Manual Actions Expressive System (MAES)

Version 1.01
© Rajmil Fischman, 2012-14

User Guide

Manual Actions Expressive System (MAES) aims to enable music creation and performance using natural hand actions (e.g. hitting virtual objects, shaking them, etc.). Gestures are fully programmable and result from tracking and analysing hand motion and finger bend, potentially allowing performers to concentrate on natural actions from our daily use of the hands. It is designed to work with the P5 Glove [1] and has been used to create music for this device (see https://vimeo.com/55093629). Nevertheless, it can also be used without a controller, offering a standard graphic interface for real time control of synthesis and processing. Furthermore, the MAX external that communicates with the P5 glove can be easily replaced by any other external or subpatch that provides all or part of the corresponding data (position X,Y,Z; orientation X,Y,Z; velocity X,Y,Z and individual finger bend), with relatively little effort (see the Technical Appendix at the end of this guide).

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details, https://www.gnu.org/licenses/licenses.html.
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>2</td>
</tr>
<tr>
<td>Installation</td>
<td>4</td>
</tr>
<tr>
<td>Installing and Running the Windows Standalone</td>
<td>4</td>
</tr>
<tr>
<td>Installing and Running the Mac OSX Standalone</td>
<td>4</td>
</tr>
<tr>
<td>Opening the Source Patch</td>
<td>4</td>
</tr>
<tr>
<td>System Requirements</td>
<td>5</td>
</tr>
<tr>
<td>Graphic Interface</td>
<td>6</td>
</tr>
<tr>
<td>Input/output</td>
<td>7</td>
</tr>
<tr>
<td>Synthesizers/microphone capture</td>
<td>8</td>
</tr>
<tr>
<td>Audio file players</td>
<td>8</td>
</tr>
<tr>
<td>Pre-loaded lists</td>
<td>8</td>
</tr>
<tr>
<td>Lists and presets</td>
<td>9</td>
</tr>
<tr>
<td>Routing matrix</td>
<td>9</td>
</tr>
<tr>
<td>Panning parameters</td>
<td>10</td>
</tr>
<tr>
<td>Spectral shift</td>
<td>11</td>
</tr>
<tr>
<td>Linking shift processors</td>
<td>12</td>
</tr>
<tr>
<td>QLists</td>
<td>12</td>
</tr>
<tr>
<td>Spectral stretch</td>
<td>13</td>
</tr>
<tr>
<td>Linking stretch processors</td>
<td>14</td>
</tr>
<tr>
<td>QLists</td>
<td>14</td>
</tr>
<tr>
<td>Time-stretch and spectral blur</td>
<td>14</td>
</tr>
<tr>
<td>Spectral Blur</td>
<td>14</td>
</tr>
<tr>
<td>Linking time-stretch/blur processors</td>
<td>16</td>
</tr>
<tr>
<td>QLists</td>
<td>16</td>
</tr>
<tr>
<td>Granulator</td>
<td>17</td>
</tr>
<tr>
<td>Global parameters</td>
<td>17</td>
</tr>
<tr>
<td>QLists</td>
<td>18</td>
</tr>
<tr>
<td>Source</td>
<td>18</td>
</tr>
<tr>
<td>Cloud attributes</td>
<td>19</td>
</tr>
<tr>
<td>Grain envelope</td>
<td>21</td>
</tr>
<tr>
<td>Panning</td>
<td>22</td>
</tr>
<tr>
<td>Formant bank</td>
<td>22</td>
</tr>
<tr>
<td>Global parameters</td>
<td>23</td>
</tr>
</tbody>
</table>
QLists 24
Individual formant parameters 25
Linking formants 26
Glove control (Windows only) 27
Scaling 27
Data Smoothing 28
Calibration 28
Hand shapes 30
Save and Load 31
Tracking 31
Gesture creation and mapping 32
Mapping conditions 35
Presets 43
Storing presets 44
Saving/loading presets and other files 45
Viewing stored data 45
CPU and audio status 46
Troubleshooting 47
Technical Appendix - Replacing the P5 module by an alternative controller 48
Notes 50
Installation
The original MAX source patch and standalone versions for windows and Mac OSX are provided. **PLEASE NOTE that Glove Control will only be functional in 32-bit WINDOWS version of MAX.** It will not be functional in OSX (there are no available P5 glove drivers) or the 64-bit version of MAX (future implementation planned). However, the P5 glove external can be replaced in the source patch by any other external/subpatch that provides similar parameters: see the Technical Appendix at the end of this guide.

Installing and Running the Windows Standalone
1. Copy the entire folder `MAES_Standalone_WINDOWS` to an appropriate location in the computer.
2. Open the copied folder `MAES_Standalone_WINDOWS` in the computer.
3. Double click on `MAES.exe` to run the standalone.

Installing and Running the Mac OSX Standalone
1. Copy the entire folder `MAES_Standalone_MacOSX` to an appropriate location in the computer.
2. Open the copied folder `MAES_Standalone_MacOSX` in the computer.
3. Double click on `MAES.app` to run the standalone.

Opening the Source Patch
MAES is a MAX version 6.xx **32-bit** project. There are separate sources for Windows and Mac OSX, as follows:

**Windows:** `MAES_Max_Sources_WINDOWS`

**Mac OSX:** `MAES_Max_Sources_MacOSX`

In order to open the source:

1. Copy the sources folder `MAES_Max_Sources` to an appropriate location in the computer.
2. Run MAX.
3. In MAX, open the project `MAES.maxproj` inside the copied folder `MAES_Max_Sources`.

Top 🕑  Contents 🕑
System Requirements
Ideally, a fast processor should be used (e.g. i7 quad core). However, it is also possible to run MAES in a slower system by reducing the CPU workload in MAX (see Troubleshooting at the end of this guide).

The patch can be run as an executable (Windows: MAES.EXE or Mac OS: MAES.APP), or as a runtime MAX patch (MAES.MXE both in Windows and Mac OS) using MAX Runtime version 6 upwards (available for free from Cycling74’: http://cycling74.com/downloads/).

IMPORTANT NOTE
While the corresponding patches will run in both Windows and OS, P5 glove tracking is only possible in Windows and OS9 since there are no available drivers for OSX.

Screen Resolution
The size of the full patch is approximately 1770 x 920 pixels. Therefore, it will appear in its entirety in HD resolution, but will extend beyond the screen at lower resolutions, requiring resizing of the window to enable scrolling.
Graphic Interface

Input/output
Glove control
Routing matrix
Synthesizers/microphone capture
Presets
Spectral shift
Spectral stretch
Time-stretch

Panning parameters
Audio file players
Granulator
Formant bank
CPU and audio status

Top Contents
Input/output

This panel offers the following functions:

- MAX audio ON/OFF switch (shortcut: ‘A’)
- Input LED monitor (mono).
- Output LED monitors (mono/stereo/5.1 surround/octophonic).
- Microphone limiter and noise gate: the noise gate can be switched ON or OFF [2].
- Input gain.
- Output gain.
- Output channels:

  mono, stereo, 5.1 surround, octophonic A, octophonic B

It is also possible to record the output as an audio file with the same number of channels. In order to do so:

1. Click on ![Folder Icon] to create an output file.
2. Click on ![Record Icon] to begin recording.
3. Click on ![Stop Icon] to stop recording.
Synthesizers/microphone capture
The synthesizers offer a number of sound generators, microphone capture with an option to apply an LFO modulator. These signals can be sent directly to a mono audio output in channel 1, which is mainly useful for testing purposes. Normally, the signals will be routed to the processors or directly to the panning and output via the routing matrix.

ON/OFF switch to send test mono signals to the output (channel 1)

Direct output

Signal type: none (OFF), microphone, sinewave, pulse train, sawtooth, noise

Frequency (when applicable) LFO frequency

Amplitude LFO ON/OFF switch

Audio file players
Audio files of up to 8 channels can be played and sent directly to MAX audio outputs, preserving the original spatialisation of each file. Also, the first two channels of a file can be routed to processes and further panning via the routing matrix.

Direct output Looped toggle Add file to list

Open file Gain Play Current file ID number (triggers playback)

Edit, load and save list

Pre-loaded lists
It is possible to create file lists, with files identified by an ID number, as follows:

- The file opened with the ‘Open File’ icon (which may or may not be in the list) will always have ID 1.
- Therefore, the first file in the list will have ID 2, the second file in the list will have ID 3, and so on.
- Entering the ID number of an audio file in the ID number box will trigger its playback.
In order to add an audio file to the list:

1. Click on a file selector will appear.
2. Navigate to the desired folder and type the name of the file.
3. Click on ‘Open’: the file will be added at the end of the list.

Lists can be loaded, saved and edited: click on the corresponding buttons in order to carry out any of these operations.

Lists and presets
Lists are a useful to save the playback of specific audio files in the presets. This is because the ID number is saved as part of the preset and therefore recalling the latter will trigger playback.

Routing matrix

Synthesis and processing modules are interconnectable by means of a matrix emulating a patchbay, which also enables the connection of the outputs of any of the synthesisers, file players and processes to ten independent spatialisers (‘Panning and Output’). The columns represent the outputs of the processes (e.g. the first column is the output of synthesiser 1, column 12 is the output of spectral stretch 1, etc.) and the rows represent the inputs of the processes to which these outputs are routed (e.g. row 6 is the input of time-stretch 2, row 12 is the input of spatialiser 1, etc.).

It is worth bearing in mind that the granulator has its own spatialisers that send the signals directly to the MAX output; therefore, the granulator’s outputs do not require routing to the patchbay.
spatialisers. The audio file players can also be sent directly to MAX’s audio outputs and only the first two channels can be routed to the spatialisers: in other words, the patchbay is useful to spatialise mono and stereo files. On the other hand, multichannel files that are already distributed in space should be sent to the output directly using the audio file players.

In order to patch an output click on the intersection of the corresponding column (output to be routed) and row (input of routing destination): For instance, in the figure below, synthesiser 1 is sent to the granulator, and synthesiser 2 is sent to formants 1, 2, 3 and 4, which are in turn sent to spatialisers 3, 4, 5 and 6. In order to cancel an existing connection, click on it again.

Clicking on ‘Clear’ deletes all the connections.

Routing matrix Top Contents

Panning parameters

Each of the spatialisers can be set individually to produce new positions within a determined time interval and spatial range. Also, Doppler shift can be enabled or disabled, and the velocity of the sound source used in Doppler calculations can be increased/reduced by varying the distance between the speakers.

Radius - Measures the distance of the assumed speaker circle from the centre, which is significant when Doppler shift is enabled: for a given time interval, the larger the radius the faster the sound travels between speakers within that specific interval. Therefore, increasing the radius makes the Doppler shift more pronounced.

Period – The spatialisers generate new positions at regular time intervals given by the period.

Doppler – ON/OFF toggle.

Max Pan X and Z – These determine the spatial range within which positions are generated. The space is assumed to be a square with dimensions X (left-right) and Z (front-rear), where 0 is the centre, -1 is full left (rear) and 1 is full right (front). Therefore, a maximum X (Z) pan of 1 means that the positions will be generated randomly between full left (rear) and full right (front), a maximum X (Z) pan of 0.5 means that the positions will be generated randomly between half way between the centre and full left (rear), and half way between the centre and full right (front), etc. The spatialiser can also generate positions beyond the speakers if given values larger than 1.

Clicking on resets the corresponding Max Pan value to 1.

Routing matrix Top Contents
Spectral shift [3]
Spectral shift in this implementation multiplies frequencies above certain threshold - or split frequency - by a factor [4]. There are two independent units that can operate on different inputs assigned in the routing matrix. The process is carried out by recording on the fly a spectral analysis of the input, applying the current shift value and resynthesising the spectrum as audio samples. The spectral analysis is stored in frames: the more frames that are stored, the longer the audio sample.

All the parameters can change in real time and the user can also stop recording on the fly, in which case the time-varying shift will be applied to a recurring loop.

- ON/OFF toggle: when the shifter is not in use, turning it off will reduce the CPU load.

- Record ON/OFF toggle.

Frames – the number of frames to record.

dB Gain – Gain in decibels.

Split (Hz) – Split frequency.

Factor – Factor multiplying the shifted frequencies.

From, To – When using the P5 glove (or any controller) to control shift, these determine the range within which the shift factor can vary.

The dark windows at the bottom display corresponding spectrograms of the outputs.
Linking shift processors
In its default state, the shift will process independently the inputs assigned using the routing matrix: each shifter with its own individual parameters. However, there are situations when it is desirable to process two channels identically; for instance, when treating a stereo audio file or a stereo microphone. In such cases, the shifters may be linked so that any change in the parameters of shift 1 is automatically applied to shift 2.

In order to link the channels, click on [Unlinked], which will toggle to [Linked]. Also, an arrow pointing to the direction from 1 to 2 should appear: [1 → 2]. The parameters of shift 2 will remain the same until changes are effected in shift 1. Also, in this state, it is not possible to edit the parameters of shift 2.

QLists
The shift parameters can be automated using QLists [5] by means of text files containing time-varying parameters. These files can be saved, loaded and edited: click on the corresponding buttons in order to carry out any of these operations. Different QLists have different ID numbers, which are displayed in the number box. QList execution can be turned ON and OFF.

Clicking on ‘Help’ brings up a tutorial window with an explanation and samples on how to create QList text files for this module.

NOTE
QLists will only work when MAX audio is ON. This is because smooth transitions between breakpoints are implemented using MSP objects.
Spectral stretch [3]
Spectral stretch in this implementation multiplies frequencies above certain threshold - or split frequency - by a factor which increases with frequency (in contrast to a constant factor in the case of spectral shift) [4]. There are two independent units that can operate on different inputs assigned in the routing matrix. The process is carried out by recording on the fly a spectral analysis of the input, applying the current stretch values and resynthesising the spectrum as audio samples. The spectral analysis is stored in frames: the more frames that are stored, the longer the audio sample.

All the parameters can change in real time and the user can also stop recording on the fly, in which case the time-varying shift will be applied to a recurring loop.

- ON/OFF toggle: when the stretcher is not in use, turning it off will reduce the CPU load.
- Record ON/OFF toggle.

Frames – the number of frames to record.

dB Gain – Gain in decibels.

Split (Hz) – Split frequency.

Factor – Factor multiplying the stretched frequencies.

From, To – When using the PS glove (or any controller) to control stretch, these determine the range within which the stretch factor can vary.

The dark windows at the bottom display corresponding spectrograms of the outputs.
Linking stretch processors
In its default state, the stretch will process independently the inputs assigned using the routing matrix: each stretcher with its own individual parameters. However, there are situations when it is desirable to process two channels identically; for instance, when treating a stereo audio file or a stereo microphone. In such cases, the stretchers may be linked so that any change in the parameters of stretch 1 is automatically applied to stretch 2.

In order to link the channels, click on [Unlinked], which will toggle to [Linked]. Also, an arrow pointing to the direction from 1 to 2 should appear: [1] + [2]. The parameters of stretch 2 will remain the same until changes are effected in stretch 1. Also, in this state, it is not possible to edit the parameters of stretch 2.

QLists
The stretch parameters can be automated using QLists [5] by means of text files containing time-varying parameters. These files can be saved, loaded and edited: click on the corresponding buttons in order to carry out any of these operations. Different QLists have different ID numbers, which are displayed in the number box. QList execution can be turned ON and OFF.

Clicking on ‘Help’ brings up a tutorial window with an explanation and samples on how to create QList text files for this module.

NOTE
QLists will only work when MAX audio is ON. This is because smooth transitions between breakpoints are implemented using MSP objects.

Time-stretch and spectral blur [3]
Time-stretch changes the duration of a recorded sound without affecting its pitch [4]. There are two independent units that can operate on different inputs assigned in the routing matrix. The process is carried out by recording on the fly a spectral analysis of the input, applying the current time-stretch and blur values and resynthesising the spectrum as audio samples. The spectral analysis is stored in frames: the more frames that are stored, the longer the audio sample. In fact, because of the nature of a time-stretch operation a minimum number of frames is required (e.g. 25 frames or more) – this process will not be successful if very few frames are used. Also, because of this, there will be a noticeable delay between the recorded sound and the time-stretched output.

All the parameters can change in real time and the user can also stop recording on the fly, in which case the time-varying shift will be applied to a recurring loop.

Spectral Blur
Spectral blur in this case randomly combines the components of the current spectral frame with those of succeeding or preceding frames [6].
- ON/OFF toggle: when the time-stretch/blur unit is not in use, turning it off will reduce the CPU load.

- Record ON/OFF toggle.

**Frames** – the number of frames to record.

**dB Gain** – Gain in decibels.

**Speed** – How fast the audio is played relatively to its original speed, as follows:

- 1 plays the audio at its original speed.
- Values smaller than 1 result in slower speeds.
- Values larger than 1 result in faster speeds.
- Negative values reverse the audio, according to the same principles (-1: original speed; >-1: slower; <-1: faster).

**Blur Size** – The maximum number of succeeding or preceding frames with which to combine the current frame. Positive values indicate succeeding frames; negative values indicate preceding frames.

**Blur Mode** – This determines what happens when approaching the end of the recording sample:
• In **cycle mode** the processor warps around the end of the sample. For instance, if the last frame is reached and blur combines with up to three frames ahead, the processor will use frames 1, 2 and 3.

• In **clip mode**, the processor only uses the frames available up to the end of the sample. For instance, if the penultimate frame is reached and blur combines with up to three frames ahead, the processor will only use the last frame; when the last frame is reached there will be no blur because there will be no frames left to combine with.

The dark windows at the bottom display corresponding spectrograms of the outputs.

**Linking time-stretch/blur processors**

In its default state, the time-stretch/blur will process independently the inputs assigned using the routing matrix: each time-stretch/blur unit with its own individual parameters. However, there are situations when it is desirable to process two channels identically; for instance, when treating a stereo audio file or a stereo microphone. In such cases, the time-stretch/blur units may be linked so that any change in the parameters of time-stretch/blur 1 is automatically applied to time-stretch/blur 2.

In order to link the channels, click on ![Unlinked](image) which will toggle to ![Linked](image). Also, an arrow pointing to the direction from 1 to 2 should appear: ![Arrow from 1 to 2](image). The parameters of time-stretch/blur 2 will remain the same until changes are effected in shit 1. Also, in this state, it is not possible to edit the parameters of time-stretch/blur 2.

**QLists**

The time-stretch/blur parameters can be automated using QLists [5] by means of text files containing time-varying parameters. These files can be saved, loaded and edited: click on the corresponding buttons in order to carry out any of these operations. Different QLists have different ID numbers, which are displayed in the number box. QList execution can be turned ON and OFF.

Clicking on ‘Help’ brings up a tutorial window with an explanation and samples on how to create QList text files for this module.

**NOTE**

QLists will only work when MAX audio is ON. This is because smooth transitions between breakpoints are implemented using MSP objects.
Granulator

The granulator implements real-time asynchronous granular synthesis [7], recording audio into a buffer and reading small grains out of the latter, while allowing control of a large number of features including source read position, wander and speed (time-stretch); grain density, duration, transposition and spatial scatter; and cloud envelope.

Global parameters

- **ON/OFF toggle**: when the granulator is not in use, turning it off will reduce the CPU load.

- **Record ON/OFF toggle**.

- **Audio ON/OFF toggle**: the granulator has its own independent spatialisation and output mechanism. Furthermore, spatialisation can be controlled by the P5 glove (see glove control below). Therefore, although the grains can be patched to spatialisation and output via the routing matrix, this is not necessary.

- **Play/Stop toggle**: if the looping mode is OFF, playback will stop automatically when the source duration is reached.

- **Looping mode toggle**: when the looping mode is ON, the play/stop toggle is hidden.

**Voices** – The granulator is implemented using a MAX poly~ object [8] which attempts to distribute use among all the threads available in the CPU. The number of voices is the number of instances implemented in the poly object. More voices reduce the chances of having to drop out or curtail grains, but at the same time they increase CPU usage.

**Reset** – resets granulation parameters to default values designed to produce no output.

- **View mode: ENTER Params** - This toggle concerns parameters controlled by the P5 glove (or equivalent), switching between the following two modes:
  
  - **ENTER Params** – allows the user to enter parameters in the various number boxes.
  
  - **TRACK Params** – tracks the instantaneous values produced by the controller according to the mappings implemented by the user (see glove control below).
QLists

Granulation parameters can be automated using QLists by means of text files containing time-varying parameters. These files can be saved, loaded and edited: click on the corresponding buttons in order to carry out any of these operations. Different QLists have different ID numbers, which are displayed in the number box. QList execution can be turned ON and OFF.

Clicking on ‘Help’ brings up a tutorial window with an explanation and samples on how to create QList text files for this module.

NOTE
QLists will only work when MAX audio is ON. This is because smooth transitions between breakpoints are implemented using MSP objects.

Source

Dur (sec) – duration of the source sample (maximum 30 seconds).

Read Pos (msec) – position in the source sample from which the playback begins. If looping mode is on, the loop begins at this position.

Speed -

Wander – If this value is 0, playback advances the read position sequentially through the source. As wander increases the read position deviates from this sequence, leaping randomly forward or backwards: the maximum possible deviation is given by the value of the wander parameter.

Speed – How fast the audio is played relatively to its original speed, as follows:

- 1 plays the audio at its original speed.
- Values smaller than 1 result in slower speeds.
- Values larger than 1 result in faster speeds.
• Negative values reverse the audio, according to the same principles (-1: original speed; >-1: slower; <-1: faster).

**Cloud attributes**

- Graphic function editor [9]: allows graphic editing of the envelope using the mouse.
- Clicking on any of the icons resets the envelope to the displayed shape.

To type manually the values of a breakpoint:

• Click on ‘Keyboard’: the following window will appear:

  - Type in the timing of the breakpoint below ‘X msec’ and a value between 0 and 1 below ‘Y [0-1]’.

  - Click on ‘Create Point’ or type ‘e’ on the keyboard: the window will close and the breakpoint will appear in the graphic function editor.

**Env Dur (sec)** – envelope duration in seconds.
Env Mode – Envelopes and source recording can be applied on the flight within presets to prompt one of the following actions:

Constant – no envelope is applied.

Envelope – In envelope mode, clicking on one of the envelope icons will trigger the envelope as well as resetting it.

REC – obsolete.

Trigger Envelope – selecting this mode triggers the current envelope.

Trigger REC – selecting this mode triggers recording.

REC triggers Envelope and Trigger REC + Envelope - selecting any of these modes triggers recording together with the current envelope.

- Triggers the envelope displayed breakpoint function editor. Changes the Env Mode to ‘Envelope’.

Overall Gain(dB) – overall cloud gain in decibels.

The following pairs of number boxes set an upper and a lower boundary within which parameters are generated randomly:

Density (grains/sec) – Number of grains generated in one seconds.

Grain Dur (msec) – Grain duration in milliseconds.

Transp (semitones) – Transposition in Semitones.

- Clicking on this resets the corresponding values to 0.

- Determines the type of link between the boundaries as follows:

  - X - The boxes are unlinked and therefore can be modified separately.

  - -> - When the ‘From’ number box is modified, its value is copied to the ‘To’ number box. The latter can be modified without affecting the ‘From’ number box.

  - <-> - When the ‘To’ number box is modified, its value is copied to the ‘From’ number box. The latter can be modified without affecting the ‘To’ number box.
Grain envelope

The user can select from a variety of envelopes from the menu, according to the following choices:

- Sinusoidal bell types (depicted in the figure above), including Hanning, Hamming, sine, Blackman and Blackman-Harris.
- Trapezoid (attack-sustain-decay): in this case, the attack and decay can be set using number boxes.
- Chant: this envelope is similar to that used in order to create formants, consisting of a sinusoidal attack (sigmoid), and exponentially decaying steady state and a linear decay. The steepness of the steady state is controlled by the bandwidth parameter: the larger the bandwidth, the steeper the decay. Attack, decay and bandwidth can be set using number boxes.
- Graphic Edit: this option opens a graphic editor [10], allowing the user to design a custom made envelope.

**Attack (msec)** – Sets the attack for a trapezoid and chant envelopes.

**Decay (msec)** – Sets the decay for a trapezoid and chant envelopes.

**Chant BW (Hz)** – Sets the bandwidth for a chant envelope.
Panning
The granulator has its own independent spatialisation and output mechanism, producing new positions within a determined time interval and spatial range, similarly to spatialisers in the routing matrix. Likewise, Doppler shift can be enabled or disabled, and the velocity of the sound source used in Doppler calculations can be increased/reduced by varying the distance between the speakers. In addition, it also allows further scattering of the grains relatively to the cloud.

Spkr Radius (m) - Measures the distance of the assumed speaker circle from the centre, which is significant when Doppler shift is enabled: for a given time interval, the larger the radius the faster the sound travels between speakers within that specific interval. Therefore, increasing the radius makes the Doppler shift more pronounced.

Period (msec) – The spatialisers generate new positions at regular time intervals given by the period.

Doppler – ON/OFF toggle.

Max Pan X and Z – These determine the spatial range within which positions are generated. The space is assumed to be a square with dimensions X (lef-right) and Z (front-rear), where 0 is the centre, -1 is full left (rear) and 1 is full right (front). Therefore, a maximum X (Z) pan of 1 means that the positions will be generated randomly between full left (rear) and full right (front), a maximum X (Z) pan of 0.5 means that the positions will be generated randomly between half way between the centre and full left (rear), and half way between the centre and full right (front), etc. The spatialiser can also generate positions beyond the speakers if given values larger than 1.

Scatter – Factor that multiplies the randomly calculated positions, scattering the grains from the cloud: larger values produce more scattered grains and smaller values gather the grains closer to the cloud.

Clicking on resets the corresponding value to 1.

Formant bank [3]
A bank of formants consists of a number of overlapped filters which define a spectral envelope, as shown in the figure below. The centre/cut-off frequencies and bandwidths of these filters can change in time [11].
In this implementation, the formants consist of FFT (Fast Fourier Transform) filters for which an archetype frequency response can be drawn by the user using the mouse on a special display. The formants drawn can be shifted in frequency using the sliders provided underneath each display. The process is carried out by recording on the fly a spectral analysis of the input, applying the current filters and resynthesising the spectrum as audio samples. The spectral analysis is stored in frames: the more frames that are stored, the longer the audio sample.

Global parameters

- ON/OFF toggle: when the granulator is not in use, turning it off will reduce the CPU load.

- Record ON/OFF toggle.

Frames – the number of frames to record.
- Deletes all the formants, resulting in a flat zero response that stops all frequencies.

- Toggle for controlling the frequency shift of the formants, consisting of two states:
  
  - **4 Sliders** – each formant shift is controlled independently by its own slider.
  
  - **1 Slider** – the first slider (green) controls the shift of all formants. However, not all formants will necessarily move uniformly in the same direction, since it is possible to introduce a random jitter percentage (see the jitter parameter below). Therefore, only if the jitter is 0% will the formants move uniformly.

The dark window at the bottom displays a spectrogram of the output.

- This toggle switches between a logarithmic and a linear display of the output’s spectrogram.

**QLists**

All the parameters can be automated using QLists by means of text files containing time-varying parameters. These files can be saved, loaded and edited: click on the corresponding buttons in order to carry out any of these operations. Different QLists have different ID numbers, which are displayed in the number box. QLlist execution can be turned ON and OFF.

Clicking on ‘Help’ brings up a tutorial window with an explanation and samples on how to create QLlist text files for this module.

**NOTE**

QLists will only work when MAX audio is ON. This is because smooth transitions between breakpoints are implemented using MSP objects.
Individual formant parameters

- indicates the normalised amplitude (0 to 1) and frequency of the first absolute maximum in the formant. It is calculated automatically when the user draws a formant.

- Lowest frequency of the formant.

- Highest frequency of the formant.

- Deletes the formant, resulting in a flat zero response that stops all frequencies.

- Algorithm used to calculate the width of the formant shift. This affects the size of the frequency shift of a formant. It can be one of the following:
  - Full bandwidth: calculation based on the difference between the lowest and highest frequency of the formant.
  - Maximum: calculation based on first absolute maximum in the formant.
  - Geometric average: calculation based on the geometric average of the lowest and highest frequency of the formant.
  - Arithmetic average: calculation based on the arithmetic average of the lowest and highest frequency of the formant.

- Percentage of the calculated width of the formant shift covered by the slider (see below). The larger the percentage, the larger the bandwidth covered by the slider.

- The slider allows the user to shift the frequencies of the formant: the frequency corresponding to the slider’s current position is displayed in the number box. However, this position will not always yield the same frequency; since it is possible to introduce a random jitter percentage (see the jitter parameter below). Therefore, only if the jitter is 0% will the position will always yield the same frequency shift. Clicking on brings the slider to its centre position.
- determines a maximum random deviation for the frequency shift of a formant, as a percentage of the calculated bandwidth. A new random deviation is calculated every time a formant is shifted or any of the formant parameters is changed.

- Formant gain in decibels.

**Formant bank**      **Top**      **Contents**

### Linking formants

In order to maintain the spatialisation of processed stereo files, each channel has to be connected to a different formant input. However, this presents the inconvenience of having to maintain identical parameters for both channels, which might become laborious and inaccurate; especially when drawing a formant. To resolve this issue, it is possible to link formants so that input into one of the filters is replicated automatically in the other, and parameters can be copied with one click. This is done by establishing a master-slave link using the matrix appearing to the left of the displays.

The rows of the matrix represent the master formants and the columns represent the slaves. When a master is linked to a slave, an encircled downward arrow appears on the latter’s graphic display and its parameters cannot be edited: when the user makes any changes to the master, the latter’s frequency response and parameters are copied to the slave. For example, in the following figure formant 1 is the master and formant 2 is the slave (note the matrix connection and encircled arrow in formant 2).

By contrast, the figure below shows formant 2 as master and formant 1 as slave (note the matrix connection and where the encircled arrow appears).
Another way of copying the master to the slave is clicking on the encircled arrow button appearing in the last row of the master parameters: for example, in the previous figure this would be in formant 2.

Finally, note that a master can control more than one slave.

**Glove control (Windows only)**

Although MAES can function without an external digital controller, it is designed to work with the *P5 Glove* [1], offering facilities to calibrate, track, load and save settings, and, most important, map P5 data to the implemented processes in order to create musical gestures. When using the P5, it is strongly recommended to fit the glove with adjustable Velcro attachments placed between the base of the finger and the original plastic rings supplied with the glove in order to couple the rubber bands to the fingers. This is because the plastic rings on their own are prone to slippages that affect the reliability and repeatability of finger bend measurements.

- **ON/OFF toggle for glove tracking.**

**NOTE**

If the P5 is out of range and tracking is ON, MAX may not respond. This is easily remedied by bringing the P5 into range and toggling the tracking to OFF.

**Sampling period (msec)** – This is the sampling period for tracking glove data.

**Scaling**

**Velocity factor** – The P5 MAX external tracks absolute velocity, but this might be too slow to make a significant impact on some parameters. Therefore, it is multiplied by a factor. Experience suggests that 30 is a good default value.

**Normalise Position [-1, 1]** – It is possible to normalise position tracking to the interval [-1, 1] (recommended). The switch toggles calibration ON and OFF.
Data Smoothing
This provides an option to apply a low pass filter in order to smooth the tracked data. Filters can be switched ON/OFF for position, orientation and velocity the cut off frequency can be specified individually for each filter. Finding the appropriate cut off for a filter involves a balance between obtaining smoother tracking (by lowering the cut off value) and avoiding sluggish reactions (by increasing the cut off value).

Experience suggests that, in most situations, it is only necessary to filter orientation data. The default values are also based on experience using the P5.

Calibration
It is strongly recommended that the glove is calibrated before every performance: although calibration data can be saved for each user, there is too much variability whenever the glove is re-fitted to the same hand.

Calibration process:

1. Click on (or type the shortcut key ‘c’): this invokes the calibration window:

   ![Calibration Window](image)

   - **Glove Buttons**:
     - A - Restart calibration
     - B - Next value / Exit and save
     - C - Exit without saving

   - **Diagram**:
     - **TOP**
     - **REAR** (Towards lower)
     - **LEFT**
     - **RIGHT**
     - **FRONT**
     - **BOTTOM**

1. Move the hand to the leftmost position and click on the glove button labelled B. The right calibration position should appear:
2. Repeat this procedure for the remaining parameters: right, top, bottom, front, rear, orientation X (rotate the wrist on the left-right axis), orientation Y (rotate the wrist on the vertical top-bottom axis), orientation Z (rotate the wrist on the front-rear axis), maximum finger bend (clenched fist) and minimum finger bend (open hand). After the last press of button B following minimum finger bend the calibration will be complete, its values will be saved in memory for the rest of the MAX session and the window will disappear automatically.

3. Pressing glove button A restarts the calibration process.

4. Pressing glove button C exits without calibration.
NOTE
Do not close the calibration window by clicking the close window icon since this may interfere with proper opening and closing of this window in subsequent calibrations.

Hand shapes
MAES implements hand-shape recognition, by measuring individual finger bend within specified deviation tolerances. Tolerances are necessary in the case of the P5 because of the slippages resulting from coupling the bend measuring rubber bands to the fingers (even when Velcro rings are used, as explained above).

It is possible to store up to 64 shapes: each shape is assigned a unique index. When a shape is recognised, this will be indicated in an arrangement emulating a hand. Each of the fingers is a slider indicating the bend of each finger (black is fully bent) and, when a shape is recognised, the palm shows its index number. For example, the figure below indicates that shape 1 was recognised, consisting of a fully bent thumb, a fully stretched middle finger, and the index, ring and little fingers stretched to a lesser extent.

![Image of hand shapes]

Deviation – This is the deviation tolerance: maximum bend is normalised to values between 0 and 1. For example, if the thumb finger bend to be recognised is 0 and the tolerance is 10%, values of up to 0.1 would still be considered to be equivalent to 0 finger bend so that the thumb would conform to the shape.

- Deletes all stored hand shapes.
- Posts the number of shapes to the MAX window.

To store a shape:

5. Click on (or type the shortcut key ‘h’): this invokes the hand shapes window.
6. Form a shape with the hand and click on the glove button labelled B: the shape will be stored and the index will be incremented.

7. Repeat this procedure to store further shapes.

8. Pressing glove button A advances the index without storing a shape.

9. When the shapes have been stored, press the glove button C to exit.

Further shapes can be stored by repeating steps 1 to 5. This will not delete shapes already stored. However, the shapes will not remain in memory after quitting MAX. Therefore, in order to keep these shapes, these have to be saved together with all the glove settings to a file.

**Save and Load**
Glove settings can be saved to a file: this includes calibration values, shapes and the deviation tolerance.

- Saves glove settings.

- Loads glove settings.

**Tracking**
(shortcut key ‘t’) – This option opens a window that tracks the glove parameters in real time. All parameters are displayed in the number boxes at the bottom. Also, graphic tracking of position, orientation and velocity can be activated using corresponding ON/OFF switches.
Tracking is mainly useful for testing purposes; checking that the glove is functioning properly and well calibrated. The graphic displays can be used to check appropriate smoothness of the data with and without filters. Therefore, once these checks are carried out, it is recommended to turn off graphic tracking and close the window in order to reduce the CPU load.

Glove control

Gesture creation and mapping

MAES provides configurable mappings via a matrix. It is possible to establish mappings between 15 streams of glove tracked data and 18 continuous processing parameters, a physical model for throwing/sowing particles, two soundfile triggers, and a preset increment (figure 3). Mappings can be combined simultaneously to generate more complex metaphors. They can also include activating conditions so that the glove’s tracked data effects the corresponding parameter only if such conditions are met.

Mappings are created by means of the following matrix, which appears when is clicked.
Its rows represent glove track data (e.g. position X, Y, and Z) and the columns represent parameters (e.g. granular cloud density, duration and transposition boundaries), throw/sow, soundfile triggers and preset increments. Note that in addition to individual finger bend, it is possible to map average finger bend for the whole hand. It is also possible to map identified hand shapes. Some of the intersections are disabled (appearing blurred) either because the connections do not make sense (e.g. mapping hand shape detection to continuous parameters) or because the mappings are not implemented in this particular version of MAES.

Conditions are set using a purposely designed window, which appears when the user clicks on the caption of a destination parameter. For example, clicking on

will bring up the following window:
Continuous targets can have more than one condition: for instance, if the figure above represented a mapping between Position X and Density 1, this mapping would also be conditional on the value of position Y being greater than 0.5.

To create gestures using mappings:

1. Click on : the mapping matrix will appear.

2. Click on the intersections of the corresponding rows (tracked data) and columns (mapping destination). For example, in the figure above, Position X is mapped to Density 1, Position Y is mapped to both Transposition 2 and Scatter, Velocity Y to Density 2 and finger bend average to PRESETS INC (presets increment). In order to cancel an existing connection, click on it again.

3. Establish the mapping conditions by on the caption of the destination parameter: the conditions window will appear.

4. Close the conditions window.

5. Repeat steps 3 and 4 for all the mappings.

6. Close the mapping matrix.
Mapping conditions
Conditions vary depending on the data being mapped and the type of target they are mapped to. As a result, there are four types of mapping window.

1. Continuous targets that can only assume positive values: for instance, cloud density, spectral shift factor, etc.

- Direct/inverse mapping toggle: mapping is direct when increments (decrements) in the source parameter cause corresponding increments (decrements) in the target parameter. It is inverse when increments (decrements) cause corresponding decrements (increments).

- Mapping condition. The following options are available depending on the type of tracked data and target parameter:

  **A absolute value** – Unconditional. The absolute value of tracked data is mapped to density. For example, if Position X is mapped to Density 1 and the density boundary is 50 grains /second, Position X = 0 will map to 0 grains/second, normalised Position X = 1 will map to 50 grains/second and normalised Position X = -1 will also map to 50 grains/second.

  **R convert to positive range [0, max]** – Unconditional. The normalised value of tracked data range will be rescaled from [-1, 1] to [0, 1]. For example, if Position X is mapped to Density 1, and the density boundary is 50 grains /second, Position X = 0
will map to 25 grains/second (because 0 is rescaled to 0.5), normalised Position X = 1 will map to 50 grains/second and normalised Position X = -1 will also map to 0 grains/second (because -1 is rescaled to 0).

>A activate above threshold - absolute value – Implements the same mapping as ‘A absolute value’. However, there is a condition: the tracked data value must be greater than a given threshold. A number box will appear so that the threshold can be entered. This is the case of Position Y in the figure above.

>R activate above threshold – range - Implements the same mapping as ‘R convert to positive range [0\, \text{max}]’. However, there is a condition: the tracked data value must be greater than a given threshold. A number box will appear so that the threshold can be entered.

<A activate below threshold - absolute value value – Implements the same mapping as ‘A absolute value’. However, there is a condition: the tracked data value must be smaller than a given threshold. A number box will appear so that the threshold can be entered.

<R activate below threshold – range - Implements the same mapping as ‘R convert to positive range [0\, \text{max}]’. However, there is a condition: the tracked data value must be smaller than a given threshold. A number box will appear so that the threshold can be entered.

Sa activate absolute value on hand shape - Implements the same mapping as ‘A absolute value’. However, there is a condition: the mapping will only be active if a hand shape is recognised. A number box will appear so that the hand shape index can be entered.

Sr activate range on hand shape - Implements the same mapping as ‘R convert to positive range [0\, \text{max}]’. However, there is a condition: the mapping will only be active if a hand shape is recognised. A number box will appear so that the hand shape index can be entered.

Sa create hand shape and activate absolute value – Implements the same mapping and condition as ‘Sa activate absolute value on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

Sr create hand shape and activate range – Implements the same mapping and condition as ‘Sr activate range on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

- Depending on the mapping option: number box for input of additional data (e.g. threshold, shape index).
2. Continuous targets that can assume positive and negative values: for instance, cloud transposition, formant shift, etc.

The only difference between this and the previous window is found in the mapping condition options:

- **+ signed values [-max, max]** – Unconditional. A straightforward mapping of positive tracking data to positive target values and negative tracking data to positive target values.

- **R+ convert to positive range [0, max]** – Unconditional. The normalised the tracked data range will be rescaled from [-1, 1] to [0, 1] and mapped only to positive values.

- **R- convert to negative range [-max, 0]** – Unconditional. The normalised the tracked data range will be rescaled from [-1, 1] to [-1, 0] and mapped only to negative values.

- **> activate above threshold** – signed values – Implements the same mapping as ‘+ signed values [-max, max]’. However, there is a condition: the tracked data value must be greater than a given threshold. A number box will appear so that the threshold can be entered.

- **>+ activate above threshold** – positive range – Implements the same mapping as ‘R+ convert to positive range [0, max]’. However, there is a condition: position must be greater than a given threshold. A number box will appear so that the threshold can be entered.

- **>- activate above threshold** – negative range – Implements the same mapping as ‘R- convert to negative range [-max, 0]’. However, there is a condition: the tracked data value must be greater than a given threshold. A number box will appear so that the threshold can be entered.

- **< activate below threshold** – signed values – Implements the same mapping as ‘+ signed values [-max, max]’. However, there is a condition: the tracked data value must be smaller than a given threshold. A number box will appear so that the threshold can be entered.

- **<+ activate below threshold** – positive range – Implements the same mapping as ‘R+ convert to positive range [0, max]’. However, there is a condition: position must be smaller than a given threshold. A number box will appear so that the threshold can be entered.

- **<- activate below threshold** – negative range – Implements the same mapping as ‘R- convert to negative range [-max, 0]’. However, there is a condition: the tracked data value must be smaller than a given threshold. A number box will appear so that the threshold can be entered.

- **S activate signed value on hand shape** – Implements the same mapping as ‘+ signed values [-max, max]’. However, there is a condition: the mapping will only be
active if a hand shape is recognised. A number box will appear so that the hand shape index can be entered.

**S+ activate positive range on hand shape** – Implements the same mapping as ‘R+ convert to positive range [0ν, max]’. However, there is a condition: the mapping will only be active if a hand shape is recognised. A number box will appear so that the hand shape index can be entered.

**S- activate negative range on hand shape** – Implements the same mapping as ‘R- convert to negative range [-maxν, 0]’. However, there is a condition: the mapping will only be active if a hand shape is recognised. A number box will appear so that the hand shape index can be entered.

**S create hand shape and activate signed value** – Implements the same mapping and condition as ‘S activate signed value on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S+ create hand shape and activate positive range** – Implements the same mapping and condition as ‘S+ activate positive range on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S- create hand shape and activate negative range** - Implements the same mapping and condition as ‘S- activate negative range on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

3. **Throw/sow**, which can be used to emulate throwing or sowing particles. Sowing consists of moving the arm in an arched trajectory while opening the hand to release the particles, which spread as they travel through the air; as opposed to throwing, in which particles are released at once without following the arm’s trajectory.
- Menu choice between throw and sow modes.

- Mapping condition. The following options are available depending on the type of tracked data and throw/sow activation:

> T activate above threshold – Activates when the tracking data value is greater than a given threshold. A number box will appear so that the threshold can be entered.

< T activate below threshold – Activates when the tracking data value is smaller than a given threshold. A number box will appear so that the threshold can be entered.

<> activate between two thresholds – Activates when the tracking data value is between two thresholds. Two number boxes will appear so that the thresholds can be entered.
**S<> activate between two thresholds on hand shape** – Activates when the tracking data value is between two thresholds AND a hand shape is recognised. Two number boxes will appear for the thresholds and another box will appear for the hand shape index.

**S>< activate beyond two thresholds on hand shape** – Activates when the tracking data value is outside two thresholds AND a hand shape is recognised. Two number boxes will appear for the thresholds and another box will appear for the hand shape index.

**S>T create hand shape and activate above threshold on hand shape** – Implements the same mapping and condition as ‘S>T activate above threshold on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S<T create hand shape and activate below threshold on hand shape** – Implements the same mapping and condition as ‘S<T activate below threshold on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S<> create hand shape and activate between two thresholds on hand shape** – Implements the same mapping and condition as ‘S<> activate between two thresholds on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S>< create hand shape and activate beyond two thresholds on hand shape** – Implements the same mapping and condition as ‘S>< activate beyond two thresholds on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

- Activates a hand shape is recognised. A number box will appear for the thresholds and another box will appear so that the hand shape index can be entered.
In addition there are a number of parameters used to shape the physics of the throw and sow algorithms in two dimensions: X (left-right direction) and Z (rear-front direction).

**Time Exponent** – This is used to warp the time so that it does not increment linearly as follows:

1: time progresses linearly.

1: time progresses increasingly faster.

-1: time progresses increasingly slower.

**Velocity factor** – Multiplies the tracked velocity by a factor to amplify the momentum transferred to the particles.

**Gravity** – Gravity acceleration pulling the particles when they are released. It can be positive (acceleration) or negative (deceleration).

**Sampling period** – Sampling period for the position of the particles.

**Release (sec)** (sow mode only) – The time it takes to release all the particles.

---

4. **Trigger soundfile/preset increment**

This is mapping type is used to trigger soundfiles and increment presets.
- Mapping condition. The following options are available depending on the type of tracked data and throw/sow activation:

> T activate above threshold – Activates trigger/increment when the tracking data value is greater than a given threshold. A number box will appear so that the threshold can be entered.

< T activate below threshold – Activates trigger/increment when the tracking data value is smaller than a given threshold. A number box will appear so that the threshold can be entered.

<> activate between two thresholds – Activates trigger/increment when the tracking data value is between two thresholds. Two number boxes will appear so that the thresholds can be entered.

>< activate beyond two thresholds (below lower/above higher) – Activates trigger/increment when the tracking data value is outside two thresholds (greater than the higher threshold and smaller than the lower threshold). Two number boxes will appear so that the thresholds can be entered.

S>T activate above threshold on hand shape – Activates trigger/increment when the tracking data value is greater than a given threshold AND a hand shape is recognised. A number box will appear for the threshold and another box for the hand shape index.

S<T activate below threshold on hand shape – Activates trigger/increment when the tracking data value is smaller than a given threshold AND a hand shape is recognised. A number box will appear for the threshold and another box for the hand shape index.

S<> activate between two thresholds on hand shape – Activates trigger/increment when the tracking data value is between two thresholds AND a hand shape is recognised. Two number boxes will appear for the thresholds and another box will appear for the hand shape index.

S>< activate beyond two thresholds on hand shape – Activates trigger/increment when the tracking data value is outside two thresholds AND a hand shape is recognised. Two number boxes will appear for the thresholds and another box will appear for the hand shape index.

S>T create hand shape and activate above threshold on hand shape – Implements the same mapping and condition as ‘S>T activate above threshold on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

S<T create hand shape and activate below threshold on hand shape – Implements the same mapping and condition as ‘S<T activate below threshold on hand shape’
and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S<> create hand shape and activate between two thresholds on hand shape** – Implements the same mapping and condition as ‘S<> activate between two thresholds on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

**S>< create hand shape and activate beyond two thresholds on hand shape** - Implements the same mapping and condition as ‘S>< activate beyond two thresholds on hand shape’ and invokes the hand shapes window so that the user can store a new hand shape for use with this mapping.

- Activates trigger/increment a hand shape is recognised. A number box will appear for the thresholds and another box will appear so that the hand shape index can be entered.

- Form/Release toggle.

  - Form - trigger when shape is formed.
  - Release - trigger when shape is released.

**Presets**
All the relevant parameters in MAES and its subpatches can be saved in presets, including glove mappings. Furthermore, because presets are implemented using pattrstorage and autopattr, data may be recalled as an interpolated value between two stored values [12]. The large white number box contains the number of the current preset. Entering a number in this box invokes the corresponding preset. Presets can also be incremented and decremented sequentially in the following ways:

(shortcut key ‘k’) - increment presets.

(shortcut key ‘j’) - decrement presets.

Toggle to enable/disable use of the spacebar to increment presets.

Presets ➩   Top ➩   Contents ➩

Storing presets
Presets can be stored in the following ways:

Entering a number in this box will store the data in a preset with the same number.

(shortcut key ‘s’) - This is a quick way to store presets sequentially: when clicking (or typing ‘s’) the number in the ‘Store=>’ box will first be incremented and then the data will be stored in a preset with the resulting number. This can be used in combination with the increment preset keystrokes in order to copy quickly a block of presets into a higher location. For example, assume we wish to copy presets 3-7 to locations 23-27. This can be done relatively quickly in the following steps:

4. Type ‘3’ in the large white number box at the top to recall preset 3.
5. Type ‘23’ in the ‘Store=>’ box to store this preset in location 23.
6. Press ‘k’ (or the spacebar if enabled) to advance to the next preset.
7. Press ‘s’ to store this preset in the next location.
8. Repeat steps 3 and 4 three more times. This should copy the remaining presets.

- deletes all stored presets.

Presets ➩   Top ➩   Contents ➩
Saving/loading presets and other files
Stored presets can be loaded and saved: click on the corresponding buttons in order to carry out any of these operations. Presets are saved as XML format, with the extension “.json” [12].

Often, it is necessary to load additional files in order to be able to trigger audio files, execute qlists and recall PS glove settings (if applicable). For this reason, an option to load all these files from a single folder is provided. The only condition is that corresponding files must have standard names as follows:

- **P5settings.txt**: stores P5 glove data.
- **Presets.json**: presets.
- **Soundfile1Preload.txt**: audio file pre-load list for player 1.
- **Soundfile2Preload.txt**: audio file pre-load list for player 2.
- **QListShift.txt**: spectral shift QLists.
- **QListStretch.txt**: spectral stretch QLists.
- **QListTimeStretch.txt**: time-stretch QLists.
- **QListGranulator1**: granulator QLists.
- **QListFormants.txt**: formants QLists.

It is not necessary to create all these files in the folder if they are not needed (e.g. no granulator QLists used, no pre-loaded audio files used, etc.), but if they do not appear an error message will appear in the MAX window. Nevertheless, this will not affect the functionality of the patch.

In order to load from a single folder:

1. Click on the file selector: a file selector will appear.
2. Navigate to the desired folder and click on it.
3. Click on ‘OK’: the file will be added at the end of the list.

Viewing stored data
The stored data can be viewed and edited manually:

Clicking on this button opens the `pattrstorage` object’s stored data window. If there are any stored presets these will appear in successive columns. Each value can be edited by double clicking on it and typing [12].
**CPU and audio status**

Whenever ‘Poll’ is ticked and MAX audio is ON, this panel indicates CPU usage.

Clicking on invokes the standard MAX Audio Status window [13].
Troubleshooting

High CPU Usage
Depending on the CPU capabilities and the amount of processing carried out, CPU usage may rise to 100% and the patch will not work properly. In many cases, this can be remedied by reducing the signal vector size in the MAX Audio Status window [13]. In order to do this:

1. Click on to invoke the standard MAX Audio Status window.
2. In the Audio Status window increase the value of the Signal Vector Size (normally a power of 2): the larger the Signal Vector Size value the lower the CPU load. For instance, if the Signal Vector Size is 32, change this to 64.

Broken Audio/QList not Working Smoothly
If the CPU is not critically high but the audio sounds broken or the QLists do not interpolate smoothly between breakpoints, it is possible to use the MAX scheduler in ‘Overdrive’ and/or in ‘Audio Interrupt’ by ticking the corresponding option in the Audio Status window (circled in the figure above). This will come at the cost of a more sluggish interface and, more significantly, it might affect the responsiveness to the P5 controller. However, if the P5 is not used, this will improve overall performance.
Technical Appendix - Replacing the P5 module by an alternative controller

Furthermore, the MAX external that communicates with the P5 glove can be easily replaced by any other external or subpatch that provides all or part of the corresponding data. The following data is processed by MAES:

- Position X,Y,Z
- Orientation X,Y,Z
- Velocity X,Y,Z
- Individual finger bend.
- Recognised hand shape index.

In order to replace the P5 glove module use the MAX sources:

1. Open the project MAES.maxproj.
2. Click on the tracking window will appear.
3. Exit the presentation view (toggled by Ctrl+Alt+E).
4. Enter edit mode (toggled by Ctrl+E).
5. Enlarge the window and scroll until the P5 module P5GloveRF is found:
6. Replace this external for an alternative external or subpatch.

P5GloveRF inputs:

Input 1- multiple data:

- On/OFF switch to start/stop tracking data
- Normalise position toggle: normalise <1 or 0> (1: normalise, 0: do not normalise)
- Command to set the hand shape deviation tolerance: hstolerance <tolerance-fraction>
- Command to store hand shapes: hsstore <thumb-bend> <index-bend> <middle-bend> <ring-bend> <little_finger-bend>
- Command to delete all the stored shapes: hsclear
- Command to save settings: write
- Command to load settings: read

Input 2 - sampling period.

PSGloveRF outputs:

Output 1 – Position X
Output 2 – Position Y
Output 3 – Position Z
Output 4 – Orientation X
Output 5 – Orientation Y
Output 6 – Orientation Z
Output 7 – Velocity X
Output 8 – Velocity Y
Output 9 – Velocity Z
Output 10 – Acceleration X (not used in MAES)
Output 11 – Acceleration Y (not used in MAES)
Output 12 – Acceleration Z (not used in MAES)
Output 13 – Index of hand shape (if recognised)
Output 14 – Thumb bend
Output 15 – Index bend
Output 16 – Middle bend
Output 17 – Ring bend
Output 18 – Little finger bend
Output 19 – ‘A’ button state (1: ON, 0: OFF)
Output 20 – ‘B’ button state (1: ON, 0: OFF)
Output 21 – ‘C’ button state (1: ON, 0: OFF)
Output 22 – Dumping float information (unused in MAES)
Output 23 – Bang after settings are loaded or saved
Notes


[10] This is the graphic editor for the MAX table object: see MAX help pages (e.g. http://cycling74.com/docs/max5/refpages/max-ref/table.html, 21/8/13).

