**Sergi Jordà**
Music Technology Group, Audiovisual Institute
Pompeu Fabra University
Passeig de la Circumval·lació 8
08003 Barcelona, Spain
sergi.jorda@iua.upf.es

# FMOL: Toward User-Friendly, Sophisticated New Musical Instruments

The design of new instruments and controllers for performing live computer music is an exciting field of research that can lead to truly new music-making paradigms. The ever-increasing availability of sensing technologies that enable virtually any kind of physical gesture to be detected and tracked has indeed sprouted a wealth of experimental instruments and controllers with which to explore new creative possibilities. However, the design of these new controllers is often approached from an essentially technical point of view in which the novelty of the sensing technologies deployed overshadows the attainable musical results. Although new instruments are not constrained to the physical restrictions of traditional ones, an integrated approach in which the instrument is designed as a whole and not as a combination of arbitrary input and output devices can lead to the creation of more rewarding and more musical instruments.

This article describes an attempt at an integrated conception, called F@ust Music On-Line (FMOL), which is a simple mouse-controlled instrument that has been used on the Internet by hundreds of musicians during the past four years. FMOL has been created with the complementary goals of introducing the practice of experimental electronic music to newcomers while trying to remain attractive to more advanced electronic musicians. It has been used by musicians of diverse skills for the collective composition of the music of two important shows, including one opera of the Catalan theatre group La Fura dels Baus. It is also being played in live concerts.

Computer Music Journal, 26:3, pp. 23–39, Fall 2002


## New Musical Instruments and New Music-Making Paradigms

The conception and design of new musical interfaces is a burgeoning multidisciplinary area where technological knowledge (sensor technology, sound synthesis and processing techniques, computer programming, etc.), artistic creation, and a deep understanding of musicians' culture must converge to create new interactive music-making paradigms. If new musical interfaces can be partially responsible for shaping future music, these new musical paradigms should not be left to improvisation. We are not asserting that the design of new instruments must follow any given tradition, but when trying to define new models, we cannot ignore existing ones. A wide knowledge of these, together with the personal beliefs and intuitions of each designer, should provide orientation about what could be changed and what could be kept in incoming designs.

Prior knowledge may originate from different fields or practices, but we could roughly consider that new instruments designers have three major backgrounds to deal with: the first is a millennial one, as old and rich as the history of music-making, while the two others are less than one half-century-old. We are talking about computer music and the realm of human-computer interaction (HCI). What we are suggesting is that new musical controllers must obviously include new capabilities that have been unavailable and even unimaginable in traditional mechanical instruments. However, they should also try to recover essential components from the acoustic legacy that have been left out during the recent decades of computer music owing to the former technical limitations in digital technologies. Moreover, they could benefit as well from all the existing corpus of knowledge and research in non-musical areas of HCI (Wanderley 2001; see also www.csl.sony.co.jp/~poup/research/chi2000wshp/papers/orio.pdf).

## Musical Interfaces Are Not Musical Instruments

Musical instruments transform the actions of one or more performers into sound. An acoustic instrument consists of an excitation source under the control of the performer(s) and a resonating system that couples the vibrations of the excitation to the surrounding atmosphere. This affects, in turn, the precise patterns of vibration. In most acoustic instruments (apart from organs and other keyboard instruments) the separation between the input and the excitation subsystems is unclear. Digital musical instruments, on the other hand, can always be divided into a gestural controller or input device that takes control information from the performer(s) and a sound generator that plays the role of the excitation source. Digital instruments may also include a virtual resonating system, but real resonators are usually missing and substituted by a digital-to-analog converter (DAC) that feeds an amplifier connected to a speaker system. Often, when discussing new digital instruments, we are referring in fact only to an input device component. In that sense, the Theremin is a complete musical instrument, while a MIDI Theremin is only a controller or an interface, and not necessarily a musical one.

The separation between gestural controllers and sound generators as standardized by MIDI led to the creation and development of many new alternative controllers with which to explore new creative possibilities (e.g., Chadabe 1996; Wanderley and Battier 2000). However, this separation should not always be seen as a virtue or an unavoidable technical imposition. The most frequent criticisms of this division deal with the limitations of the protocol that connects these two components of the instrument chain. Are today's standard music communication protocols (e.g., MIDI) flexible enough, or are the potential dialogues between controllers and generators limited or narrowed by these existing protocols? MIDI does indeed have many limitations (low bandwidth, low resolution, inaccurate timing, half-duplex communications, etc.) (Moore 1988), but we optimistically believe that there is still room for experimentation and improvements while remaining within MIDI's constraints. A recent study toward that direction was offered by Goebl and Bresin (2001), who investigated the measuring and reproducing capabilities of a Yamaha Disklavier MIDI piano. They concluded that the weakest point of that system does not lie in the recorded MIDI information but in how this information is used for reproduction.

Our complaint is much more banal but at the same time frequently overlooked, if we consider the number of generic new musical controllers designed today. Is it possible to design nonspecific controllers, to develop highly sophisticated interfaces without a profound assumption of how the sound or music generators will work? Apart from the truism that not all combinations of controllers–generators are equally satisfying (nothing will match a MIDI wind-controller when driving a physical model synthesizer of a woodwind instrument), the final expressiveness and richness of any musical interface (controller) cannot be independent of its generator and the mapping applied between them.

We firmly believe that a parallel design between controllers and generators is needed to create new sophisticated musical instruments and music-making paradigms. Both design processes should therefore be treated as a whole. A complementary approach to this separation problem is proposed by Ungvary and Vertegaal (2000), creators of the SensOrg Cyberinstrument, a modular and highly configurable assembly of input/output hardware devices designed for optimal musical expression.

## Mapping and Feedback Strategies

Whether following a parallel design or maintaining two independent approaches, the physical and logical separation of the input device from the sound production necessitates multiple ways of processing and mapping the information coming from the input device. Mapping becomes therefore an essential element in designing new instruments.

Mappings in acoustical instruments are often multidimensional and slightly nonlinear. Blowing harder on many wind instruments not only affects dynamics but also influences pitch, and in these control difficulties lie much of their expressiveness. As Johan Sundberg suggests, the voice, possibly the most expressive instrument ever, can also be considered to be one of the most poorly designed from an engineering point of view (Sundberg 1987), taking into account the complex relationship between articulator movement and formant frequency changes that makes articulation/formant frequency a one-to-many system.

For the sake of simplicity, many electronic instruments use basic one-to-one linear mappings. More complex mappings such as one-to-many, many-to-one, or many-to-many, with optional nonlinear relations, memory, and feedback (so that the output is not only a function of the last input value, but also of previous input and/or output values) involve deeper design strategies. Such strategies do not exist in a separated conception approach. They also require more demanding programming techniques not easily attainable in the environments commonly used for mapping design purposes, such as Max and similar data-flow graphical programming languages (Puckette 1988; Puckette and Zicarelli 1990; Winkler 1998).

Memory of the prior input values is needed, for instance, if we want mappings capable of detecting performers' physical gestures. (See, for example, Wanderley and Battier 2000, for a survey of gesture in the musical domain.) On the other hand, feedback (that is, output control values are dependent on previous output values) can be used for modifying the generator responsiveness. Simple feedback does not imply the need of a bi-directional communication between the controller and the generator, as the mapping task is often handled by the generator, which can easily store previous output values in a memory buffer. However, bi-directional communications may be indeed one of the solutions leading to wider and richer communication between controllers and generators and, as a result, between players and their musical output.

Duplex communications allow the existence of haptic and force feedback manipulations, which may bring richer dialog loops between the instrument and the performer (Bongers 1994; Gillespie 1999). However, force feedback is not the only feedback loop present in traditional music playing. An acoustical instrument resonates, affecting its patterns of vibration, and this acoustic feedback that takes place inside the instrument is fundamental to the overall sound production process. We could metaphorically imagine that by this feedback, the instrument is being ''conscious'' of the sound it is producing and that this information is also given back to the performer.

Digital controllers, on the other hand, generally are not ''aware'' of the output they are generating. Embedding this feedback information into an instrument which may be sounding though a speaker several meters away may seem at first an absurd task. We will show later how in the case of the FMOL instrument, the audio feedback presented to the performer in an interactive visual form intuitively helps the understanding and the mastery of the interface, enabling the simultaneous control of a high number of parameters that could not be possible without this visual feedback.

### ''Mouse-Music''

The previous statements may bring to mind sophisticated and costly hardware controller devices that involve the work of specialized engineers. Moreover, the increasing availability of new sensing technologies and analog to MIDI interfaces (ADB I/O, I-Cube, Sensorlab, SAMI, AtoMIC Pro, etc.), or even cheaper and easily programmable microcontrollers such as the Basic Stamp, has led many people to assert that there are no reasons to do ''mouse-music'' anymore.

We believe, however, that the research for clever mappings and better communication among the different components of an interactive music system does not imply that low-cost and widely available input devices such as mice, joysticks, or computer keyboards should be considered completely obsolete. On the contrary, there is still a lot of research to be done in the area of interactive graphical user interfaces (GUIs).

During two decades of widespread use, the mouse has proven to be an invaluable controller for interacting with computers GUIs (e.g. Card, English, and Burr 1978; Rutledge and Selker 1990; Douglas and Mithal 1994). However, many kinds of GUIs exist, and the live performance of music—with such specific topics as simultaneous multiparametric control and timing—is far more demanding than many other HCI activities (Wanderley 2001; see also www.csl.sony.co.jp/~poup/research/chi2000wshp/papers/orio.pdf). No matter how fancy they may look, it is also true that many GUIs do still emulate physical knobs, buttons, and sliders. In these cases, playing music with real physical knobs or sliders will undoubtedly allow for a faster and easier simultaneous multiparametric control (Hunt and Kirk 2000).

However, not all GUIs follow this model. A clever musical exception can be found, for example, in Music Mouse, the interactive music software developed by Laurie Spiegel in the mid 1980s (Gagne 1993). There are no hidden knobs or sliders on its screen; this program is the perfect paradigm of a musical instrument designed to be played with a mouse. Its outstanding musical results can be heard on the compact disc *Unseen Worlds*, which contains live improvised pieces by the author of the software (Spiegel 1991).

Mice (and joysticks to a minor extent) have another important quality: they are universally available, which is not a minor feature when considering the design of popular and ''democratic'' new music interfaces. For that reason, FMOL is also a mouse-driven musical instrument (which allows simultaneous multiparametric control).

## Who Will Be Using These Instruments?

Creating music in real-time using computer-based new instruments also poses new questions about who will be using them, and how.

### *Individual vs. Collective Performance*

Although performing music has typically been a collective event, traditional musical instruments have been mostly designed for individual use (even if some of them, like the piano or percussion instruments, can be used collectively). This restriction can now be freely avoided when designing new interfaces, which leads to a new generation of distributed instruments with a variety of possible different models following this paradigm (e.g., statistical control, role-playing performers, etc.; see for example Weinberg and Gan 2001). Implementations of musical computer networks date back to the late 1970s with performances by groups like the League of Automatic Music Composers (Bischoff, Gold, and Horton 1978) or the Hub in the 1980s (Gresham-Lancaster 1998). This collective approach may also be common to many interactive sound installations that respond to public movements and gestures (see, for example, Ulyate and Biancardi 2002), which leads us to another important point: the skills and the know-how of the performer(s).

### *Dilettante vs. Professional Performers*

I have developed several computer-based interactive music systems since 1989. Some of them, like PITEL or Afasia, were conceived for trained musicians or even for specific performers (Jordà 1991, 2001), while others, like Epizoo, were meant to be controlled by members of an audience in public performances (see www.telefonica.es/fat/ejorda.html). The demands for the two genres are obviously different. Complicated tools, which offer great freedom, can be built for the first group; the second group, however, demands simple but appealing tools that, while giving their users the feeling of control and interaction, must produce ''satisfactory'' outputs (e.g., Ulyate and Bianciardi 2002). These two classes are often mutually exclusive. Musicians become easily bored with the ''popular'' tool, whereas the casual user may get lost with the sophisticated one. Is this trend compulsory? Is it not possible to design instruments that can appeal to both sectors: tools that, like many traditional acoustical instruments, can offer a "low entry fee with no ceiling on virtuosity" (see www.csl.sony.co.jp/~poup/research/chi2000wshp/papers/wessel.pdf)? This is what FMOL is all about.

Often, more difficult-to-master instruments lead to richer and more sophisticated musics (e.g., the piano vs. the kazoo), but expressiveness does not necessarily imply difficulty. In that sense, one of the obvious research trends in real-time musical instruments design can be the creation of easy-to-use and, at the same time, sophisticated and expressive systems (considering that the best way to understand and appreciate any discipline, whether artistic or not—and music is no exception—is by "doing" and being part of it). More efficient instruments that can subvert the aforementioned effort-result quotient will bring new sophisticated possibilities and the joy of real-time active music creation to non-trained musicians. Let us try, as Robert Rowe suggests, to develop virtual instruments and musicians that do not just play back music for people, but become increasingly adept at making new and engaging music with people, at all levels of technical proficiency (Rowe 1992, p. 263).

### *The Kalimba and the Scissors Paradigm*

At the opening session of the New Interfaces for Musical Expression (NIME) workshop that took place in Seattle during the CHI 2001 conference, Michael Lyons showed the kalimba as an example of a "great musical instrument." The kalimba is truly easy and fun to play, and its musical output can be very interesting and sophisticated. Most people would agree that the kalimba is a "better" instrument than the kazoo, but what makes this so?

Going back several centuries in the history of instrument design, we can see that some instruments are more difficult to play than others. Wind instruments, for instance, are frequently hard to blow; the absolute novice cannot produce any controllable sound. (There are exceptions, as with the recorder family.) Fretless string instruments are impossible, for a novice, to play in tune. Also, with multistringed fretted instruments, the novice may have a difficult time deciding which string to use among all possible options to obtain a desired pitch. Keyboard instruments are in all senses easier: they produce sound easily, and there is a bijective relationship between pitches and keys. Extremely complex and difficult music has been written for the piano, but a non-musician can still improvise beautiful music on the instrument. Still, for many reasons, the piano can be a much more intimidating instrument than the kalimba.

Kalimbas have few notes (all of them "correct" ones). Its shape and construction intuitively invites everyone to play it the correct way: with the thumbs. This recalls Donald Norman's discussion of scissors:

> Consider a pair of scissors: even if you have never seen or used them before, you can see that the number of possible actions is limited. The holes are clearly there to put something into, and the only logical things that will fit are fingers. The holes are affordances: they allow the fingers to be inserted. The sizes of the holes provide constrains to limit the possible fingers: the big hole suggests several fingers, the small hole only one. The mapping between holes and fingers—the set of possible operations—is suggested and constrained by the holes. Moreover, the operation is not sensitive to finger placement: if you use the wrong fingers, the scissors still work. You can figure out the scissors because their operating parts are visible and the implications clear. The conceptual model is made obvious, and there is effective use of affordances and constraints (Norman 1990).

It is true that the kalimba has little dynamic control, which further restricts its freedom, but it cannot be considered as a restricted instrument, at least compared with the average public interactive sonic installation. It is a well-designed instrument "for all ages."

## Should We Search for a Standardized Instrument?

Not many recent electronic instruments have attained even the reduced popularity of the Theremin, created around 1920. The next standard electronic instruments after the Theremin may have really not arrived yet. Or they have, in fact, but they are still not digital ones. During the 1960s, the electric guitar led to the birth of new kinds of sound and playing techniques, completely different from the ones attainable with its acoustic counterpart, and it is safe to assert that the origins of rock music come precisely from the use of this new instrument. Since the 1980s, the turntable, played in a radically unorthodox and unexpected manner, has become a real musical instrument that has developed its own musical culture, techniques, and virtuosi (e.g., Poschardt 1995; Shapiro 1999).

For Leonello Tarabella, the fact that it is not easy to define how to use the computer as an instrument the way one can with traditional instruments is clear proof that we still are at the ''Stone Age'' of computer music live performance (see www.iua.upf.es/~abarbosa/docs/mosart-interactivity_pannel.pdf). Are we supposed to see a digital instrument standardization? Do we really need ''the'' new musical controller? Probably not. On the other hand, highly idiosyncratic instruments that are used only by their respective creators may not be the best strategy for a serious evolution in this field.

## FMOL (F@ust Music On-Line)

The FMOL project (1997–2002) started when the Catalan theatre group La Fura dels Baus proposed to me the conception and development of an Internet-based music composition system that could allow cybercomposers to participate in the creation of the music for La Fura's next show, F@ust 3.0, freely inspired by Goethe's work. I knew I did not want to allow people to compose General MIDI music on a keyboard and send us attached MIDI files via electronic mail. There were also several ideas with which I wanted to experiment, the first of which was collaborative music on the Internet (Jordà 1999). The second was the conception of a real-time composition system that would work directly with sound more than with macro-structural parameters or, put another way, a synthesizer that could deal directly with form (Jordà 2001). I also wanted this tool to be simple and complex at the same time, so that it would not dishearten amateurs but would still be able to produce very diverse musics. I wanted to allow rich and intricate control to people with various stages of training and different learning curves.

Because I wanted to introduce newcomers into experimental electronic music making, for obvious availability reasons, the instrument had to be mouse-driven software. All this led to the development of FMOL 1.0, a freeware stand-alone program with real-time interactive synthesis and World-Wide Web capabilities. Written in C++ for the ''Wintel'' platform, the 1998 version requires at least a Pentium 100 with any DirectX-compatible multimedia sound card.

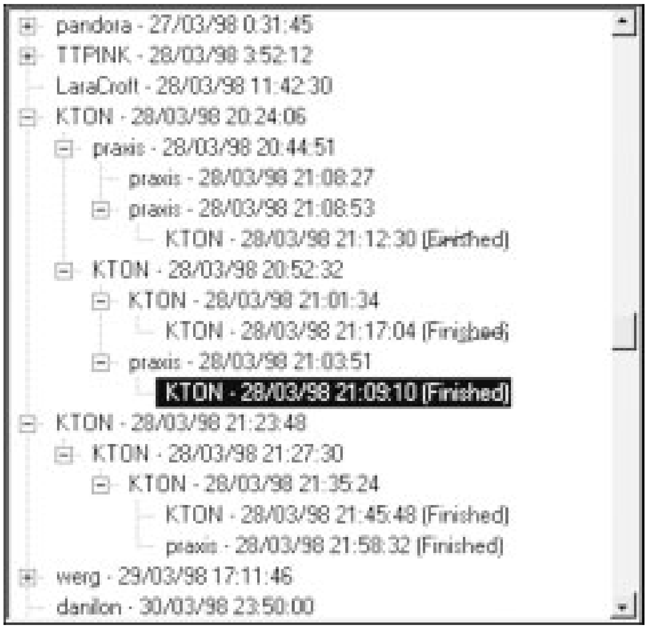
### FMOL Architecture

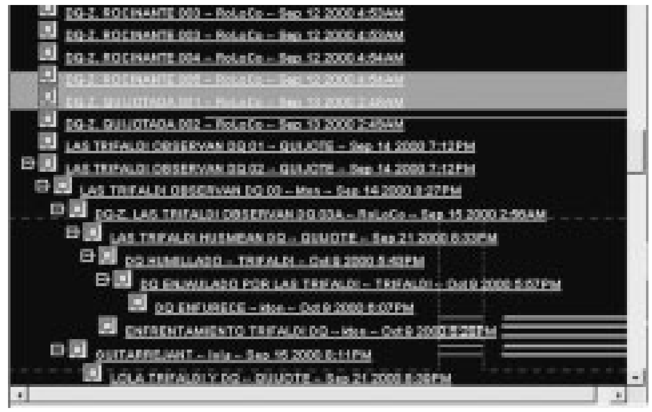
FMOL 1.0 (1998) consists of a hypertext transfer protocol (HTTP) component, a real-time synthesis engine, two GUIs (*Bamboo* and *Medusa*) that communicate with this engine, and two orchestra configuration components (one for each of these interfaces). FMOL 2.0 (2000) eliminates the HTTP component and communicates directly with the http browser application. This makes the evolution of the database observable from any computer without needing to have FMOL installed (Wüst and Jordà 2001).

### Collective Composition

Apart from the aforementioned structural difference, versions 1.0 and 2.0 share the same collective composition paradigm. They do not implement any structures for real-time concurrent ''net-jamming.'' (This feature is included in FMOL 3.0, which will be downloadable in 2003.) Compositions are stored

Figure 1. Tree-like representation of pieces in the database.



in a relational database as small proprietary score files, and no audio is transmitted between the server and the FMOL client. For several authors to collaborate on the same piece, they must store their respective contributions on the FMOL database server before any other author can continue working on them. Each participant is allowed to start new compositions from scratch as well as overdub or process any of the pieces stored in the database.

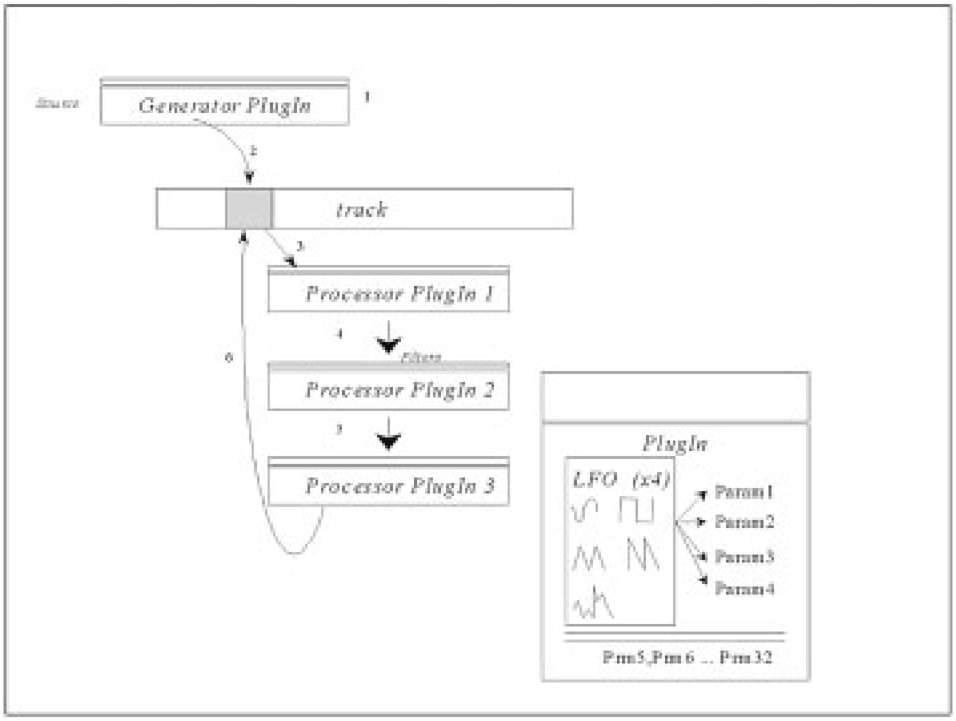
The relational database's main entity is the compositions table, which has a recursive relationship to itself, allowing for a tree-like representation of the pieces, as shown in Figure 1. A user can pick up any existing composition, listen to it, work on it, and save a new version in the database. This new version will be automatically stored as a child node of the one the user picked (Wüst and Jordà 2001). This permits composers to modify and enlarge already existing pieces an endless number of times, while keeping the integrity of the original pieces.

The collaborative paradigm is based on a vertical-multitrack model (as opposed to a horizontal approach, which would allow the pasting of sonic fragments one after the other). Besides, FMOL's synthesizer architecture not only allows users to add new sound layers to previous compositions, but also to apply further processing to any of the previous tracks, modulating or distorting what other composers did in unpredictable manners. That way, a musical idea brought by one composer can grow and evolve in many different directions unexpected by its original creator.

## The Synthesis Engine Architecture

When I was designing the synthesis kernel, performance and creative possibilities were more important than the implementation of state-of-the-art synthesis and signal-processing techniques. The engine supports eight stereo real-time synthesized audio tracks or channels. Each track uses an inde-

Figure 2. The structure of one track.




pendent audio buffer and is constituted of four sound modules (''plug-ins'' in FMOL terminology): a sound *generator* (sine, square, Karplus-Strong, sample player, etc.) and three chained *processors* (filters; reverbs; delays; resonators; frequency, amplitude, or ring modulators; etc.), which can be chosen by each composer from more than 100 different synthesis algorithms or variations. In FMOL 3.0, most of the available generator plug-ins also embed an additional *high frequency oscillator* (HFO) for optional frequency modulation.

Each plug-in (generator or processor) includes up to four control parameters, and one of them can be modulated by a *low frequency oscillator* (LFO), in which frequency, amplitude, and shape (sinusoidal, square, triangular, saw tooth, or random) can be interactively modified. This makes a total of 32 available LFOs (8 tracks × 4 plug-ins per track × 1 LFO per plug-in) that can be interactively controlled from the GUI. Figure 2 illustrates the structure of one track, and Table 1 shows several of the available algorithms with the two principal control parameters of each.

Each of these eight tracks or buffers (which are several seconds long for implementing delay and other feedback algorithms) are partially computed and written several times per second. The frame rate of the system is configurable and varies with the soundcard and the CPU's processing power. A 20 Hz frame rate is attained with no glitches on lower-end machines (e.g., Pentium 100 MHz), while newer machines (1 GHz or better) can surpass 100 Hz (Jordà and Aguilar 1998).

Moreover, the user can also choose to apply a processor algorithm to the generator plug-in of any track, creating a slave track that takes its input from the output of another master track (also selectable by the user). This mechanism can be used

**Table 1. Some of FMOL Synthesis and Processing Algorithms and Their Two Primary Parameters**

| *Algorithm* | *Type* | *Param. 1* | *Param. 2* |
|---|---|---|---|
| BASIC OSCILLATORS | | | |
| Sine Wave | Generator | Pitch | Amplitude |
| Square Wave | Generator | Pitch | Amplitude |
| Sawtooth Wave | Generator | Pitch | Amplitude |
| Triangular Wave | Generator | Pitch | Amplitude |
| Pulse Train | Generator | Pitch | Amplitude |
| White Noise | Generator | NO | Amplitude |
| SAMPLE PLAYERS | | | |
| Sample Player | Generator | Pitch | Amplitude |
| Wavetable Sample Player | Generator | Wave Sound # | Amplitude |
| Scratch Sample Player | Generator | Frequency | Amplitude |
| Granular Sample Player | Generator | | |
| BASIC MODULATORS | | | |
| Ring Modulation | Processor | Pitch | Modul % |
| Amplitude Modulation | Processor | Pitch | Modul % |
| FEEDBACK ALGORITHMS | | | |
| Karplus-Strong | Generator | Pitch | Amplitude |
| Binary Modulation | Generator | Primary Pitch | Amplitude |
| Karplus-Strong Processor | Processor | Pitch | NO |
| Binary Mod. Processor | Processor | Primary Pitch | NO |
| LINEAR DIFFERENCE EQUATION FILTERS | | | |
| High-Low Filter | Processor | Cutoff Freq. | NO |
| Resonant Filter | Processor | Res. Freq. | Q |
| Comb Reverb | Processor | Feedback | Delay |
| Comb Eco | Processor | Feedback | Delay |
| Comb Delay | Processor | NO | Delay |
| Comb Comb | Processor | Gain | Delay |
| OTHERS | | | |
| Pitch Shift | Processor | Pitch Shift | NO |
| Panning | Processor | Angle | NO |
| Line In | Generator | NO | NO |

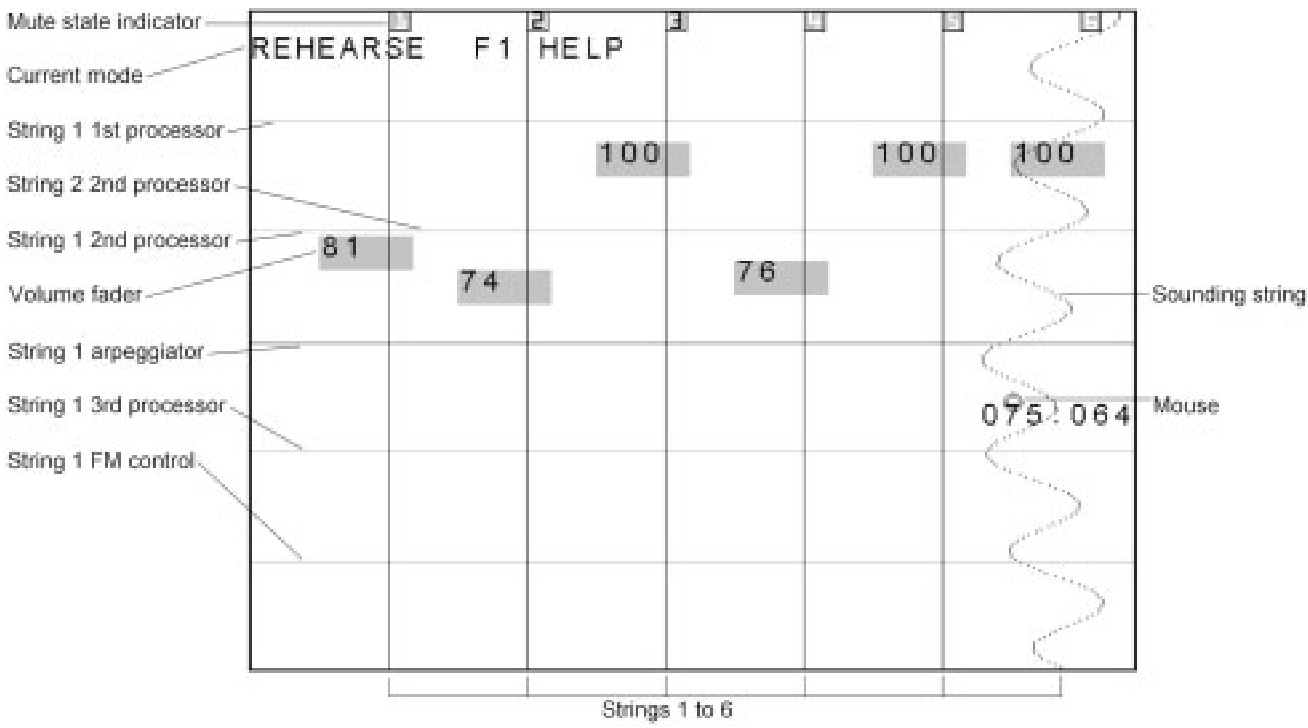
with complete flexibility and allows for rich and intricate sound processing and modulation configurations. It also eases the ''revamping'' of existing pieces.

## The Bamboo Interface

Bamboo, my favorite FMOL graphical mouse-controlled interface, was developed in parallel with the synthesis engine. (Medusa, the second FMOL GUI, is not presented in this article.) Its design and functionality are so tightly bound to the engine architecture that almost every feature of the synthesizer is reflected in a symbolic and non-technical way in this interface.

As shown in Figure 3, in its rest position, Bamboo looks like a simple grid or lattice, made of six vertical and five horizontal lines. These lines con-

*Figure 3. The Bamboo interface in its "rest" position.*

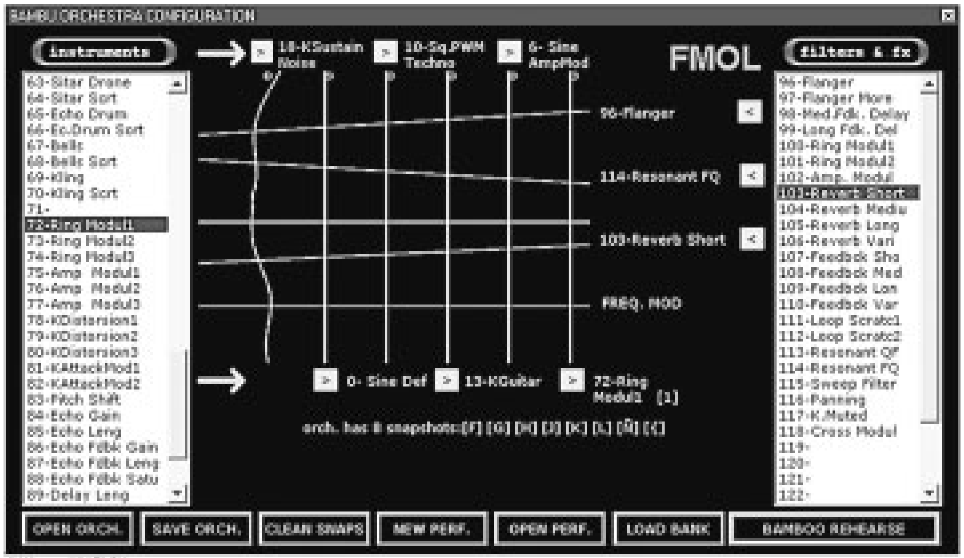



stitute Bamboo's control core. They function both as input devices (controllers) that can be picked and dragged with the mouse, and as output devices that give dynamic visual and "sonic" feedback. The six vertical lines are associated with the synthesis generators (Bamboo uses only six of the eight available synthesis channels), while three (out of the five available) horizontal lines control the synthesis processors. The two additional horizontal lines are used for arpeggiator and frequency modulation control. The selection of which generators and processors are applied to each of these lines is done on the configuration window shown in Figure 4.

**Vertical Lines**

As in a virtual guitar, vertical lines (or strings) can be plucked or fretted with the mouse. However, as mentioned previously, they do not behave like mere input devices: they oscillate like real strings, and, as a result, they dynamically reshape according to the sound they are generating (like an oscilloscope). Once a string is picked with the left mouse button, the string is pulled and divided into two segments (with its vertex following the mouse) onto which the resulting waveform is drawn and permanently refreshed (as shown in the sixth string of Figure 3). Moving the mouse dynamically displaces the vertex and modifies the first two parameters of the generator applied to this string, according to the mouse coordinates. (Normally, the $x$ position is associated with the amplitude and the $y$ with pitch, as shown in Table 1.) Bamboo and the synthesizer engine use the same internal clock, so this input control and feedback mechanism always run in perfect synchronization with the synthesizer at the configured frame rate (20–100 Hz),

*Figure 4. The configuration window.*




giving a time resolution and latency no longer than 10 msec in faster machines. Pressing both mouse buttons affects the third and fourth parameter of the generator. Once the left mouse button is freed, the string releases. To allow polyphony, strings can be sustained. They can also be muted on and off. Sustain and mute states for each string are switched from the computer keyboard and are visually indicated in the Bamboo GUI by means of color changes.

### Horizontal Lines

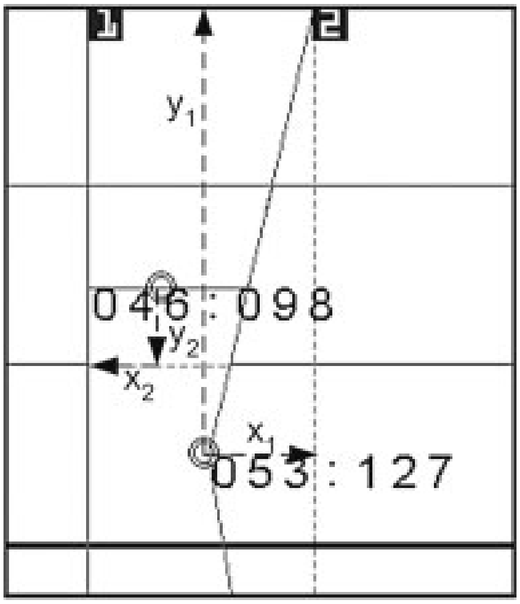
Three of the five horizontal lines control the synthesizer's serial processors. These horizontal lines are divided into six independent segments, each of which is associated with the string/generator situated at its right, as indicated in Figure 3. When a horizontal segment is picked with the mouse, it does not bend like strings do, but it is dragged vertically instead. This vertical position modifies the associated processor's second parameter, while the horizontal position of the mouse over the segment controls the first parameter. Figure 5 illustrates these control mechanisms for both strings and segments.

### Low Frequency Oscillators

When used in combination with several computer keys, picking segments with the mouse activates and controls the LFO of the processor (instead of its first and second parameters). In that case, the horizontal position of the mouse over the segment controls the oscillation frequency, its vertical deviation determines the oscillation amplitude, and the key being pressed selects the shape of the oscillator (sine, square, saw, triangle, or random). Whenever this oscillation mode is activated, visual feedback reappears, and the horizontal segment starts to move up and down with the precise frequency and amplitude of the oscillation.

Two additional segments are available in the Bamboo interface that are not associated with any processor (see Figure 3). The thicker one can be considered an arpeggiator. It controls an LFO directly associated with the pitch of the generator, and it is used for triggering new notes in a cyclical fashion. The bottom one (not yet available in the

Figure 5. Control mechanisms for both strings and segments.




current downloadable version of the software) applies additional frequency modulation to its generator, by controlling the modulation index and the carrier-to-modulator ratio of an additional HFO. This HFO modulator is embedded in every type of generator, except for the ones that do not recognize a pitch parameter (e.g., white noise).

## Visual Feedback and Cues

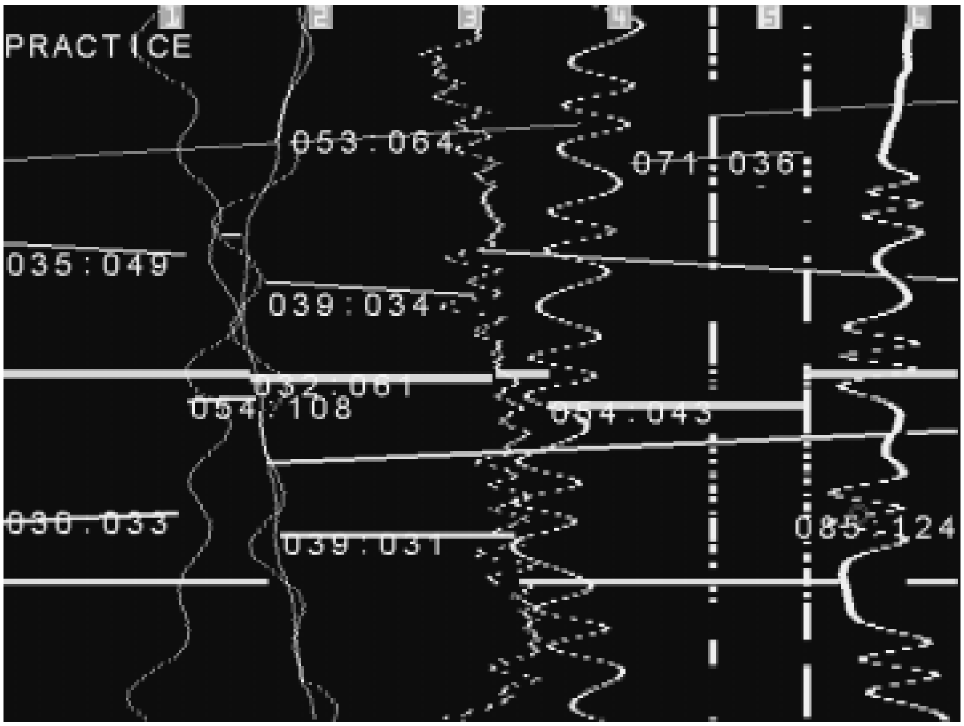
When multiple oscillators or segments are active, the resulting geometric ''dance,'' combined with the six-channel oscilloscope information given by the strings, tightly reflects the temporal activity and intensity of the piece and gives multidimensional cues to the player. Looking at a screenshot shown in Figure 6 (which is taken from a quite dense FMOL fragment), the player can intuitively ''feel'' the loudness, frequency, and timbral content of every channel, the level of different applied effects, and the activity of each of the thirty LFOs. Furthermore, anything in the Bamboo screen can behave simultaneously as an output and as an input.

## Time Control and Reproducibility

FMOL+Bamboo is a musical instrument, and for that reason, it can only be used in real time. It can act as a traditional instrument (which can be used for hours without recording anything), but also as a multitrack recorder and player. Its playback and overdub capabilities are indeed fundamental for listening to compositions downloaded from the database and for further modifying them. However, for conceptual and aesthetic reasons, these capabilities are restricted. FMOL is a ''playable'' instrument, not a compositional environment, and for that reason, users cannot edit performances or trigger prerecorded sequences while improvising. It is therefore difficult to play a fixed sequence of pitches or a precise rhythm, as the interface is good for large-scale or statistical control but poor for detailed and thorough work.

Several additional features are included, however, for those who seek a bit of reproducibility. Users can dynamically record and retrieve all their mouse movements as many times as they want with the help of two keyboard keys (although this material vanishes when they leave the session). Other possibilities include the storage and retrieval of snapshots that hold the state of all the parameters at a given time (and these snapshots can be saved within the score file). Although snapshots are instantaneous (they could be called ''presets'' using the standard synthesizer terminology), they do include the information for the thirty active LFOs (frequency, amplitude, and wave shape), which are in fact responsible for much of the time control and the dynamic evolution of FMOL pieces. FMOL can also be used to generate sonic fragments that can be later employed in other compositional environments, as the system offers a non real-time rendering option that converts recorded compositions into 16-bit stereo wave files.

Figure 6. Screenshot from a dense FMOL fragment.




## How Is FMOL Being Used?

### F@ust Music On-Line and *DQ Don Quijote en Barcelona*

Unlike any other musical system I have designed, FMOL has been used by hundreds of ''Internet composers.'' From January to April 1998, the first FMOL Internet database received more than 1,100 brief pieces by around 100 composers, some of whom connected nightly and spent several hours a week creating music. One of our main goals (to conceive a musical system which could be attractive to both trained and untrained electronic musicians) was fully attained. We know now that several of the participants had no prior contact with experimental electronic music, and a few were even composing or playing for the first time. All of them took it, however, as a rather serious game, and the final quality of the contributions was impressive. After a difficult selection process (only 50 short pieces could be chosen and included on the show's soundtrack), and considering that a great number of interesting compositions had to be left out, we decided some months later to produce a collective CD with a mixture of old and new compositions (Jordà and La Fura dels Baus 1998; Strother 2001).

A web site and a newer version of the software were back online during September 2000 for La Fura´s new show, the opera *DQ Don Quijote en Barcelona*. This opera premiered in October 2000 at the Gran Teatre del Liceu in Barcelona. During one month, more than 600 compositions were submitted, and the selected ones constitute the electroacoustic parts of an otherwise orchestral score.

Listening to all the music that has been produced during this period, we can affirm that, like a traditional instrument, FMOL music has an distinct flavor that results from the peculiar possibilities and limitations of the instrument, but the music is still diverse enough to reflect the composers' individual characters.

### Learning the Instrument

I have given several FMOL workshops for non-musicians. In September 2000, a one-week workshop for visual artists took place in Lisbon. It concluded with several conducted collective improvisations with astonishing results. In October 2001, I gave three one-day workshops in Germany. One was for children and teenagers, while the two others were for adults, some of them musicians. At the end of each day, public concerts were given.

It takes about a half-hour to begin having fun with the instrument, and several hours to acquire some confidence and produce controllable results. However, after five years of playing it, I am still learning it and do often discover hidden features.

### The FMOL Trio

FMOL was originally designed as a cheap and freely available system for online collaborative composition and ''experimental electronic music proselytism.'' To my own surprise, it also turned to be my favorite instrument for live concerts. Since 1999, the FMOL Trio (Cristina Casanova and I on FMOL computers, plus Pelayo F. Arrizabalaga on saxophones, bass clarinet, and turntables) performs free-form improvised electronic music, while two projectors connected to each of the computers give complementary visual feedback. This setup, which enables the audience to watch the music and how it is being constructed, has proven to be a valuable addition to the concerts, giving the public a deeper understanding of the ongoing musical processes and adding new exciting elements to the show (FMOL Trio 2000).

## Future Work and Conclusions

### On-Line Real-Time Jamming

The newer FMOL version, currently under development, will finally support real-time ''jamming.'' The problem of performing a jam session over the Internet has some constraints imposed by the current network technologies (e.g., Bongers 1998; Tanaka and Bongers 2001; Young 2001). The most relevant problem is related to the high network latencies. An event played on a computer placed on one end of an Internet connection will arrive with a perceptible delay to workstations on other ends of the network. This is actually unacceptable for playing most music. Nevertheless, the type of music that is produced with FMOL is more timbral than rhythmic in nature and can therefore better tolerate timing inaccuracies.

In October 2001, several freely improvised duets took place between Barcelona and Dresden and Barcelona and Berlin, using a peer-to-peer prototype of the system. Events played on each computer were locally monitored and synthesized in real time (so that each player could not feel the latency), and sent to the other computer which resynthesized them locally. Consequently, every participant listened to a slightly different version of the collaborative piece, but delays were always less than 100 msec using a standard 56K connection, and the system successfully proved its playability.

The full version (which should be available in 2003) will include a real-time messaging server, probably based on Phil Burk's Transjam protocol (Burk 1998, 2000; Young 2001), which will be accessible through a web-based interface and will provide services such as an FMOL session manager. Active jam sessions will be monitored, and users will be able to create new sessions or join any of the currently open ones provided the maximum number of participants per session has not been reached (Jordà and Barbosa 2001; Wüst and Jordà 2001).

## Conclusions

Apart from the aforementioned new jamming capabilities, the FMOL instrument keeps evolving as we keep playing it. Many minor modifications are related to the synthesis engine capabilities, for example adding new algorithms or improving existing ones. To make FMOL accessible to a broader range of users, it is currently being ported to the Linux platform. This version, which will be open-source, will be released during 2003 as part of the Linux distribution of the AGNULA project ⟨www.agnula.org⟩, where it will form part of the base system together with other contributions from the Music Technology Group at UPF. AGNULA is a European project, started on 1 April 2002, which will last for two years. The project's main task is the development of a reference distribution for the GNU/Linux operating system, completely based on free software and devoted to professional and consumer audio applications and multimedia development.

During these years, the diversity of music that many composers of very different skills have produced with FMOL has proved to us that it is possible to design new instruments which, like the kalimba, can be appealing and interesting to a wide range of musicians without expensive or sophisticated hardware add-ons. From a more personal point of view, it has also been an unexpected surprise that, after years of designing and improvising with interactive music systems, my favorite one has become the simplest one—the one I did not explicitly design for myself, but for everybody.

Improvising with FMOL in different contexts and with different musicians also gives us cues about new features than could be added to the control interface. But we feel that this control complexity cannot be permanently increased, as there are limits to what can be efficiently achieved in real time by means of a mouse and a computer keyboard. Building an external FMOL controller for a faster and more precise multiparametric control is therefore a tempting idea (we are not "mouse fundamentalists"), which needs some serious reflection. Designing a video detection or ultrasound system that would allow musicians to interact on a big Bamboo projection screen, grabbing and moving strings with their hands, was the first idea we had. This would surely add a lot of visual impact to live concerts, but musical control and performance may not necessarily improve with it. For that reason, a hybrid solution, which could also include custom data gloves or another kind of wearable hardware interface, seems to be a more promising approach.

## Acknowledgments

I would especially like to thank my former student Toni Aguilar for his invaluable collaboration in the development of FMOL 1.0, Carlos Padrissa from La Fura dels Baus without whom this project would have never been possible, Cristina Casanova and Pelayo F. Arrizabalaga (members of the FMOL Trio), and all the FMOL Internet composers who during the last four years have contributed their music and their suggestions. I also thank Alvaro Barbosa, Xavier Serra, and Xavier Amatriain from the Music Technology Group at the Pompeu Fabra University for their comments and ideas, and Golo Foellmer from the University of Halle in Berlin for his permanent interest and support of the FMOL project.

FMOL 1.0 was developed with the financial support of the Sociedad General de Autores y Editores (SGAE). The FMOL software can be downloaded from the FMOL home page (www.iua.upf.es/~sergi/FMOL).

## References

Bischoff, J., R. Gold, and J. Horton. 1978. "Music for an Interactive Network of Computers." *Computer Music Journal* 2(3):24–29.

Bongers, B. 1994. ''The Use of Active Tactile and Force Feedback in Timbre Controlling Musical Instruments.'' *Proceedings of the 1994 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 171–174.
Bongers, B. 1998. ''An Interview with Sensorband.'' *Computer Music Journal* 22(1):13–24.
Burk, P. 1998. ''JSyn: A Real-Time Synthesis API for Java.'' *Proceedings of the 1998 International Computer Music Conferenc*e. San Francisco: International Computer Music Association, pp. 252–255.
Burk, P. 2000. ''Jammin' on the Web: A New Client/Server Architecture for Multi-User Musical Performance.'' *Proceedings of the 2000 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 117–120.
Card, S. K., W. K. English, and B. J. Burr. 1978. ''Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT.'' *Ergonomics* 21(8):601–613.
Chadabe, J. 1996. *Electric Sound: The Past and Promise of Electronic Music.* Upper Saddle River, New Jersey: Prentice Hall.
Douglas, S. A., and A. K. Mithal. 1994. ''The Effect of Reducing Homing Time on the Speed of a Finger-Controlled Isometric Pointing Device.'' *Proceedings of the CHI 94 Conference*. New York: ACM Press, pp. 411–416.
FMOL Trio. 2000. *Live at Metronom* (Audio compact disc). Barcelona: Hazard Records 010.
Gagne, C. 1993. ''Interview with Laurie Spiegel.'' In *Soundpieces 2: Interviews With American Composers*. Metuchen, New Jersey: Scarecrow Press, pp. 295–333.
Gillespie, B. 1999. ''Haptic Manipulation.'' In P. Cook, ed. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*. Cambridge, Massachusetts: MIT Press, pp. 247–260.
Goebl, W., and R. Bresin. 2001. ''Are Computer-Controlled Pianos a Reliable Tool in Music Performance Research? Recording and Reproduction Precision of a Yamaha Disklavier Grand Piano.'' *Proceedings of the 2001 Workshop on Current Research Directions in Computer Music*. Barcelona: MOSART, pp. 45–50.
Gresham-Lancaster, S. 1998. ''The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music.'' *Leonardo Music Journal* 8:39–44.
Hunt A., and R. Kirk. 2000. ''Mapping Strategies for Musical Performance.'' In M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music* (CD-ROM). Paris: IRCAM–Centre Pompidou.
Jordà, S. 1991. ''A Real-Time MIDI Composer and Interactive Improviser by Means of Feedback Systems.'' *Proceedings of the 1991 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 463–466.
Jordà, S. 1999. ''Faust Music On Line: An Approach to Real-Time Collective Composition on the Internet.'' *Leonardo Music Journal* 9:5–12.
Jordà, S. 2001. ''Improvising with Computers: A Personal Survey (1989–2001).'' *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 1–8.
Jordà, S., and T. Aguilar. 1998. ''A Graphical and Net Oriented Approach to Interactive Sonic Composition and Real-Time Synthesis for Low Cost Computer Systems.'' *Proceedings of COST G6 Conference on Digital Audio Effects*. Barcelona: Institut Universitari de l'Audiovisual, pp. 207–210.
Jordá, S., and A. Barbosa. 2001. ''Computer Supported Cooperative Music: Overview of Research Work and Projects at the Audiovisual Institute—UPF.'' *Proceedings of the 2001 Workshop on Current Research Directions in Computer Music.* Barcelona: MOSART, pp. 92–96.
Jordà, S., and La Fura dels Baus. 1998. F@ust 3.0—FMOL (Audio compact disc). Madrid: Fundación Autor 02801.
Moore, F. R. 1988. ''The Disfunctions of MIDI.'' *Computer Music Journal* 12(1):19–28.
Norman, D. A. 1990. *The Design of Everyday Things*. New York: Doubleday, pp. 12–13.
Poschardt, U. 1995. *Dj-Culture*. London: Quartet Books Ltd.
Puckette, M. 1988. ''The Patcher.'' *Proceedings of the 1988 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 420–429.
Puckette, M., Zicarelli, D. 1990. *Max: An Interactive Graphical Programming Environment*. Program documentation. Menlo Park, California: Opcode Systems.
Rowe, R. 1992. *Interactive Music Systems: Machine Listening and Composing*. Cambridge, Massachusetts: MIT Press.
Rutledge, J.D., and T. Selker. 1990. ''Force-to-Motion Pointing in Human-Computer-Interaction.'' *Proceedings of the INTERACT'90*. Amsterdam: North-Holland, pp. 701–706.
Shapiro, P. 1999. ''The Primer: Turntablism.'' *The Wire* 179:40–45.

Spiegel, L. 1991. *Unseen Worlds* (Audio compact disc). New York: Aesthetic Engineering 11001–2.
Strother, E. 2001. ''Sergi Jordà: Faust Music Online.'' *Computer Music Journal* 25(1):94–95.
Sundberg, J. 1987. *The Science of the Singing Voice*. Dekalb, Illinois: Northern Illinois University Press.
Tanaka, A., and B. Bongers. 2001. ''Global String, A Musical Instrument for Hybrid Space.'' *Proceedings of the Cast01 Conference on Communication of Art, Science and Technology.* Schloss Birlinghoven, Sankt Augustin, pp. 177–181.
Ulyate, R., and D. Biancardi. 2002. ''The Interactive Dance Club: Avoiding Chaos in a Multi-Participant Environment.'' Computer Music Journal 26(3):40–49.
Ungvary, T., and R. Vertegaal. 2000. ''Designing Musical Cyberinstruments with Body and Soul in Mind.'' *Journal of New Music Research* 29(3):245–255.
Wanderley, M. 2001. *Performer-Instrument Interaction: Applications to Gestural Control of Music*. PhD Thesis. Paris: University Pierre et Marie Curie—Paris VI.
Wanderley, M., and M. Battier, eds. 2000. *Trends in Gestural Control of Music* (CD-ROM). Paris: IRCAM–Centre Pompidou.
Weinberg, G., and S. Gan. 2001.''The Squeezables: Toward an Expressive and Interdependent Multi-Player Musical instrument.'' *Computer Music Journal* 25(2):37–45.
Winkler, T. 1998. *Composing Interactive Music: Techniques and Ideas Using Max.* Cambridge, Massachusetts: MIT Press.
Wüst, O., and S. Jordà. 2001. ''Architectural Overview of a System for Collaborative Music Composition over the Web.'' *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 298–301.
Young, J. P. 2001. ''Using the Web for Live Interactive Music.'' *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 302–305.