



HAL
open science

Optimized real-time simulation of objects for musical synthesis and animated image synthesis

Jean-Loup Florens, Aimé Razafindrakoto, Annie Luciani, Claude Cadoz

► **To cite this version:**

Jean-Loup Florens, Aimé Razafindrakoto, Annie Luciani, Claude Cadoz. Optimized real-time simulation of objects for musical synthesis and animated image synthesis. ICMC'86, Oct 1986, La Haye, Netherlands. pp.65-70. hal-00491718

HAL Id: hal-00491718

<https://hal.science/hal-00491718>

Submitted on 25 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A.C.R.O.E
Association pour la Création et la Recherche sur les Outils d'Expression
L.I.F.I.A.
Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle
B.P 68 - 38402 Saint-Martin d'Hères cedex - FRANCE

Optimized real time simulation of objects for musical synthesis
and animated image synthesis .

Jean-Loup FLORENS -Aime RAZAFINDRAKOTO
Annie LUCIANI - Claude CADOZ

Abstract :

This paper describes four systems for the synthesis of sounds or animated images , based on the simulation of one or several mechanical objects . Those are real time systems which enable an user gestual manipulation of the simulated objects, intended to acoustic or visual signal productions. These realisations prefigure the CORDIS-ANIMA system, a more ambitious system based on the same synthesis method, with composition abilities of instrumental objects and synthesis works . These examples allowed the experimental demonstration of the validity of our approach . They also contributed to solve complex technical problems , raised by the necessary combination of modularity and efficiency of simulation.

1 Introduction

The purpose of our research is the design of a specific tool for musical creation and animated images synthesis , using recent computer techniques : the CORDIS-ANIMA system:

The tool key-function is to simulate objects , so that the user (the artist) is in sensory interaction with them , as in the physical world . It follows from this some basic principles :

- necessity of a digital processing machine , powerful enough to enable real time simulation ;
- specific devices converting physical data from sensory channels into digital data (and vice versa) : the transducers ;
- the "sensible" nature of simulated objects , i.e their property to be excited and perceived through gesture, vision and audition, implies their definition in terms of mechanical models .

Nevertheless the simulation function described above does not realize all the conditions of a creative activity ; the system must also accept a conceptual activity, closely linked with the simulation, to which computer techniques may give new and powerful developments. This conceptual activity addresses different levels , in particular the design of the object or scene and consequently relies on a language for the composition of mechanical objects. The system will enable the elaboration of objects libraries that will be used in a direct way, or as elements for the creation of new objects.

This second function of CORDIS-ANIMA - modularity - implies, in addition, compilation mechanisms to produce an efficient simulation algorithm for each object or scene.

In parallel with CORDIS-ANIMA development, we have set up several non modular simulation systems. The data structure of the simulated object or scene is pre-determined, except for a few quantitative data, allowing elaboration of optimized algorithms and assuming a quite good simulation quality.

Our purpose in these particular simulations was first to prove, through experiments, the validity of the approach, as regards the possibility of simulating relatively complex objects and scenes, rich enough to provide a basic material for an artist . These simulations were presented in several public manifestations, and submitted to many different users .

The other purpose was more technical : the research of optimization methods for the simulation, and their possible extension to a modular simulation system .

II - Realisation of specialized simulations

We focused on typical cases of objects, with a relative acoustic or visual richness. Four examples were studied. The first three are better aimed to sound production : bowed string, plucked strings, maracas. The last one - interacting balls - is meant only for animated image generation.

II.1 Bowed string

It consists in the simulation of a vibrating string, associated to a bow friction mechanism . A specific transducer, substitute of the bow, has been built for this simulation. This artificial bow is composed by a displacement sensor which represents the bow transverse displacement and a pressure sensor which allows the user to control the pressure of the bow on the string (fig.1).

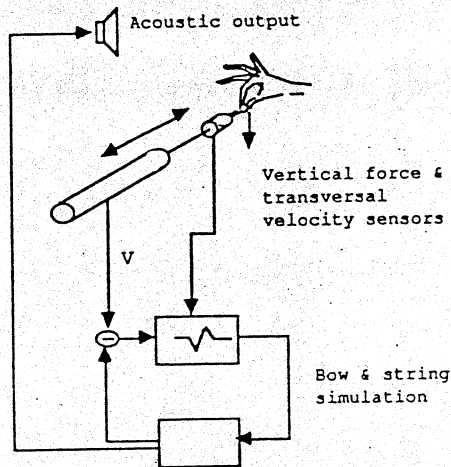


fig 1

The acoustic signal is computed from the value of the force exerted by the string on one of its fixing points ; this represents an important simplification according to the instruments with resonance chamber. The string can be displayed in real time ; it enables to observe the string shape evolution which is in complete coherence with Helmholtz theory .

The string model is simplified insofar as there is only one possible deformation direction, compared with at least four in the case of a real string : 2 transverse displacements, torsion and longitudinal deformation .

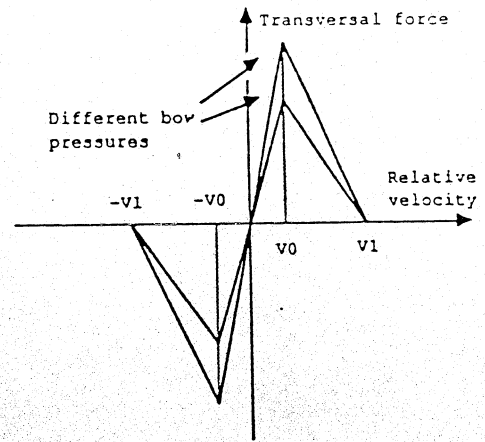


fig 2

The friction model is characterized by a non linear relationship between the sliding speed of the bow and the transversal force exerted on the string ; this function is described (fig. 2). It includes negative-slope segments which are necessary for the conservation of oscillation . The effect of a variation of the bow pressure on the string could be modelled by a variation of F_{max} along the axis of forces on this characteristic function. The slope of this function at the origin cannot exceed a limit value, which would imply the instability of the most damped modes. This restrains the validity of the simulation to "reasonable" bow pressures . This limit also prevents from reaching a "solid" friction which could be realized by setting the values of both thresholds v_0 and v_0' to zero.

II.2 Plucked strings

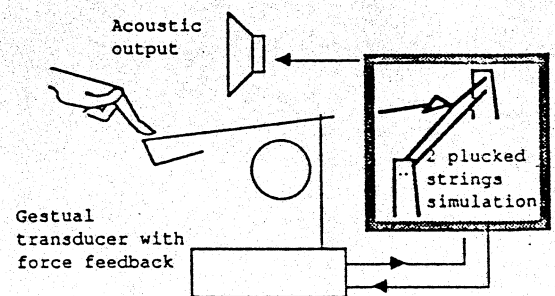


fig 3

The string models are similar to the former example. Two strings are simulated ; the operator manipulates them with a simulated plectrum , which plucks one string then the other, according to its current position (fig. 3). The user manipulates this plectrum through a specific transducer with mechanical feedback ; this feedback represents the reaction of strings to the gestual action. Another transducer, a force sensor, with no feedback effect, is used during the simulation to modify the tension of strings, and consequently the pitch of sounds. We could reproduce some characteristic effects of double strings instruments. The distance between the strings and difference from strings pitch can be independently adjusted. According to the values of the parameters, the user can perceive whether two distinct strings, or for low values, a single string having a phasing effect and a tactile granulation effect. In this case, both effects work toward the perception of a complex single object (mandoline strings effect).

II-3 - Maracas

A sort of maracas is simulated by a rectangular box containing several particles. These particles are punctual masses and can move along one axis, parallel to one of the box surfaces. This box is a resonant cavity excited by the impact of particles on its surfaces. The operator shakes the box with the retroactive transducer ; he feels its inertia , and the hit of particles on the surfaces. An image of the box and particles is also displayed in real time (fig.4) .

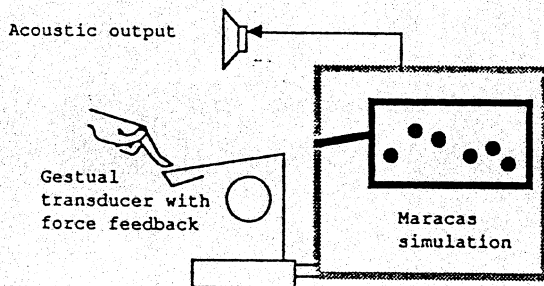


fig 4

The interaction model consists in a double elastic obstacle with no damping. Particles motions have undamped limit oscillations which are not perceptible due to their low amplitude.

The vibration of the cavity is produced by the association of 5 oscillators, representing its resonance modes. These oscillators are excited by the interaction forces between masses and box. The reaction of the vibrating structure on masses is considered as insignificant .

II.4 About sound-producing objects simulation

In these three cases, the vibrating structure of the object could be simulated separately from the other object components, with the same part of program. This method can be used more generally , in a relatively optimal form , for most of vibrating structures (cf fig. 5).

What we call a vibrating structure is the set of points of a sound-producing object that can oscillate at acoustical frequencies. Most vibrating structures can be described with a linear differential equation. It enables to the elaboration of an equivalent model which consists of elementary and independent oscillators and where each oscillator represents one resonance mode. The structure of such a model is quite simple and does not depend on the vibrating object. The corresponding algorithm is also simple and easy to implement on an array processor .

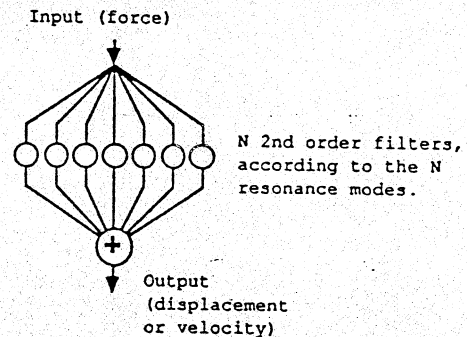


fig 5

This method is not only adapted to discretized models of vibrating structures ; continuous structures, like strings, possess an infinite number of resonance modes, but only low frequency modes may be considered ; the error in the simulation of a continuous model consists in the suppression of high frequency modes, which are generally not relevant above a particular value. The error inherent to the regular discretization of the continuous model would be more perceptible .

II.5. Interacting balls

This experiment is only dedicated to image production ; the scene does not include any sound-producing object.

As compared to the restricted number of objects in the former cases, this example investigates the simulation of scenes composed by a great number of elementary interacting objects.

Several balls are inclosed in a rectangular box where they can move freely. The box also contains a circular goblet which is manipulated with the retroactive transducer. These objects can collide ; the goblet can contain several balls; the operator feels the reactions of balls on the goblet (weight, impacts...) through the retroactive transducer .

We notice that in such a scene , a very simple gestual action can produce many different and complex auto-evolutions .

A ball is described as a punctual mass (in a 2D space), connected to a circular envelope determining its visual characteristics and the conditions of its interactions with other objects.

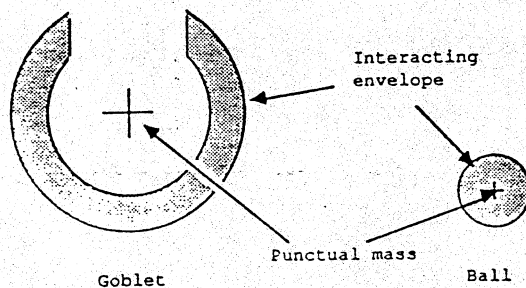


fig 6

The goblet is also described as a punctual mass, but have a more complex envelope (cf fig.6) .

In this example, an optimized module was elaborated to simulate a large number of material points (approximately 80 at 25 Hz) and their interactions.

This module may be used in a more general system to build more complex objects such as plastic solids ; in such cases indeed (for instance balls), the interactions are determined by space exclusion rules, conditioned themselves by the objects envelope and each object can be modelled as a set of points, connected together with attractive links .

Optimizing techniques for this module rely on simplifications in non linear processings according to a primary model : all the non linear functions - i.e. interaction between balls, ball-goblet interaction, distance calculation - are integrated in a single function. It follows from this a general interaction function described on fig. 7.

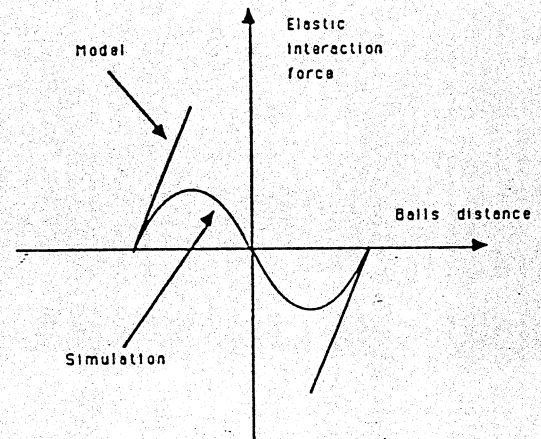


fig 7

III - Optimization methods - Application to CORDIS-ANIMA system :

Compared with other synthesis techniques, simulation is more demanding in digital processing capacity . This explains why we try to take advantage of the specific features of simulated objects, even in quite simple cases . But the method we used enables to define some general principles that will be developed in the future .

III.1. Methods

III.1.1 Model optimization

Each model has its intrinsic complexity ; the computation power necessary for model simulation depends on this complexity which can be measured, according to some possible criteria : the bandwidth, the number of masses, the nature of interactions.

The first step in optimization is related to the choice of the model, or more exactly, to the simplifications that must be applied to a primary model ; in the maracas example, both mechanical functions of the box (resonant cavity and containing balls) were separated, each one being simplified in its own way : the resonant cavity reacts to the impact of different masses as if there were only one point of impact ; the box is considered as one passive material point, with an elastic envelope with no mass, which may interact with other material points .

III.1.2 Algorithms optimization

This is another optimization level .

A first method consists in reducing the system of continuous equations representing the model to a simpler continuous form, and then resolving this one with a numerical integration method. Another method consists in sampling the system of continuous equations and then reducing them to obtain a sampled optimized model, as a set of finite difference equations.

In each example mentioned, a sampled model of the object or scene is built from sampled models of its components. The resulting algorithm may consist in the mere juxtaposition of the algorithms of components. But in most cases, it can be simplified by elimination of some variables or processings. Moreover, interesting approximations can be sometimes made, considering the limits in the requested precision.

III.1.3 Implementation optimization

The last optimization level lies in the implementation of the simulation algorithm in the array processor . This machine (Floating Point System: AP 120B) demands vectorized algorithmic structures, i.e many similar processings that can be led in parallel . In most cases, we took advantage of the multiplicity of some parts of the model, but we also tried to group similar processing elements : in the maracas case, the computing of masses motions and the computing of vibratory modes of the box are both realized in the same vectorized unit. This procedure can save approximately 20% on the maximum frequency .

III. 2. Application to a modular system

For each optimization level mentioned above, we shall study the possible extensions of the methods used to the case of a modular system .

III.2.1 Model generation

There is no general optimization criteria at this level, and automatic processings on models for their optimization can not be realised for the moment .

Such criteria may appear in an experimental phase . Some modeling tools such as object description languages or formal processings may then become useful. This type of problem depends probably more on the know-how of the object designer and each case getting its own solution .

III.2.2 Algorithm generation

The description of the object to be simulated can be given to the simulator in two hierarchical forms :
- A set of finite difference equations (what we call sampled model) , defining the mechanical function of the object .

- A combination of more elementary objects or mechanical functions, for which a description was formerly given to the simulator ; this mode of description relies on an object specification language .

A correct algorithm can be deduced from this sampled model by mere duplication (this was the case in former CORDIS-ANIMA versions), if the set of equations must be computable. This implies that variables at step n depend only on variables at former steps.

In the general case, we can thus distinguish three types of formal processings to determine the algorithm :

- Compilation, elaborating a sampled model (i.e a set of differential equations) from an analytic description ; this means the juxtaposition of elementary objects equations, and identification of variables which support mechanical relationships between components .

- Sampled model reduction : this first form is not necessarily optimal, for example several equations may be dependent, as a result of the combination of similar objects .

- Formal calculation of some equations, to reach the explicit form needed to determine the algorithm .

Introducing some rules in objects specification - for example, exclusive rules in the elements assembly - may avoid formal processings of second and third types . This method is also used in former CORDIS-ANIMA versions. The algorithm of the resulting object is the simple composition of the algorithms of its components .

III.2.3. Simulation program generation

Automatic optimized code generation to an array processor or any simulation-oriented architecture is a very complex problem in overall cases.

** This problem can be partly escaped when the algorithm can come down to a set of blocks, for which the optimized code was written before. Let's take the example (cf 2-2) of an algorithm which is the direct combination of more elementary algorithms. Each elementary algorithm may then be pre-optimized, i.e. a simulation program is formerly written in an efficient form for each "combinable" object or mechanical function.

This method is powerful enough, and avoids most complex formal processings. Nevertheless its efficiency will be improve (for a given object) when this object can be made of a smaller number of pre-optimized elements. It is then interesting to build the richest possible set of basic objects and mechanical functions (including complex objects). Two functions of this type were realized in simulations described above (II 4 - II 5).

** optimization must be comprehensive to be really efficient. In this way, the former method - algorithm decomposition in pre-optimized blocks - will be more powerful, if optimization methods are designed in taking into account implementation criteria rather than only the data structure of the model (see III-1-3).

In the simulations described above, we introduced one single representation mode for the algorithmic level and machine code level, what we call a dependance graph representation. The dependance graph representing the algorithm is splitted up into the processor elementary instructions, then reorganized in taking into account parallelism implementation constraints as the instruction timing and the exclusion between some operations. This is a more efficient manner to produce optimized code than the usual method where algorithm analysis and processor instructions scheduling are performed successively and thus independantly. In the last case, the algorithm organization is fixed before the implementation specifications, decreasing optimization possibilities.

An important formalization work is still to be done, to take advantage of existing automatic code generation methods, but also of more intuitive solutions developped in the cases mentioned above and which take into account some general characteristics of our problem.

IV- Conclusion :

We tried to demonstrate that the specialized simulations we described are a step in the definition of a complete and modular simulation system. These realisations contributed to raise several problems about the conception of this system, and to bring out some solutions.

We proved in particular that the great difficulty in the design of musical creation tool is to find a suitable and adapted compromise solution assuming both modular and efficient properties. This induces two complementary means :

- pre-optimisation, based on the combination of integrated structures.
- formal processing which purpose is to achieve the largest efficiency of simulation, in overall cases.

Bibliography.

1. BENAË A.H.
Fundamentals of Musical Acoustics, New York Oxford University Press 1976.
2. CADOZ C., LUCIANI A., FLORENS J.L.
Responsive input, devices and sound synthesis by simulation of instrumental mechanism : the Cordis system.
Computer Music Journal, Vol 8 No 3 1984
3. CARRIER G.F.
On the non linear vibration problem of elastics strings. Quar. Appl. Math 3(2) 157-165 (1945)
4. LUCIANI A., CADOZ C.
Modelisation et animation gestuelle d'objets. Le système Anima.
Premier colloque image, Biarritz 1984
5. LUCIANI A., CADOZ C.
Un outil informatique de création d'images animées : modèles d'objets, langage, contrôle gestuel temps réel.
Thèse D.I. INP Grenoble 1985
6. RUIZ P.M.
A technique for simulating the vibrations of strings with a digital computer. University of Illinois 1969.
8. SMITH J.O.
Techniques for digital filter design and system identification with application to the violin. Standford University June 83.
9. NICOLAU A.
A hierarchical approach to code generation for horizontally microcoded architectures. Proceedings of the Array conference, New Orleans, Louisiana, april 1985.