

# COOPER

---

**Version 1.0**

**Maximilian Marcoll 2004/05**

<b>0. Introduction</b>	<b>4</b>
<b>1. Modules</b>	<b>5</b>
1.1 What is a module?	5
1.2 Loading and removing modules	5
1.3 Representation of modules	6
<b>2. Connecting modules</b>	<b>7</b>
2.1 A simple Connection	7
2.2 Controlling the matrix	7
<b>3. Getting sound out of COOPER</b>	<b>7</b>
3.1 Two important shortcuts	7
3.2 The mixer	7
3.3 Displays	8
3.4 Recording	8
<b>4. Fast access</b>	<b>9</b>
4.1 What?	9
4.2 How?	9
<b>5. Local figures</b>	<b>10</b>
5.1 What is a figure?	10
5.2 Recording	10
5.2.1 local	10
5.2.2 Using the representation	10
5.2.3 Using the fast access feature	10
5.3 Playing back	11
5.3.1 local	11
5.3.2 Using the fast access feature	11
<b>6. Advanced connections</b>	<b>12</b>
<b>7. Global figures</b>	<b>13</b>
7.1 Recording	13
7.2 Playing back	14
7.3 The matrix in global figures	14

<b>8. Integration of Modules</b>	<b>15</b>
<b>8.1 A generator: SawOsc</b>	<b>15</b>
<b>8.1.1 The template</b>	<b>15</b>
<b>8.1.2 Conditions</b>	<b>15</b>
<b>8.1.3 Input Handling</b>	<b>15</b>
<b>8.1.4 Naming objects</b>	<b>16</b>
<b>8.1.5 Messages to COOPER</b>	<b>17</b>
<b>8.1.6 Connections</b>	<b>18</b>
<b>8.2 Modulator vs generator</b>	<b>19</b>
<b>8.3 Connecting a module to the fast access feature</b>	<b>19</b>
<b>9. Custom Templates</b>	<b>19</b>
<b>10. Shortcuts</b>	<b>20</b>
<b>11. Where to save things</b>	<b>20</b>
<b>12. The figure format</b>	<b>21</b>
<b>13. Integrated midi connections</b>	<b>22</b>

## **0. Introduction**

This document is an introduction to the environment COOPER.

All features are discussed and exemplified where necessary.

COOPER is written in MaxMSP.

The idea was to create a flexible environment which can be controlled from meta-parameter-levels without losing direct control of single parameters or giving responsibility to the computer.

The audio processing is encapsulated into modules, which get connected through a matrix.

Every input a user sends to a module can be recorded in figure files for later playback.

These figures allow the generation of complex musical structures without the need to control every single parameter by hand.

It's even possible to create figures with other programmes, since the entire figure management is based on text files.

In addition, COOPER can easily be extended with your own modules.

This document assumes that you are familiar with MaxMSP.

Particularly this is required for the chapters about integrating custom modules.

# 1. Modules

## 1.1 What is a module?

COOPER is an environment (framework). It provides a lot of things, but it can't produce any sound on its own.

For producing sound COOPER needs modules.

A module is a patch (plug-in), which can be integrated in the COOPER environment.

There are two different classes of modules:

generators (gen) and  
modulators (mod)

A Generator generates sound, whereas a modulator modulates sound.

## 1.2 Loading and removing modules

First of all, there are different interfaces for different screen resolutions.

Open the Patch with the resolution of your display (e.g. COOPER\_1024x768).

In the lower right portion of the patch there are fields for loading and removing modules:



To load a module, select it from one of the two menus in the left portion.

To remove a module, enter its number in the corresponding number box on the right, or, to remove all modules at once, click on the „remove all“ button.

When you load a module, it's displayed on the screen.

Generators get loaded at the top of the screen, modulators are loaded to the bottom of the screen.

Using the button labelled „write init“ you can save the current setup (all loaded modules) in a textfile. (Resulting files will be saved in COOPER/lib/ named Nsetup.txt .)

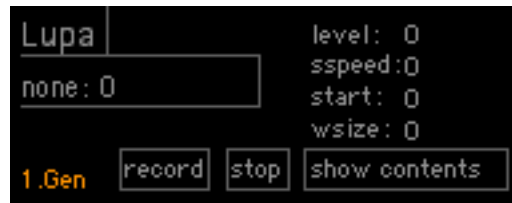
The button labelled „load init“ opens a file open dialog to load such a file.

All the modules will be loaded automatically.

### 1.3 Representation of modules

In fact what you see when you load a module is not the module itself, but a representation of the module.

The representations of modules all share the same layout:



In the upper left corner there's the name of the module.

On the right is a list with the most important parameters.

Below the name there is a field displaying the active parameter (we will discuss active modules in just a moment).

In the lower left corner the number of the module is shown (zero-based).

To the right of the number there are three buttons (record, stop and show contents).

The representation of a module can not be used for controlling a single parameter. For controlling the module directly, you have to open it.

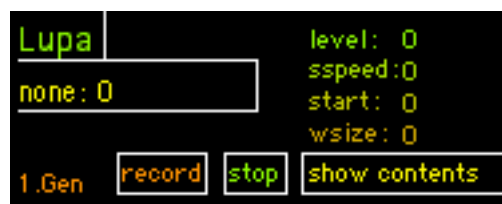
Click on the button labeled „show contents“ to open a module.

The corresponding patch is opened and you can now control it directly.

If the module should be controlled with the computer keyboard or other external devices (like midi controllers), the data of these devices gets sent to the module only when it is active.

To activate a module, click on its representation, but of course not in one of the three buttons.

The module is now active and the representation changes its appearance:



## **2. Connecting modules**

### **2.1 A simple connection**

To send a signal from one module to another, you have to establish a connection between the two modules.  
You can do this using the matrix, in the right portion of COOPER's interface. On the X-axis there are all the generators, on the Y-axis there are the modulators.  
If you loaded any modules please remove them now and load the generator SineOsc (a simple sine wave oscillator) and the modulator DirOut. The modulator DirOut sends the incoming signal directly to the mixer, without any further processing.

### **2.2 Controlling the matrix**

With a click in the appropriate field in the matrix (in this case 0/0) You create a connection between the two modules.



But there is another possibility to control the matrix:  
Press and hold the ctrl-key.  
A cursor appears. You can move the cursor with the arrow keys.  
To change the status of a field in the matrix press enter.  
(But keep holding the ctrl-key down.)

## **3. Getting sound out of COOPER**

### **3.1 Two important shortcuts**

Press option+shift+p to open the DSP status window.  
To turn audio on and off, press option+shift+a.

### **3.2 The mixer**

Turn audio on.  
Open the module SineOsc and set up a signal using the level fader and the numberbox (pitch).  
The levelmeter should give you appropriate feedback.  
In the centre of the COOPER-interface there's the mixer.

For every Modulator there is a fader. On the right there is the master fader. Push the fader number 0 (the leftmost one) and the master fader to reasonable values and you will hear the sine wave.

### **3.3 Displays**

In the centre of the COOPER-interface, above the mixer, there are a few displays. On the top there is the status display (audio on/off). Next are the record-button and a time display. This display shows the elapsed time since you turned audio on.

### **3.4 Recording**

COOPER can record the audio output to an aiff file. To start the recording click on the record button (labeled "recording off"). (You can also use the shortcut `optio+shift+r` to turn recording on /off.) Resulting files will be named `cooper01.aiff` to `cooper99.aiff`. They will be saved in `COOPER/lib/`. If there are already files named `cooperxy.aiff`, they will not be overwritten.

## 4. Fast access

### 4.1 What?

Below the displays in the centre there is the fast access feature control for module-parameters.

Here you can control two parameters of a module directly.



### 4.2 How?

Press the space bar.

The fast access feature control will be highlighted.

Now you can choose a module with the keyboard:

Select a generator with the keys 1 2 3 4 5 6 7 8 9 0 B `

or select a modulator with the keys q w e r t z u i o p ü +

If the selected module supports the fast access feature you can now control the two parameters the module made available for fast access by clicking and dragging in the two-dimensional fader.

The fast access feature works even if the selcted module is not active!

## **5. Local figures**

### **5.1 What is a figure?**

All input data a module receives from the user (from mouse, keyboard or other external devices) can be saved in textfiles. These files are named figures.

### **5.2 Recording**

#### **5.2.1 local**

Open a module (e.g. SineOsc).

At the bottom of the patch there are two buttons:

record and  
stop

Using these buttons you can control the recording of local figures.

Click on „record“.

In the console (above the lower buttons of the module) you see an appropriate message.

Move the level-fader and click „stop“. The recording was stopped and the console shows an appropriate message.

When you click on „record“ the current state of the module will be integrated into the figure as the initial state.

The naming of resulting figure files follows the following scheme:

<day>-<month>-<hour>-<minute>-<second>.txt

You can of course change the names after the recording.

#### **5.2.2 Using the representation**

You can start the recording of a local figure without having the module open. It has to be active for this to work though.

Close the modules window and activate it by clicking on itt representation.

Click on the button labeled „record“ in the representation to start the recording and in „stop“ to stop the recording.

#### **5.2.3 Using the fast access feature**

Even input data given to module by using the fast access feature can be recorded into figures. Press the space bar to activate the fast access feature and select a module with the keyboard.

Activate the selected module and start the recording process.

Now control the module using the two-dimensional fader and stop the recording.

## 5.3 Playing back

### 5.3.1 local

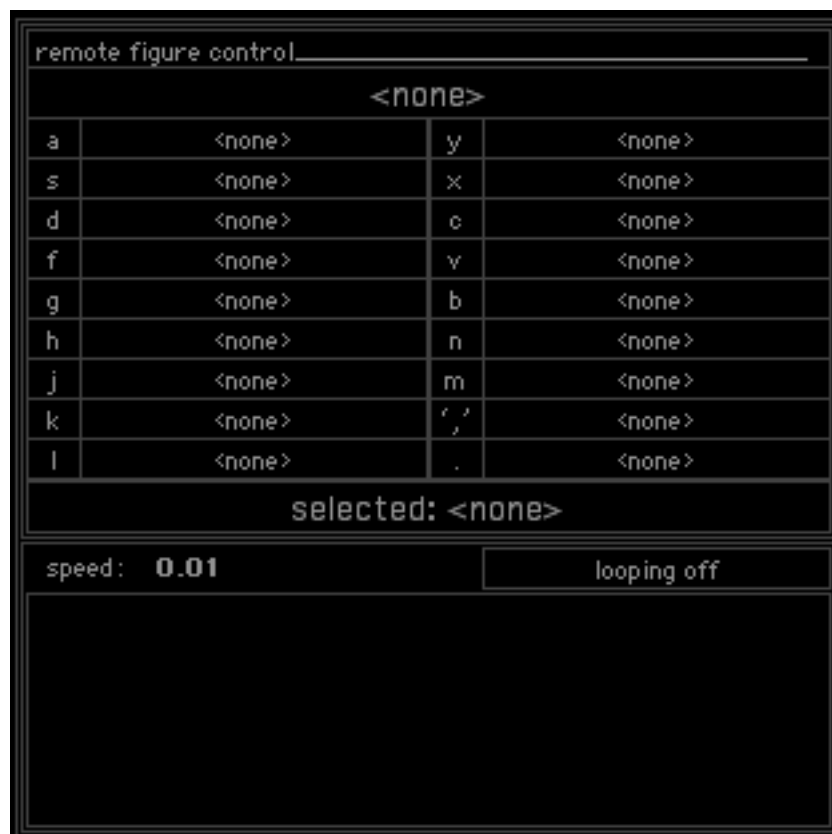
Above the buttons for starting and stopping the recording of figures in a module's patch there is a menu object with all the available figures for the module.

To play back a figure simply select the desired figure in the menu.

Use the numberbox labelled „speed“ to control the playback speed of the figure.

### 5.3.2 Using the fast access feature

In addition to the fast access feature for module parameters, there is a fast access feature for the playback of local figures. It is hiding in the left portion of the COOPER-interface:



Press caps lock to activate the fast access feature control.

It gets highlighted. Now you can select a module using the keyboard.

As in the case of the fast access feature control for module parameters, select a generator with the keys 1 2 3 4 5 6 7 8 9 0 ß ´

and a modulator with the keys q w e r t z u i o p ü +

When a module is selected, all available figures are shown in the table. For playing back a figure simply press the associated key on the keyboard (a s d f g h j k l y x c v b n m , .). You can control the playback speed using the slider labeled „speed“. In addition you can turn automatic looping of a figure on or off with the button labeled „looping off“.

Local figures may contain figure calls. Meaning that a figure can start the playback of another figure. Record a figure. Now start the recording of a second figure and start the playback of the first figure. Change some values manually and stop the recording. When you play back the second figure, you will notice that COOPER does not simply copy the data of the first figure into the second, but that there is a real figure call inside of the second figure file. You can create figures with as many figure-call-levels as you want. (A figure can start a figure, which starts a figure, which starts a figure, which starts a figure, which starts a figure...)

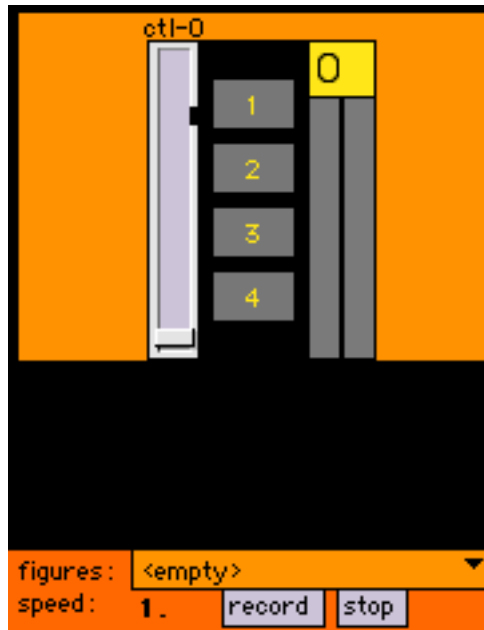
## **6. Advanced connections**

A generator can be connected to as many modulators as there are slots for them. But so far the signal of a generator can only once pass a modulator before it is sent to the mixer. To overcome this limitation, COOPER provides busses.

Every modulator (except DirOut) has a small matrix at the bottom of its patch. Using this a matrix you can send a modulator's output signal to one of four busses.



The signals of these busses can be sent to other modulators using the generator FrwrD.



This is the generator Frwr.

Select the bus you want to reintegrate in your signal network with the matrix inside of Frwr and move the fader to a reasonable value.

Now you can connect the signal of the selected bus to other modulators, just like you do with any other generator.

## 7. Global figures

### 7.1 Recording

Local figures contain data for one particular module.

Unlike local figures, a global figure may contain data for all modules loaded.

The controls for global figures are located on the left of the mixer:



To start the recording of a global figure, simply click on „record“. To stop the recording click on „stop“.

## 7.2 Playing back

Simply select a figure from the menu.

The playback speed can be controlled using the numberbox labeled „speed“.

## 7.3 The matrix in global figures

It is possible to include data from the matrix into global figures.

Above the upper left corner of the matrix there is the corresponding control item:

include matrix data in global figures :

## **8. Integration of modules**

### **8.1 A generator: SawOsc**

#### **8.1.1 The template**

In order to make the integration of modules as simple as possible there are two templates as a starting point.

For generators use the file TemplateGen.

If you want to write a modulator use the file TemplateMod.

#### **8.1.2 Conditions**

If a patch should be integrated into the COOPER environment, there are some conditions to fulfill. Otherwise COOPER will not be able to register or to communicate with your module.

Most of the steps necessary to inform COOPER about a module is done automatically (scripting).

But a few things have to be done by the author of the module.

If you write a new module, the first thing to do is to create a copy of the template file.

While working on the module, it should definitely be saved in COOPER/lib/. Otherwise important subpatchers will not be found.

Create a copy of the file TemplateGen and rename it to SawOsc.

Open the file and create a subpatch with a name of your choice.

This subpatcher will contain all the objects necessary for the processing of your module.

In this case you only need a number box, a [saw~]-object, an inlet and an outlet.

Connect the inlet to the numberbox, the numberbox to the [saw~] and the [saw~] to the outlet.

That is it.

#### **8.1.3 Input Handling**

In the case you want to use an external device (keyboard, midi-controllers etc.) insert the corresponding code into the subpatch [p input] which is already created for you.

Even if you do not want to use external devices for controlling your module, do not delete this subpatch!

Make sure that the output of this subpatch (which should have only one outlet!) follows the form <parameter> <value> (e.g. level 127).

## 8.1.4 Naming objects

Now let us design your user interface.

I strongly recommend to use the objects of the templates.

If you want to exchange objects, please have a look at the chapter *Custom templates*.

You have to have one object for each parameter of your module which is directly connected to the corresponding processing-object.

For example it is very reasonable to create a fader object and to connect it to the left inlet of the left [gain~]-object.

If you created such an object (responsible for receiving values from external controlling-devices and/or for sending values to the processing subpatch) please give it a name (using the name... dialog).

Your module will be registered to COOPER using these names.

In return, COOPER will take care of the connections to the external devices (if any), to the COOPER-environment and to the figure management.

So, create a fader, give it a name (e.g. levelfader) and connect it to the left inlet of the left [gain]~-object.

Now you need a numberbox for the frequency.

Insert a numberbox into the patch, connect it to the inlet of your processing-subpatch and give it a name.

## 8.1.5 Messages to COOPER

In the template there are two messageboxes you have to edit:

The first message starts with „parameternames“.

This message is going to be sent to the representation of your module.

The message is necessary for the representation in order to be informed about what it actually should display.

The messagebox whose content begins with „parameternames“ contains multiple messages, separated by commas.

These messages are: parameternames, name and number.

Replace <none> with the names of the parameters of your module, separated by spaces, and with the name of your module (in the case of the name message). It is not necessary to supply all parameters.

The representation is not able to display more than 5 parameters.

So just insert the most important parameters of the module.

The argument of the message „number“ does not need to be changed.

It will automatically be set to the number of the module when it is loaded.

The second messagebox you have to edit begins with „parameters“.

The messages in this box will connect your module to COOPER.

The messages are: parameters, objectnames, name, number and create.

Insert all parameters of the module after the word „parameters“, separated by spaces.

The message „objectnames“ is a little bit more complicated.

Its arguments are the names of the objects used for connecting (the ones you assigned names to) and the number of the inlet and the outlet which should be used for the connections.

An example for that:

You will probably use numberboxes and faders for your parameter-values.

But it is also possible that the value of a parameter is not a number, but a symbol (the name of a sound file or a path to a folder etc.).

In this case you could want to use the middle outlet of a menu object to be used for the connection.

You couldn't do this if you could not tell COOPER which inlets and outlets to use.

The format of the objectnames message is the following:

```
objectnames <object1> <inlet> <outlet> ... <objectN> <inlet> <outlet>
```

The numbering of inlets is zero based left to right.

The order of the arguments of the objectnames message must be the same as the order you used in the parameters message in reference to the parameters!

Insert the name of your module after the word „name“.  
You do not need to change the argument of the „number“ message.  
The message „create“ will not get any arguments.  
It will initiate the connection to COOPER.

### **8.1.6 Connections**

Do not trigger the messageboxes!

If you by accident did trigger the messageboxes,  
click the undo message (below the [thispatcher] object) to remove  
automaticaly created objects and connections.  
(Error-messages in the Max window are normal.)

In order to be able to record and play back figures, your module needs  
a folder for the figure files.

Have a look at the chapter *Where to save things*.

It also answers the question where to save your module.

## 8.2 Modulator vs. generator

The development of modulators is almost identically to the development of generators.

The only difference:

You have to connect the [receive~ #1ModIn1] and [receive~ #1ModIn2] Objects to the soud input of your processing subpatch.

## 8.3 Connecting a module to the fast access feature

in COOPER/lib/ there is a text file named TPmodulesParam.txt.  
this file contains lines of the following format:

```
<modulname>, x <x-parameter> <min> <max> y <y-parameter> <min> <max>;
```

Insert an appropriate line for your module.

If your module should not be connected to the fast access feature you should insert the follwing line:

```
<modulename>, x <none> 0 0 y <none> 0 0;
```

## 9. Custom templates

Below is a list with necessary objects/objectnames for a module.  
Only user-interface-objects are covered.

- A [ubumenu]-object  
name : figuremenu  
(Must have at least one entry!)
- A nummerbox  
name: figurespeedbox
- Two [ubumenu]-objects with [ubutton] objets on top of them  
(conections: [ubutton]:1 -> [ubumenu]:0)  
names of the [ubumenu]-objects: recordbutton and stopbutton
- A [jsui]-object (console.js)  
name: console

In the case you (accidently) removed it:

- A subpatcher with one inlet and one Outlet.  
name: input

## 10. Shortcuts

ctrl	(hold) activates the keyboard control of the matrix. Move the cursor with the arrow keys. Switch the state of a field with the enter key.
Caps Lock	activates the fast access feature for local figures. Select a module with 1 2 3 4 5 6 7 8 9 0 β ´ (gen) and q w e r t z u i o p ü + (mod). Select a figure with a s d f g h j k l y x c v b n m , .
Space	activates the fast access feature for module parameters Select a module with 1 2 3 4 5 6 7 8 9 0 β ´ (gen) and q w e r t z u i o p ü + (mod).
Option+Shift+p	opens the dsp status window
Option+Shift+a	turns audio on/off
Option+Shift+r	turns recording on/off

## 11. Where to save things

<b>figures</b>	are saved in COOPER/lib/figure-files/ This folder has three subdirectories:  gen/ global/ and mod/  Global figures are saved in global/.  Local figures of generators are saved in subdirectories of gen/ which have the same names as the modules the figure files are for.  Local figures of modulators are saved in subdirectories of mod/ which have the same names as the modules the figure files are for.
<b>generators</b>	are saved in subdirectories of COOPER/lib/gen/ which have the same name as the module which they contain.
<b>modulators</b>	are saved in subdirectories of COOPER/lib/mod/ which have the same name as the module which they contain.

## 12. The figure format

You can write figure files by hand or generate them with other programs.  
The format of a local figure file entry is:

```
<delay> <command> <value(s)>;
```

The format of a global figure file entry is:

```
<delay> <module> <command> <value(s)>;
```

delay            the time since the execution of the previous line in milliseconds

module           specifies a module to send commands to.

command         can be any parameter you have in the module  
(in the case of a figure-call this is "play")

value(s)         the value the parameter should be set to.  
in the case of a figure-call this is the name of a figure file  
(with or without path) and the speed.

for instance:

```
0  0Gen  level 127;  
10 0Gen  pitch 440;  
0  0Gen  play figure1.txt 1;
```

In the case you included matrix data in a global figure (or you want to do so):  
The command looks like this:

```
<delay> matrixFigureCommands <x> <y> <value>;
```

for example:

```
0 matrixFigureCommands 0 0 1;
```

Meaning that the first gen gets connected to the first mod.

Note that each line has to end with a semicolon!

## **13. Integrated midi connections**

The faders of the mixer are connected to the midi controler ids 16 to 27 by default.

The master fader is connected to the midi controler id 31.

These assignements can be changed in the file COOPER/lib/midiMixer.