensity_dbspl` can be used now to generate tones of specific sound intensity measured in dB SPL; e.g. `intensity_dbspl="40.0"`. See **Appendix C** for an example of an adjustment file.

Parameters controlling the number of cycles

Parameter `nrepeats` specifies the number of times the stimulus sequence should be played. E.g. `nrepeats=150` will repeat the sequence 150 times, making a total stimulation time of 600 seconds (10 minutes) if `timetotal_s="4.0"` and `timepresilence_s=timepostsilence_s=0`.

Another parameter (`nrepeats_infile`) specifies how many times the sequence should be repeated in the wave file. Typically this parameter is set to one.

Playing a prerecorded waveform

A prerecorded waveform can be played by the program. Variable `playfromfile` should be set to 1 and another variable (`playfile`) should point to a correct wave file name; e.g. `playfile="C:/Temp/wave.wav"`. Note that setting `playfromfile` to zero will not play the requested file. If this way of stimulus generation is selected all other parameters in the ini file are ignored except `intensity_db` which attenuates (positive values) or amplifies (negative values) the prerecorded sound.

Saving log and wave files

The following two pairs of variables control saving of the generated wave file and log file respectively.

```
savewave=1
wavefile="C:/Temp/wave.wav"

savelog=1
logfile="C:/Temp/wave.log"
```

The log file contains information about the generated sound sequence such as precise values of tone pip frequencies.

Upon completion of the stimulus program saves exact timing of the sound generation in the preset file `C:/Temp/time.txt` .

Appendix A

Example of complete **contstim.ini** file. Note that all entries for the topography stimulus are preceded with a semi-colon, which turns them into a comment. This way the user can keep all stimuli in one file and uncomment the entries for the required stimulus only.

```
[contstim]
MonitorResolutionWidth="800"
MonitorResolutionHeight="600"
refreshratehz="120.0"
monitorwidthcm="40.0"
monitorheightcm="30.0"
screendistancecm="40.0"

BoundBoxCenterXDeg="10"
BoundBoxCenterYDeg="10"
BoundBoxCenterWidthDeg="-1.0"
BoundBoxCenterHeightDeg="-1.0"

contrastpct="50"
backgroundpct="50"

;;Orientation stimulus
iniangle="-1.0"
finangle="11.0"
anglecycle_min="1.0"
clockwise="-1"
tfcycle_sec="2.0"
sfcycle_deg="0.2"
striperatio="0.5"

;;Topography stimulus
;iniangle="0.0"
;finangle="90.0"
;tfcycle_sec="0.125"
;sfcycle_sec="0.01"
;stripesizedeg="2.0"
;direction_deg="90.0"
```

Appendix B

Example of an auditory **contstim.ini** file.

```
; bits_per_sample=(bits+1)*8, use bits=0, 1, or 3(fake 32bit) only
bits=3
; Channels 0 - mono, 1 - stereo
channel=0
; Sample rate 0 - 11025, 1 - 22050, 2 - 44100, 3 - 88200,
; 4 - 96000, 5 - 192000, 6 - 200000, 7 - 250000 Hz
sample_rate=5

timetotal_s="3.0"
timeoffset_s="0.0"
timeinterval_s="0.125"
timeduration_s="0.03"
timerise_s="0.005"
; this time is extra to timetotal_s
timepresilence_s="0.0"
; this time is extra to timetotal_s
timepostsilence_s="0.0"

frequencylow_hz="32000"
frequencyhigh_hz="2000"

; # of cycles to be played
nrepeats=200

; # of cycles to be saved in file
nrepeats_infile=1

; Intensity 0 - is max, positive values lower volume (attenuation)
intensity_db="20.0"

; Actual Intensity dB SPL, used with adjustment file only
intensity_dbspl="40.0"

; 0 - no adjustment
intensity_adjustment=0

; Adjust intensity to compensate for speaker attenuation
;adjustmentfile="C:/Temp/speaker_calibration.txt"

playfromfile=0
;playfile="C:/Temp/wave.wav"
```

```
; 0 - do not save
savewave=1
wavefile="C:/Temp/wave.wav"

; 0 - do not save
savelog=1
logfile="C:/Temp/wave.log"

; Rise type: 0 - sinus, 1 - exponential
rise=0

; 0 - flat, 1 - pips
pattern=1

; 0 - linear, 1 - exponential rise of frequency
scheme=1

; 0 - none(just envelope), 1 - sinus
oscillator=1

; Randomize pips
randomize=0
;;; Random number generator seed
seed=1357
```

Appendix C

Example of an intensity adjustment file. Lines starting from ‘#’ are comments, The line starting from #format specifies the format of the entries. The entries are (1) index, (2) test frequencies, (3) attenuation required to make the sound of preset intensity, (4) intensity (preset in this case).

```
# Frequency(Hz)     Attenuation(dB)     Intensity(dB SPL)
#format index  frequency attenuation    intensity
1 1000.000000       48.0      70.0
2 1118.286868       48.0      70.0
3 1250.565520       48.0      70.0
4 1398.490998       46.0      70.0
5 1563.914119       49.0      70.0
6 1748.904622       48.0      70.0
7 1955.777073       48.0      70.0
8 2187.119818       48.0      70.0
9 2445.827371       48.0      70.0
10 2735.136631      47.0      70.0
11 3058.667377      48.0      70.0
12 3420.467562      47.0      70.0
13 3825.063958      49.0      70.0
14 4277.518794      50.0      70.0
15 4783.493096      49.0      70.0
16 5349.317514      48.0      70.0
17 5982.071529      45.0      70.0
18 6689.672036      38.0      70.0
19 7480.972391      37.0      70.0
20 8365.873186      38.0      70.0
21 9355.446125      42.0      70.0
22 10462.072548     49.0      70.0
23 11699.598344     40.0      70.0
24 13083.507192     41.0      70.0
25 14631.114283     40.0      70.0
26 16361.782970     40.0      70.0
27 18297.167035     36.0      70.0
28 20461.481621     43.0      70.0
29 22881.806201     33.0      70.0
30 25588.423396     25.0      70.0
31 28615.197862     24.0      70.0
32 32000.000000     29.0      70.0
```