

mubu.concat~

MuBu concatenative synthesis

Description

MuBu concatenative synthesis module based on the *ZsaZsa* overlap-add synthesis engine.

The external plays segments of a given audio buffer regarding a segmentation defined by markers in a given track of a MuBu container. The module processes a sound stored in a [buffer~](#) or [polybuffer~](#) of the same name as the referred MuBu container.

The external allows for over sample accurate scheduling of segments within the DSP calculation cycle (when `period` or `periodfactor` > 0).

After starting a segment, the external outputs its segment index and buffer index from the right-most outlet.

Arguments

<i>Name</i>	<i>Type</i>	<i>Opt</i>	<i>Description</i>
name	symbol		Name of the referred MuBu container and audio buffer.

Messages

bang			Plays current segment.
markerindex	index [int]		Sets segment by marker index.
markertime	time [float]		Sets segment by marker time
refer	mubu name [symbol]		Sets the referred MuBu container and audio buffer.
segment	segment index [int] buffer index [int]		Sets segment index and buffer index (if given) and plays segment.

Attributes

<i>Name</i>	<i>Type</i>	<i>g/s</i>	<i>Description</i>
advance	float		Sets the segment scheduling advance. The advance gives the possibility to advance the segment synthesis time by an <code>offset</code> in order to distinguish properly between the position of the segment in the source sound (defined by the marker time) and its reference time for temporally precise synthesis. The scheduling advance is only taken into account when the segment synthesis is scheduled internally (<code>period</code> or <code>periodfactor</code> > 0). Externally triggered segments (using the <code>bang</code> or <code>segment</code> message), are scheduled immediately ignoring the <code>advance</code> and <code>segment offset</code> .
attack	float		Sets the segment synthesis attack time in msec or (value < 0) as factor of the segment duration.
bufferindex	int		Sets current buffer by index.
duration	float		Sets segment duration in msec and disables marker duration.
durationcol	atom		Sets or resets (without argument) the column (by index or name) of the segment duration.
durationfactor	float		Sets segment duration as a factor of the marker duration.
level	float		Sets the segment level in dB.
levelvar	float		Sets the amount of random variation of the segment level in dB.
markerchange	int		Supresses the repetition of segments.

When enabled, each segment defined within the same marker track (defined by marker and buffer index) will be played only once.

markers	atom	Sets identifier (index or name) of the referred marker track.
maxduration	float	Sets the maximum segment synthesis duration (including the increment by a negative <code>offset</code>). Longer segments will be truncated by the synthesis engine.
mubuname	symbol	TEXT_HERE
offset	float	Sets the offset in msec. When the segment synthesis is automatically controlled by a <code>period</code> or <code>periodfactor</code> , the offset anticipates a segments synthesis time (or <i>synchronization point</i>). Negative offsets advance the source position of the segment defined by the marker (increasing the segment duration) and anticipate the their synthesis time by the given offset. Positive offset also anticipate the synthesis time of the segments by the given offset, but preserve the source position of the segment as defined by the markers. The offset is limited by the segment scheduling advance.
offsetcol	atom	Sets or resets (without argument) the column (by index or name) of the segment offset.
period	float	Sets the segment synthesis period in msec.
periodcol	atom	Sets or resets (without argument) the column (by index or name) of the segment period.
periodfactor	float	Sets the segment synthesis period as a fator of the marker period. The marker period is defined by the time interval between the current segment's marker and the next.
positionvar	float	Sets the amount of random variation of the segment source position in msec.
release	float	Sets the segment synthesis release time in msec or (value < 0) as factor of the segment duration.
resampling	float	Sets the amount of resampling in cent.
resamplingvar	float	Sets the amount of random variation of resampling in cent.
spatialmode	int	Sets the spatial segment distribution mode.

The following modes are provided:

- direct
- gains

direct

In `direct` mode, segment channels are output from a given audio output. For multi-channel sources, the segment channels are distributed to succesive audio outputs starting from the the given channel offset.

gains

In `gains` mode, a single segment channel (by default the first channel of a multi-channel segement) is distributed to all audio outputs applying a linear gain for each output.

spatialparams	atom	Sets the spatial segment distribution parameters regarding the current spatial distribution mode.
---------------	------	---

The following arguments are required for the different distribution modes:

- `direct` – (base) channel [int] (def: 0)
- `gains` – a linear gain for each output [list] (def: 0)

<code>window</code>	<code>symbol</code>	<p>Sets the segment synthesis window function.</p> <p>The following window functions are provided:</p> <ul style="list-style-type: none"> - <code>trapezoid</code> – trapezoid window (default) - <code>cosine</code> – cosine or <i>hann</i> window
---------------------	---------------------	--

[Information for box attributes common to all objects](#)

Examples

See Also

<i>Name</i>	<i>Description</i>
mubu	MuBu, multi-buffer container for sound description and motion capture data
mubu.granular~	MuBu granular synthesis