EVOLUTIONARY IDENTIFICATION OF ACTIVE PARTICLE SYSTEMS

Bogdan Stanciulescu, Jean Louchet

Ecole Nationale Supérieure de Techniques Avancées
32 boulevard Victor
75739 Paris cedex15, France
e-mail: stanciul@ensta.fr
http://www.ensta.fr/~stanciul

ABSTRACT

This paper presents how it is possible to introduce active motricity into particle-bond systems used in applications such as image animation. We chose to add into some neural network capabilities over the classical approach, in order to obtain a system able to model a larger class of behaviour. Therefore a new type of binary bond enriched with a neural-based command ability is proposed and tested in this paper. This “active” bond acts like a controlled muscle in order to produce motricity.
An Evolutionary Strategy is used to optimise the particle-bond system parameters through evolving parameter sets. We tested our method both on artificially generated data and on data collected from real-life motion.
Results and comparisons between our method and other approaches show the advantage of using active particle-bond systems for image animation applications.

Keywords: computer animation, computer graphics, physically based motion modelling, particle-based modelling, evolutionary strategies, motion analysis, neural networks, neural controllers

1 INTRODUCTION

1.1 PARTICLE MODELS

Particle models have been introduced in the Seventies in the fields of Simulation and Image Synthesis, in order to simulate a variety of physical phenomena, such as explosions, flames, smokes [Desbrun96].
In the same framework, image analysis and synthesis give us a set of examples of particle-based models. Some of them are used in simulations of elastic or deformable objects; this is the case of bond-particle networks [Luciani91].
Pattern Recognition is also a field of applications in which the particle-based models could be applied. Man-machine interaction and their applications in virtual reality is another field involving spring-mass models; particularly the GROPE project of University of North Carolina, in which a system of gestural control using a force feedback is used, offers the possibility of real time modelling in molecular chemistry.
The DEXTER model, developed by E. Cohen at ArSciMed, is an example of a system allowing to develop various models fitted on real physical problems.

1.2 THE PHYSICAL MODEL

We have chosen here the particle-bond paradigm as physical model [Louchet96a]. The particles are characterized through their masses. The bonds are couples of generalised springs and dampers. They model interactions between two or more particles.
The generalised springs can be unary, binary, ternary springs depending on the number of particles involved.

- Unary springs are interactions between one particle and the global environment, for instance external fields, gravitational, electrical etc. In this case the force value may depend on the particle position.
- Binary springs [Luciani91] are the ordinary springs and they describe interactions between two particles.
- Ternary springs introduced by [Louchet96a] are interactions between three particles, the force value in this case is proportional with the cosine of the angle formed. They model flexion.

Dampers are introduced in a similar manner:

- Unary dampers are dissipative interactions between a particle and its environment, the corresponding force depends on particle velocity.

- Binary dampers model two-particle dissipative interactions, the force is a function of their relative velocity.

1.3 BUILDING A REALISTIC MODEL

The most important difficulty is to find the optimal values of all the parameters involved in the model in order to get a given behaviour.
One approach has been developed by Nicolas Szilas [Szilas95]. He established a correspondence between mass-spring systems and recurrent neural networks, applying the theory of formal neural networks in physical modelling. He used the supervised recurrent neural networks to determine the values for the mass-spring model. First he tested the Back Propagation Through Time algorithm (BPTT) and then, Real Time Recurrent Learning (RTRL). He also used mechanical constraints to design new algorithms (coupling, half-coupling and coupling-decoupling algorithms). He concluded that RTRL algorithms encounter many difficulties in succeeding, especially parameters as speed, position may become infinite. The BPTT has better results and also a good speed of convergence. In contrast with RTRL the model here is stable and the values taken by the parameters do not diverge.
In the second approach developed by one of the authors of this paper, an Evolutionary Strategy (ES) is used for parameter optimisation [Louchet96a]. For this purpose a cost function has been assigned to each model proposed and it measures a mathematical distance between the initial trajectory and the one produced by the model.
The word “trajectory” means the entire collection of positions and velocities for all system’s particles and all time steps.
The advantage of using ES is that they are not imposing any condition over the cost function, such as continuity, continuity for the derivatives etc. Once a structure is given the algorithm searches for the values which minimise the cost function. Among the differences between this method and conventional ES, a short-term cost function has been introduced, which measures the quadratic difference between the real point on the trajectory and the one predicted using the speed and the position at the previous time step [Louchet96a].
Another innovation was the introduction of the local cost function, which splits the cost function into a sum of local terms associated with each particle. This function directly acts in the mutation and crossover computation. It is based on the assumption that at a given moment the phase of a particle only depends on the phases of their neighbours. A consequence is that the number of generations needed by the algorithm to converge does not depend on the number of variables or object’s complexity but it depends on his average connectivity [Louchet96a].
These two methods have some limits; the first one is that we do not know how to identify particles’ masses (which were a priori considered as input data), and in both cases mentioned before the absence of motricity restrains their applicability to a limited class of trajectories.

2 ACTIVE ELEMENTS: MUSCLES

The connection between two different particles was modelled until now as a association of spring-damper.
This approach inspired directly from physics, gives very good results in modelling physical phenomena but it fails when we try to identify the motion of a living creature, especially the animal walk. In the ENSTA team, Teisserenc [Teisserenc97] studied the horse walk and concluded that the springs are not sufficient when one tries to model the horse walk, so he decided to introduce the dampers, in order to make the motion more realistic. He obtained better results using this kind of connections, but only when he took into consideration some particular parts such as legs, head, bend.
The further step in particle-spring modelling was the introduction of generalised springs [Delnondedieu93]. Unfortunately all these elements mentioned above act as passive devices, incapable to inject energy into the system. Therefore, the necessity of introducing active control elements, in order to have all the elements involved in a natural animal movement, has been raised as the next step.
These active elements able to generate motion (“muscles”), will also have to fulfil the role of co-ordinating the motion of different autonomous body parts.

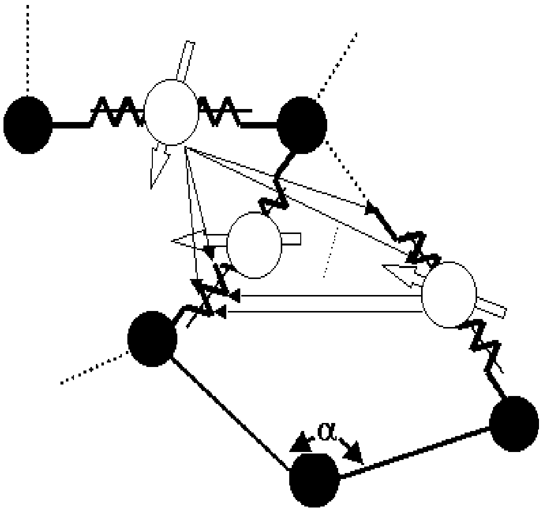
3 MUSCLE MODELLING

3.1 MUSCLE REPRESENTATION

Looking for a way to implement the muscles in designing our system brought us several ideas worth to be outlined here.
- The first idea was to develop an analogy of a laser medium, as a collection of individual oscillators, spatially distributed together with the possibility to excite them with a moving wave. The second choice could be the study of coupled oscillators, in a more general framework, where damping and forcing over the system are both present. This approach led us to a system of coupled differential equations, which requires a numerical resolution. Unfortunately the computation required

is high, due to the great number of variables (force, spring parameters).

- Another approach could be the control of the damping coefficients by a command system i.e. a control over the energetic flux in the mass-spring network. This effect is very similar with the transistor effect, where the gate potential controls the current between the other two electrodes. This idea requires an additional command system able to decide to increase or decrease the damping coefficients.



The mechanical-neural bond architecture
Figure 1

- The last solution envisaged is to use neural networks as a command system for all the springs in our model. We proposed to associate with every spring a neuron, which has the task to observe the evolution of all the other springs, which are components in the mechanical system. The process of observation consists in monitoring for each spring the relative speed and position (Fig. 1). The output of the neuron will be a function that depends on all the phases in the system. If the output function is activated, the activation is the exact amount of excitability which will be applied over its associated spring.

This last proposal is the one adopted in this paper, due to its interesting interconnection properties.
A neuron input is a weighted sum of all relative speeds and positions for the other springs, except the one it is associated with.
Each neuron has its own threshold, and the difference between the input and its threshold represents the variable of its output function. We chose the hyperbolic tangent as output function.
A suitable way for neural implementation is to modify the structure of binary springs, adding the extra parameters required by a neuron. The new binary spring will be the old one plus all the connexions associated with the other springs, and the neuron's threshold.
In the following we will be calling this new kind of binary bonds "mechanical-neural" bonds. Thus, a neural control system is created in parallel with our mechanical system, in a way similar to what can be found in all basic living creatures.

3.2 SYSTEM PARAMETERS IDENTIFICATION

In this research we have chosen to extend the model identification method which had already been developed in our team for simpler particle-bond systems [Louchet96b], to the mechanical-neural bond architecture presented in Section 3.1 .

3.2.1 EVOLUTIONARY STRATEGY

The identification method is based on an Evolutionary Strategy (ES). The reason of our choice was based on ES efficiency in parameter optimisation, especially in problems out of the reach of deterministic algorithms. In our case, the system of coupled equations describing the system's behaviour cannot be solved directly. Another advantage is that the same optimisation technique is used for finding all types of parameters included in a mechanical-neural binary spring.
Each model is an individual in our ES, and is represented by a chromosome, which contains all the values for springs, dampers and neural networks' weights and thresholds.
We start with a population of models and we let them evolve using genetic operators.
The cost function represents an accurate mathematical expression of our problem. It associates a numerical value for each model, a smaller value corresponds to a better data reconstruction.
This cost function measures a generalised distance in the state space, between the trajectory predicted by the model being evaluated, and the real given trajectory.
The model which has the lowest cost function, is the one from which emerges a trajectory as close as possible to the trajectory wanted. That means that its mechanical system and also its neural control system are the best fitted simulating our data.
We introduced a local cost function [Louchet96a] which exploits the topological properties of our system: the position of a particle at a given time only depends on the recent history of its neighbourhood. The first neighbours of a particle are defined as the particles sharing a bond in common with it: the force received by this particle at time t (and therefore its position at the next time step t+1) depends on the speeds of its neighbours at the time step t. This led us to split the global-cost function into the sum of each particle's

contribution. We call these contributions, attached to individual particles, the "local cost functions". The local cost functions play a key role in the identification algorithm. The local cost function corresponding to the extremities of one bond will be more relevant than the global cost function to evaluate the estimation quality of this bond by the model.
The algorithm designed here is self-adaptive in the meaning that the mutation probabilities are controlled by input data. To this end, every individual contains in addition of his parameters, the variances for all mutations applied over the genes. We are calling these parameters mutabilities. They also obey to all the genetic operators in the ES, in order to guide the population to a global optimum. These parameters are adjusted automatically all the time during the run, giving a dynamically evolving trade-off between parameter exploration and numerical precision [Bäck95]. It is worth to remark that the computation complexity here is not affected, due to the fact that in our cost function only model parameters are involved, not their mutabilities. As the cost-function dominates in terms of computing-time, mutabilities do not introduce any adverse effect on computation time.

3.2.2 EVOLUTIONARY OPERATORS

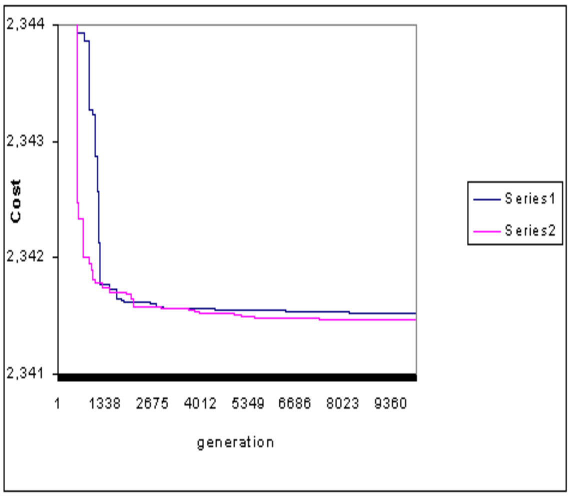
The *mutation* operator is implemented here in four concurrent variants, as in [Louchet96b] but extended in order to include the neural parameters into the evolution process. The mutability parameters control the variance of mutations, while the mutability parameters themselves are subject to mutations too.
The *crossover* operator is a multi-parent crossover. For each occurrence of a bond type, all the particles which are at an extremity of this bond are examined. A parent is chosen randomly, with a bias in the favour of locally good ones (relative to this particle). The new values of parameters for the gene corresponding to the current bond type, are calculated as the mean values of all the corresponding parameters taken from all the parents chosen for this bond type.
The selection process used here is a random choice on the population, biased in favour of the globally best individuals (with the lowest global cost function values), and implemented in such a way that the population count remains constant. In order to ensure the cost function of the best individual does not increase, the 10% best individuals in the population are protected against selection ('elitist selection').

4 EXPERIMENTAL RESULTS

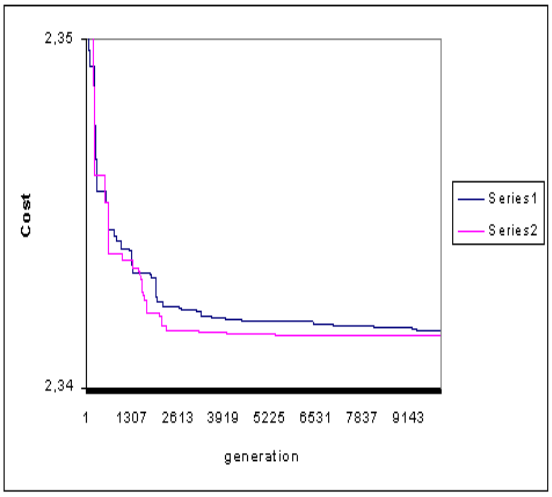
The model proposed in this paper has been tested with two trajectories.

4.1 AN ARTIFICIALLY GENERATED TRAJECTORY

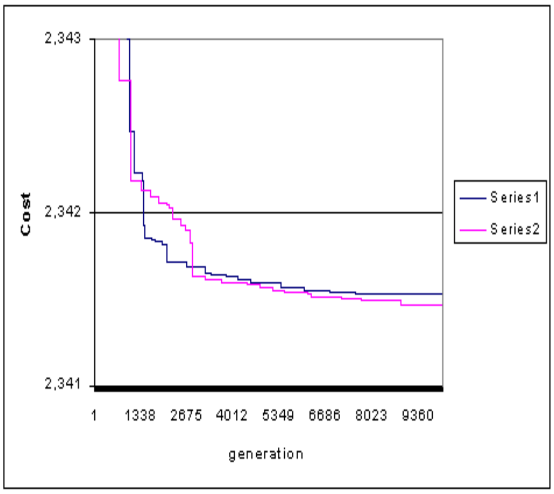
The first type is represented by a trajectory artificially generated using a standard particle-bond system, in which the bonds are viscoelastic but do not contain any neural controller.
The test consists in comparing the evolution of both methods, the standard one based only on particle-bond structure without motricity, and the new one, based on generalized bonds.
Figure 2 shows the the difference between the two evolutions.
The standard one converges better during the first generations, but it is overpassed after few thousands generations by the new one. The following two examples depicted in the figure 3 and figure 4 are a similar comparison of the same two algorithms. We observe the results are almost the same. The population size used here is 1000 individuals.
Analysing the weights found for the neurons, we conclude that adding the neural system over our mechanical structure is an acceptable way to introduce spring and damper control. The neural network weights have small values around zero, which indicates that all neurons are active. A great value for one or more weights means a neuron pulled into the saturation mode (output $\pm$ 1), due to its transfer function. The mechanical effect is that the neuron forces constantly a connection between two particles, pumping the same amount of force at each time step.
The opposite case where one or more weights are zero means that a connection between two springs is missing, that they have an independent evolution from one another, or in terms of mechanics that they are not mutually coupled.



Standard (Series 1) and new (Series 2) algorithms
(10,000 generations, 1000 individuals)
Figure 2

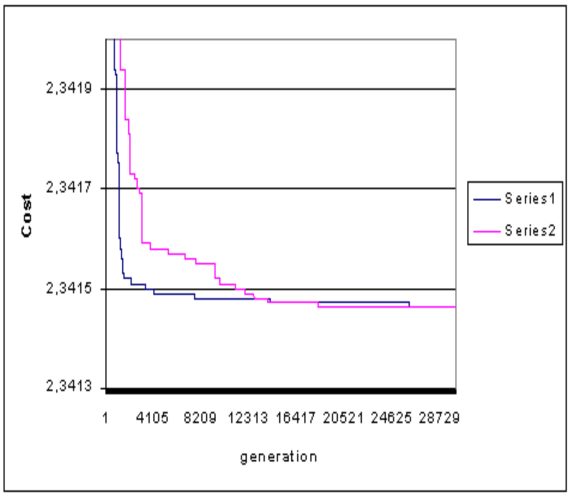


Standard (Series 1) and new (Series 2) algorithms
(10,000 generations, 1000 individuals)
Figure 3



Standard (Series 1) and new (Series 2) algorithms
(10,000 generations, 1000 individuals)
Figure 4

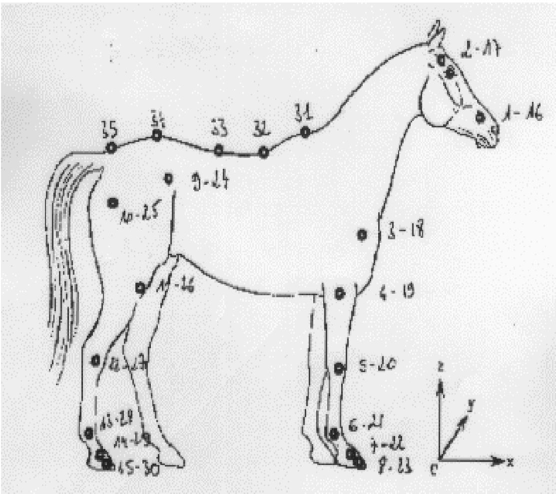
In figure 5 we did the same comparison using the same population size over 30.000 generations. The same tendency is maintained: the standard method is better at the start, but the new method is definitely better on the long term, though the difference is small. This can be explained by the fact that our test trajectory was built upon a standard particle-bond system, therefore it is not surprising that the cost function for both methods converge to zero.



Standard (Series 1) and new (Series 2) algorithms
(30,000 generations, 1000 individuals)
Figure 5

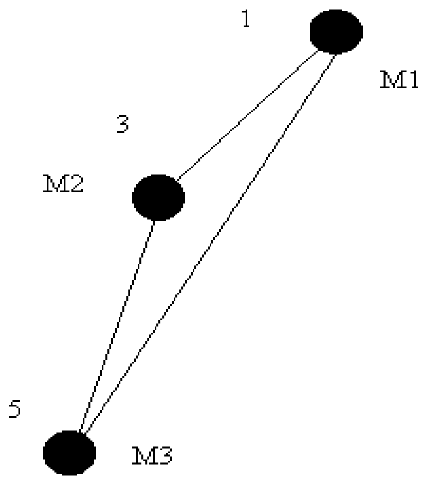
4.2 A REAL-LIFE TRAJECTORY

The second type of trajectory we chose to test our method is taken from real life, and represents the trajectory of three points located on a horse at the trot (Fig. 6) [(1)].



The points used for recording the trajectory, located on a horse body
Figure 6

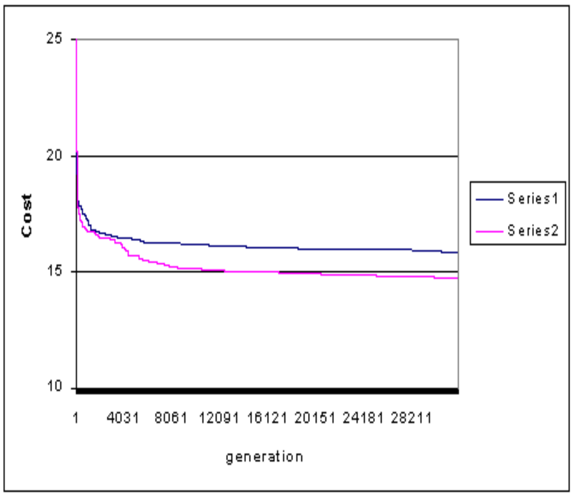
Our trajectory data were obtained placing thirty five markers on a horse body, motion was recorded using a 50 Hz sampling frequency [Teisserenc97]. There are 53 frames recorded containing our data. The first point is located at the mouth, the second point is located on the upper extremity of one leg, and the last point represents a foot (Fig. 7).



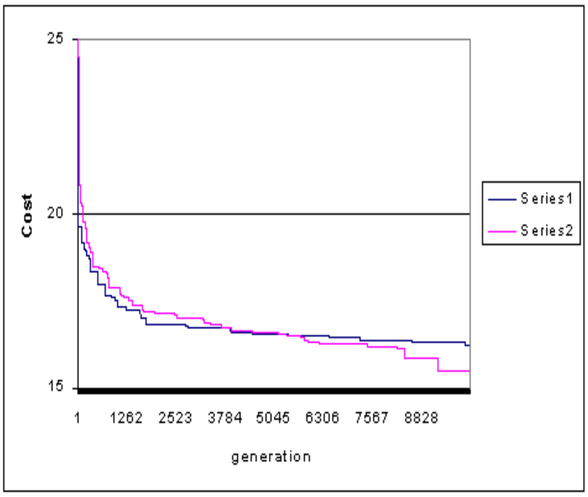
The points chosen for algorithm comparison
Figure 7

[1] Horse data kindly provided by Ecole Veterinaire, Maisons Alfort (France) and Ecole Nationale d'Equitation, Saumur (France).

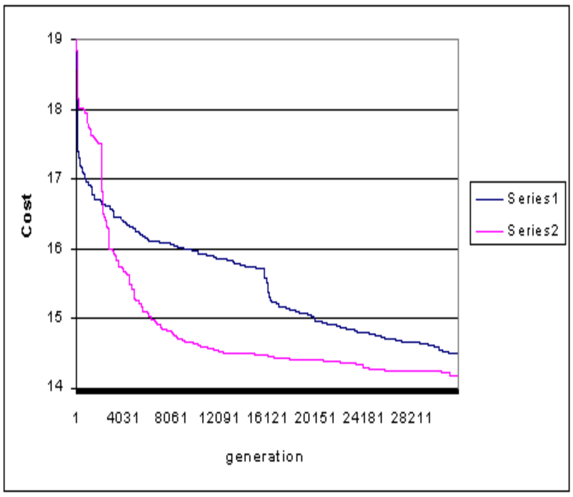
We used the ten forts frames of the trajectory. We modelled the two-particle interaction using binary spring and dampers. Over this structure the same control structure, composed of feed-forward neurons is built.
Therefore we identified the parameters in both cases (mechanical and mechanical-neural systems). Here we can find three examples of algorithm comparison, made on different populations and with a different number of generations. In Figure 8 both algorithms use a population size of 500 individuals and 50.000 generations.
In this case the aim is to better approximate all the trajectories physically implementable. The first test was to verify the self-consistency of our algorithms by testing a 100% solvable problem. As one can notice here the difference between the two evolutions is much greater in this case of a real-life trajectory than in our first test. This shows that the new model is performing better than the standard one. Rather obviously the additional neural "muscle" system enriches the system's modelling capabilities.



Standard (Series 1) and new (Series 2) algorithms (50,000 generations, 500 individuals)
Figure 8



Standard (Series 1) and new (Series 2) algorithms
(10,000 generations, 200 individuals)
Figure 9



Standard (Series 1) and new (Series 2) algorithms
(50,000 generations, 100 individuals)
Figure 10

Examples in Figs. 8-10 show that the same trend is maintained for both methods.

5 CONCLUSION

The aim of this paper was to introduce motricity for particle-spring systems in order to be able to model better a larger class of physical trajectories.
We chose a feed-forward neural network-based approach, which can be seen as a nervous system added on a mechanical structure, performing a task of co-ordinating the movement of its components. From a mechanical point of view, we may consider the neural connections between two springs in the system as mutual coupling between two oscillators. The amount of coupling and their co-operation act as a control system over the spring-particle network. Two steps were considered for the validation of our approach. To test the self-consistency of the algorithm we chose a trajectory from the class of trajectories which can be fully identified by standard particle-bond systems. The results obtained were encouraging. The new approach, based on mechanical-neural bonds, performs better than the old one, though converging to the same value. Furthermore we tested our new approach over a real-life trajectory. Less surprisingly, in this case the difference between the two methods was greater, in favour of the new one.

In this paper we have shown how it is possible to overcome one of the limitations of conventional particle-bond systems in modelling complex motion as encountered in living beings, by introducing active mechanical-neural bonds. We showed that the evolutionary identification method recently developed for conventional bonds could be extended successfully to mechanical-neural bonds and allow the use of real-life kinematic data to train the model.
However, it is still necessary to know the values of masses and the structure of the particle-bond system to enable the algorithm to identify the bonds' mechanical and neural parameters. Current research in our team aims at automatic identification of masses and structures, using more elaborate evolutionary schemes.

ACKNOWLEDGEMENTS
Thanks to J.L. Florens and the INPG/ACROE team for their helpful advice and support.

REFERENCES

[Bäck95] Thomas Bäck, Hans-Paul Schwefel: Evolution Strategies I: Variants and their computational implementation, Genetic Algorithms in Engineering and Computer Science, John Wiley & Sons, 1995.

[Delnondedieu93] Yves Delnondedieu, Annie Luciani, Claude Cadoz: Physical Elementary Component for Modelling The Sensory-Motricity: The Primary Muscle, 4th Eurographics Workshop on Animation and Simulation, pp.193-207, Barcelona September 1993

[Desbrun96] Mathieu Desbrun, Marie-Paule Gascuel-Cani Smoothed Particles: A New Approach for Animated Highly Deformable Bodies, 7th Eurographics Workshop on Animation and Simulation, Poitiers, France, September 1996.

[House92] D.H. House, D.E. Breen, P.H. Getto: On the Dynamic Simulation of Physically-Based Particle-System Models, Proceedings of EuroGraphics'92 Workshop on Animation and Simulation, Cambridge England, 5-6 September 1992.

[Louchet96a] Jean Louchet: Self-Adaptive Evolution to identify Structure from Motion, ImageCom 1996, Bordeaux, France

[Louchet96b] J. Louchet, M. Boccara, D. Crochemore, X. Provot, Building new tools for synthetic image animation using evolutionary techniques, Artificial Evolution 95, Brest, September 95 Springer Verlag, 1996

[Luciani91] A. Luciani, S. Jimenez, C. Cadoz, J.L. Florens, O. Raoult: Computational Physics: A Modeler-Simulator for Animated Physical Objects, EuroGraphics '91 conference, Elsevier Science.

[Reeves83] W.T.Reeves. Particle Systems- A Technique for Modelling a Class of Fuzzy Objects, Computer Graphics (Siggraph) vol17, no.3, pp 359-376, June 1983

[Salford] Project of National Advanced Robotics Research Centre, Salford University, UK
http://www.salford.ac.uk/

[Szilas95] Nicolas Szilas, Eric Ronco: Action for learning in non-symbolic systems, Proc. of the European Conference on Cognitive Science, pp.55-63,Saint Malo, France, 4-7 April 1995.

[Teisserenc97] Patrick Teisserenc: Identification de modèles d'animation de la locomotion équine, ENSTA Research Report, 1997 (in french); see also ENSTA/AMI activity report
http://www.ensta.fr/~louchet/AMI/