On-line Physically based Control of Mass-Interaction Models

Ali Allaoui, Matthieu Evrard, Annie Luciani

On-line physically based control of mass-interaction models

Ali Allaoui[*] Matthieu Evrard[*] Annie Luciani[*]
(*)ICA Laboratory, France
ali.allaoui@imag.fr, matthieu.evrard@imag.fr, annie.luciani@imag.fr

Abstract

Physical modeling allows producing a large set of motions for computer-based animation. The mass-interaction formalism is an interesting way to design physically based models (modularity, simplicity of each module, genericity) and it leads to produce different behaviors: smoke, paste, liquid, crowd... A specific difficulty is to deal with phenomena that are highly non-linear by nature. The point of this article is to present a way of dealing with such phenomena by on-line dynamic changes of the physical parameters, leading to non-linear dynamic behaviors. We describe a novel method inside the mass-interaction formalism that permits to physically control such dynamic parameters changes during the simulation of the physical model. The method consists in introducing in the modeling process, a "controller component" layer, also based on the mass-interaction formalism, that is in charge of the on-line modifications of the concerned physical parameters through a "parameter control module" (PCM). The article then illustrates the method on one example: a usual human gesture, a repeated jump, which requires modifying in a cyclic manner the tonicity of the muscle, during the jumps.

1. Introduction

Physical modeling has become a widely used means to design animation. The mass-interaction system is able to produce a wide variety of different behaviors [5][7][9][12]. Very soon, it has been improved to obtain complex evolving phenomena by introducing an on-line control of some physical parameters, such as the rest length [2][6][8] or the stiffness of springs [10]. The work presented here proposes a control method entirely based on mass-interaction formalism, able to modify on-line the physical parameters of the modeled object by a physically based process, in order to create physical coherent non-linear behaviors. This method is based on three components:

(1) A generic module, called "parameter control module" (PCM), for the on-line control of physical parameters of a 3D model, compatible with the mass-interaction formalism

(2) A 1D mass-interaction network representing the controller disposal that provides scalar to (1)

(3) Interaction between the 3D model and (2) that provides information on the 3D model state to (2).

In section 2, this approach is compared to other implementations of controls of physically based modules. Section 3 deals with the implementation of our method within our user-friendly modeling MIMESIS software for mass-interaction model design [9]. The method is illustrated by an exemplary modeling case, a usual human gesture, a repeated jump, which requires modifying in a cyclic manner the tonicity of the muscle.

2. Global context

Physical modeling is a generative way of producing motion. As in Physics, physical *parameters* are data such as stiffness, viscosity, thresholds in distance and velocity, static and dynamic coefficient of friction, inertia, while positions, velocities, angles, forces, torques, … are inputs or outputs *variables* of the physical model.

When dealing with the control of physically based models, we can distinguish two main directions: (1) one aiming at acting on the variables, for example forces, positions, velocity fields as done in [1] and [14]; (2) a second acting on the physical parameters of the 3D objects of the scene. In this paper, we focus on this second type of control.

The main way in computer graphics for dealing with this type of control is to specify high-level objectives in order to synthesize controllers by means of optimization methods.

Our contribution aims at considering the physical model and the controller as a whole dynamical system, designed by means of the same unified formalism.

Hence, we propose to extend the possibilities of the mass-interaction formalism as described in figure 1 with our controller system in two layers:

(1) The controller itself: it is a one-dimensional mass-interaction system. The controller is in physical interaction with the 3D object. It reacts on line to the dynamical states of the 3D objects in their physical environments.

Proceedings of ENACTIVE/07
4th International Conference on Enactive Interfaces
Grenoble, France, November 19th-22nd, 2007

(2) A set of modules, one by parameter controlled, that modify the physical parameters of the 3D objects from the data provided by the controller.

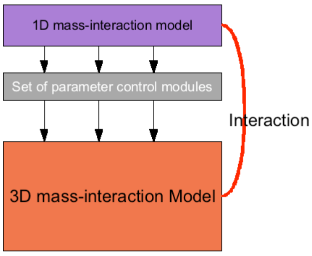

Figure 1: On-line physically based control of physical parameters

So doing, we make possible what can be called non-linear parameters auto-regulations as it is in several complex non-linear phenomena like Van der Pol oscillators for example [13].

Differently than all the previous approaches, our method allows the user to build the controller. By using the physical mass-interaction paradigm, the user designs a whole closed loop dynamical system with a simple and modular language. Actually, one can figure out easily what the system represents, because its parameters (elasticity, viscosity) are meaningful in everyday life. Of course, this simplicity is relative, as it goes more and more complex when the number of modules increases.

3. The parameter control module (PCM) in MIMESIS software

The parameter control module takes place within a user-friendly modeling MIMESIS software [3]. It is based on mass-interaction modeling ([5]). In MIMESIS, the mass-interaction formalism is the core of the creation process at hand. It is therefore able to manage a lot of modules, edit initial conditions and physical parameters, simulate the designed model, import and export the signals representing variables that evolve in time, coat the obtained movements, etc.

Until now, MIMESIS software has support for two categories of modules, the MATs and the LIAs (figure 2, up). A MAT module is a module that takes forces as in input and returns a position. It is a kind of material element which main parameter is its mass (or inertia). A LIA module connects two MATs. It represents the interaction between both MATs. This interaction can be a simple linear damped spring but also a more complex non-linear interaction such as cohesion, plasticity, friction, etc. Physical parameters of LIAs are stiffness, viscosity, thresholds on distance, static and dynamic coefficient of friction, etc..

This work introduces a new module called PCM (parameter control module). It allows modifying a physical parameter of a MAT or a LIA by the position of a MAT, i.e. the value of the parameter will depend on the position of this MAT. Unlike MAT and LIA, that are bidirectional modules (figure 5), it is an oriented module: it takes a position from the MAT and computes the parameter value through a mathematical function. As a parameter is a scalar value, the simplest choice for the MAT that modifies this parameter through the PCM module is a 1D MAT, i.e. a MAT that produces a 1D scalar data. (see figure 2, down).

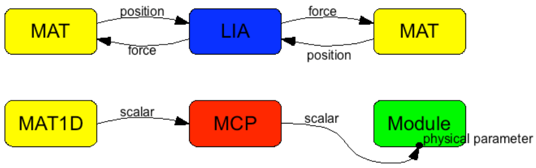

Figure 2: Modules in MIMESIS software

4. Modeling an elementary human motor gesture: jumps

4.1. The phenomenon and the model

A repeated jump requires to modify in a cyclic manner the tonicity of the muscle, during the jumps. It is a basic case of non-linear dynamic behavior. In this paper, we are interested in introducing this type of modification. We used a simple passive mass-interaction model of a jump previously introduced by Hsieh in [4]. In this model, the muscles of the leg are approximated as a unique spring connecting the foot and the pelvis (figure 3) and a jump is triggered by the initial velocity of the pelvis. In order to render a jumper who will jump repeatedly, i.e. to start a new jump as soon as he has landed, it is necessary to make this model active, for example by changing its inner dynamic state during the simulation. It may be done by changing the physical parameters according to the state variables.

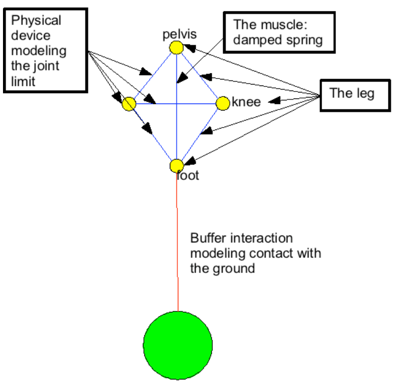

Figure 3: 3D model of a jumping leg

4.2. The controller

A way to do this is to modulate the rest length of the muscle by means of a mass-interaction controller described in figure 4.

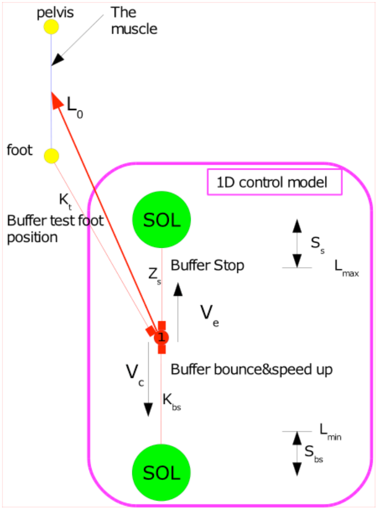

Figure 4: Repeated jump model and its control

The position of the mass n°1 (see figure 4) controls the muscle rest length through a PCM module (the arrow pointing to the muscle).

The jumper performs his jumps in a repetition of 4-steps cycle :

1. The jumper prepares a jump by contracting his muscle. In the controller view, mass n°1 moves from L_{max} to L_{min}.

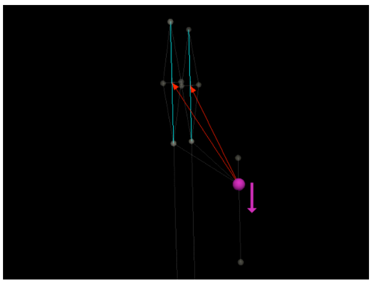
Figure 5: Step 1

2. He extends his muscle quickly. Mass n°1 bounces on the fixed mass located at L_{min}, and speeds up thanks to a well-tuned elastic buffer.

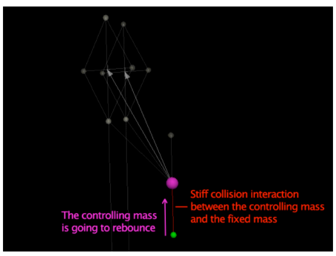

Figure 6: Step 2

3. During the take-off, when the mass n°1 reaches L_{max}, it is stopped by a highly viscous buffer, leading to maintain the legs in extension during the jump.

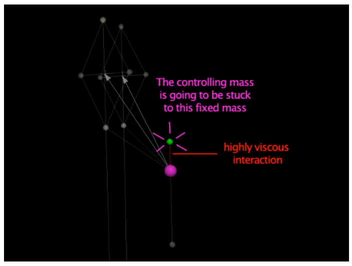

Figure 7: Step 3

4. During the landing, when the jumper hits the ground, he also hits mass n°1, so the jumper is triggering a new jump (step 1).

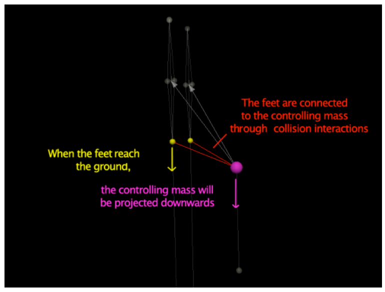

Figure 8: Step 4

4.3. Tuning the controller model

In our model, the state variables of the current jump (cycle n-1) that are relevant in the performance of next jump (cycle n) are: the height of the jump H_j, the velocity of contraction V_c, and the velocity of extension V_e.

Moreover, these 3 parameters are influencing each other, from a cycle n-1 to the next cycle n:

$H_j(n)=f(V_e(n),V_c(n))$

$V_c(n)=g(H_j(n-1))$

$V_e(n)=h(V_c(n))$

The functions f, g, h can be tuned implicitly, i.e. without having an explicit expression of them, by tuning the three parameters of the buffer interactions (buffer stop, buffer bounce and speed up, buffer test foot position, in figure 4). First, the stop buffer is just a viscous interaction, tuned to stop the agitator the closest to L_{max}. Then, the buffer "test foot position" is the softest possible to detach the agitator from the influence of the stop buffer, so that V_c is the smallest possible. Finally, the buffer "bounce & speed up" takes advantage of a discretization artifact, that can be usefully mastered thanks to [11] which makes the correlation between stiffness values and the gain or loss of energy according to the used discretization scheme.

5. Conclusion

Generating highly non-linear movements is always a tricky task in modeling. However, to obtain a certain degree of complexity, non-linearities must be taken into account. Indeed, a lot of natural well-known phenomena are themselves deeply non-linear, as those in which there are changes in the physical states of the matter (for example, fluid - solid transitions), modulation of physical parameters according to the states of the system (for example, modulation of the elasticity according to the elongation of the spring), or in active systems such as human locomotion. After the huge developments in computer animation, we assumed that it is a way to reach a new step in the complexity and the expressiveness of synthetic motions.

Two questions have been risen in the work presented here: first, how to give access to the user in the designing of such type of control? And second how implement it? We proposed to introduce a specific module for controlling physical parameters in a mass-interaction generic system and we set up methods to design controllers that are also physically based and consequently in closed dynamic interaction with the objects to be controlled. So doing, we initiated the process of implementation of inmost closed-loop control systems, as it is necessary in complex non-linear systems. This method corresponds really to a shift in the way of thinking about the control process: from explicit control to implicit dynamic control. We experimented this approach in two complementary examples: the modulation of elasticity and the modification of the tonicity of a simple muscle-like object as it occurs in an active system.

The modules and models developed in this work have been implemented for the first time, within a user-friendly interface, which allow end-users to design mass-interaction models. Consequently, they are available for any user who wants to understand the method and so doing, to become able to use it to design his own physically based models including dynamic parameters control. We assume that it could be the best "practical" way to introduce them freely in the domain of complex dynamic modeling and to constitute their own expertise by themselves.

References

[1] R. Barzel, A. H. Barr. A modeling system based on dynamic constraints. In Proceedings of SIGGRAPH 1988, ACM SIGGRAPH, 1988, pp. 179-188.

[2] P. Boulenguez, A. Luciani. A Physical Particle Model of Morphogenesis. WSCG'03 Plzen (Czech Republic).

[3] M. Evrard, A. Luciani, N. Castagné. MIMESIS : Interactive Interface for Mass-Interaction Modeling, In Proceedings of CASA 2006, Geneva, July 2006, Nadia Magnenat-Thalmann & al. editors. pp177-186.

[4] C. Hsieh, A. Luciani. Generating Dance Verbs and Assisting Computer Choreography. *Proceedings of MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pp 774—782. 2005.

[5] A. Luciani, S. Jimenez, J.-L. Florens, C. Cadoz, O. Raoult. Computational physics: a modeler simulator for animated physical objects. *Proceedings of the European Computer Graphics Conference and Exhibition*. Eurographics'91, pp 425-436, Vienna, Austria, Elsevier Ed, Sep. 1991.

[6] G.S.P Miller. The motion dynamics of snakes and worms. *Computer Graphics*, vol. 22, SIGGRAPH'88, 169-178, Aug 1988.

[7] G.S.P. Miller, A. Pearce. Globular dynamics : a connected particle system for animating viscous fluids. *Computers & Graphics*, vol. 23(3):169-178, Elsevier, 1989.

[8] Sodaplay java applet - http://www.sodaplay.com

[9] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Proceedings of Graphics Interface 1995*,Graphic Interface'95, pp 147-154, Quebec City, Canada, (May 1995).

[10] D. Terzopoulos, J. Platt, K. Fleisher. Heating and melting deformable models : from goop to glop. *Graphics Interface'89,* 219-226, 1989.

[11] F. Thil. Simulations de collisions et application à la synthèse d'objets sonores et percussifs. Rapport de stage de DEA, ACROE-ICA Grenoble 2003.

[12] D. Tonnesen. Modeling liquids and solids using thermal particles. *Graphics Interface'91,* 255-262, 1991.

[13] B. van der Pol, J. van der Mark. Frequency demultiplication. *Nature*, 120, 363-364, (1927)

[14] A. Witkin, M. Kass. Spacetime constraints. *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, p.159-168, June 1988