Keyboard Reference

Dynamical Systems Orchestration · v4.0 (MPE Edition)

Computer-keyboard MPE controller · Phase 1 player surface

Two windows, two modes

DSO v4.0 runs **two windows simultaneously**: an OpenCV video window showing the lamp with overlay graphics, and a pygame window that captures keyboard events. **The pygame window must hold OS keyboard focus** for the music keys to work — click on it before playing. The OpenCV window only receives the toggle keys at the bottom of this card.

The keyboard has two coexisting modes. **PERFORMANCE** mode plays notes on the currently selected voice slot. **EDIT** mode treats the QWERTY keys as buttons that walk through the slot's synth parameters and trigger lifecycle commands. Press `TAB` to toggle between them.

PERFORMANCE		EDIT
· Play notes on selected slot · **1–0** select slot · **[** / **]** shift octave · **space** = panic (all notes off)	**< TAB >**	· Scroll synth parameters · **1–0** select slot · **P W E F M** parameter family · **B N T D** binding commands

Slot selection (works in both modes)

The number row maps to the first ten of the sixteen voice slots. Slot selection happens immediately on key press; the selected slot is highlighted with a thicker ring in the video overlay and a row highlight in the side HUD. Slots 10–15 are reachable today only via an external MIDI controller (Phase 2) or programmatically.

1	2	3	4	5	6	7	8	9	0
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

PERFORMANCE mode

Plays MIDI notes on the currently selected slot. KEYBOARD_ONLY and HYBRID slots respond to notes; BLOB_ONLY slots ignore notes (use `B` in EDIT mode to claim them as HYBRID first). The lower QWERTY row covers one chromatic octave; the upper row covers the next. Pressing a note key sets the voice's *base_note*; the blob's routing modulation (if any) continues to be added on top.

Note layout

Upper octave (octave + 1) — sharps highlighted in lava-orange

Q	2	W	3	E	R	5	T	6	Y	7	U	I	O	P
C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C↑	D↑	E↑

Lower octave (current octave; default lower-row Z = C4 = MIDI 60)

Z	S	X	D	C	V	G	B	H	N	J	M	,	L	.
C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C↑	C#↑	D↑

Performance commands

Key	Action	Notes
`[`	Octave down	Lowers the lower row's C by 12 semitones. Range: 0–8 (default 4).
`]`	Octave up	Raises the lower row's C by 12 semitones.
`space`	Panic (all notes off)	Sends NOTE_OFF for every key currently held on the selected slot.
`TAB`	Switch to EDIT mode	Pressed keys are not held across the mode switch; released cleanly first.

EDIT mode

Each slot holds a full synth state: pitch (base_note, bend, scale, chord), wavetable (4 banks × 4 waves), envelope (ADSR), filter (cutoff, res, fmode for LP/BP/HP), plus the routing list that wires blob features to destinations. EDIT mode lets the player walk this state with a single letter for the family, arrow keys for the parameter, and `=` / `–` for coarse adjustment (10× the fine step).

Parameter family (selects what the arrow keys edit)

Key	Family	Parameters in order (← / → cycles)
`P`	pitch	base_note · bend · scale_id · chord_id
`W`	wave	wave_pos · wave_bank
`E`	env	attack · decay · sustain · release
`F`	filter	cutoff · res · fmode (0 = LP, 0.5 = BP, 1 = HP)
`M`	mod	routing_cursor (Phase 1.5 — see notes below)

Editing the focused parameter

Key	Action	Step size
← / →	Cycle parameter within current family	—
↑	Increment parameter by fine step	1 semitone · 0.05 amp · 50 Hz cutoff · 0.1 s envelope · etc.
↓	Decrement parameter by fine step	as above
`=`	Coarse increment (10× fine step)	12 semitones · 0.5 amp · 500 Hz cutoff · 1 s envelope
–	Coarse decrement (10× fine step)	as above
`R`	Reset focused parameter to default	Snaps back to SynthState.default()

Slot lifecycle commands

Key	Action	Effect on selected slot
`B`	CLAIM_HYBRID	Promote BLOB_ONLY → HYBRID. The voice's c_freq is back-computed into base_note so the first note key won't octave-jump. No-op on other states.
`N`	CREATE_EMPTY	Create a new KEYBOARD_ONLY voice in the next IDLE slot. Spawned silent (gate = 0); first NOTE_ON unmutes it.
`T`	TIE_BLOB	If an unbound tracked blob exists right now, immediately tie the most-persistent one to the selected KEYBOARD_ONLY slot (→ HYBRID). Otherwise queue the slot to bind to the next blob to be born.
`D`	DETACH	Break the blob bind on the selected slot. HYBRID and BLOB_ONLY voices become KEYBOARD_ONLY; their routings are dropped so the synth state stops being modulated.
`C`	Clear all routings	Removes every Routing on the selected slot. Useful when claimed-hybrid voices feel over-modulated and you want to play them like a static patch.

Key	Action	Effect on selected slot
TAB	Back to PERFORMANCE	Saves edit state on the slot.

OpenCV video-window keys

These keys work only when the OpenCV window has focus (clicking the video). They toggle overlays and adjust the persistence-threshold for blob detection — useful during calibration. Music keys do not work here; click the pygame window for those.

Key	Toggle / action
q	Quit application
v	Voice slot HUD on / off (right-side panel)
t	Tracking overlay (rings, IDs, info line)
o	Blob contour outlines
m	Merge tree overlay
f	Optical flow vector field
c	Wax-colour and combined mask windows
[	Persistence threshold – 2.0 (min 2.0)
]	Persistence threshold + 2.0 (max 80.0)

Voice-state ring legend

In the video overlay, every tracked blob bound to a slot gets a coloured ring. The ring colour encodes the slot's lifecycle state. The currently selected slot's ring is drawn thicker (3 px vs 2 px) and slightly larger (radius 22 vs 18).

Ring	State	Meaning
	BLOB_ONLY	Voice driven entirely by the blob's CV features. Default after a blob birth.
	KEYBOARD_ONLY	Voice has no blob bind. Player edits and plays it directly. Created via N (CREATE_EMPTY).
	HYBRID	Player plays notes; blob continues to modulate the synth. Reached via B (CLAIM_HYBRID) or T (TIE_BLOB).
	ZOMBIE	Voice's blob disappeared; soft sustain held for ~4 s while the system tries to resurrect onto a new blob nearby.

Common workflows

1. Take over a blob's voice and play notes against it

Watch the lamp. When a blob you want to play with is bound to a slot (coloured ring + S## label), select that slot and claim it as HYBRID. The blob keeps modulating cutoff / pan / amp; you control pitch.

```
TAB                     # enter EDIT mode
3                       # select slot 2 (1-indexed key, 0-indexed slot)
B                       # CLAIM_HYBRID  (BLOB_ONLY → HYBRID)
TAB                     # back to PERFORMANCE
z x c v g b ...         # play in C major over the blob's modulation
```

2. Pre-arm a blank voice for the next blob to appear

Useful for live performance: you've decided the next-born blob should have a particular timbre, and you want to set it up before the blob exists. Create empty, edit the synth, then queue the slot for tying.

```
TAB                     # EDIT mode
N                       # CREATE_EMPTY  (next IDLE → KEYBOARD_ONLY)
F                       # filter family
→ → →                     # cycle to fmode
= =                     # coarse-bump fmode toward HP
W                       # wave family
↑                       # bump wave_bank to 2 (formants)
T                       # TIE_BLOB — slot now waits for next birth
                        # When the next blob is born it binds here
                        # (instead of spawning a fresh BLOB_ONLY voice)
```

3. Edit a hybrid voice's filter while the blob plays

The most common live operation: a HYBRID voice is mid-passage and you want to push its cutoff up. The blob's routing on cutoff is additive to base_cutoff, so increasing base_cutoff pushes the whole envelope up.

```
TAB                     # EDIT mode (slot already selected)
F                       # filter family
(focus is on cutoff by default — the first param in 'filter')
↑ ↑ ↑ ↑                 # +50 Hz × 4 = +200 Hz on base_cutoff
= =                     # +500 Hz × 2 = +1000 Hz on base_cutoff
TAB                     # return to PERFORMANCE for note playing
```

4. Detach when the blob dies but you want the voice to continue

By default, when a HYBRID voice's blob dies and you've played it within the last 5 seconds, the system auto-detaches (HYBRID → KEYBOARD_ONLY). To force a detach earlier — for example, before the blob fades out — press `D` in EDIT mode.

```
TAB · D · TAB           # detach selected slot's blob bind
```

Phase 1.5 follow-up: the M / mod family currently parks the cursor for routing-list walking but does not yet emit ADD_ROUTING from arrow keys. To author routings programmatically today, call VoiceManager.add_routing(slot, src, dest, depth, curve). C in EDIT mode clears all routings on the selected slot.

Phase 2: an external MIDI controller (e.g. MIDI Fighter Twister) replaces the keyboard via the same Controller protocol. The MIDIController class in controller.py is a typed no-op stub today; activating it requires no downstream code changes.