Expressive Gesture Animation Based on Non Parametric Learning of Sensory-Motor Models

Sylvie Gibet, Pierre-François Marteau
Valoria, Université de Bretagne Sud, Campus de Tohannic, 56000 Vannes, France
{sylvie.gibet, pierre-francois.marteau }@univ-ubs.fr

Abstract

This paper presents an efficient method of learning motion control for autonomous animated characters. The method uses a non parametric learning approach which identifies non linear mappings between sensory signals and motor control. The learning phase is handled through a General Regression Neural Network model simulated by using near neighbors search algorithms (kd-tree). The resulting adaptive model (ASMM) is suitable for the expressive animation of an anthropomorphic hand-arm system involved in reaching or tracking tasks.

1. Introduction

Animating virtual human with biological realism is of fundamental concern in computer graphics and artificial life. Over the past few years, an increasing interest has emerged to new and speculative application in intelligent control of autonomous virtual actors. One of the great challenges of designing biologically-inspired models is that the resulting movements might exhibit properties inherent to human movements. Among these properties, the most significant are: plasticity, genericity and anticipation. Plasticity can be described as the capability of systems to rapidly adapt to changes of the environment. Generic control explains how similar control principles can be used for various kinds of sensory motor systems or neuro-anatomical variability among individuals. For instance, similar movement pattern can be attained through the use of different muscle combinations. Anticipation refers to the predictive capability of the Central Nervous System. Considering that the brain has only a few milliseconds to react to a given situation and to select the appropriate sensors to achieve skilled tasks, prediction becomes a necessity. The design of biomimetic systems with respects to some of these properties might lead to a greater autonomy and a better adaptation to variations of the environment.

In this paper we present a biologically motivated learning control method applied to the control of complex articulated systems. These systems can be seen as sensory-motor systems which exhibit non linear mappings between sensory signals and motor commands, that are in many respects basic components involved in the control of complex multi-articulated chains. Our approach is motivated by considerations commonly accepted in the neuro-physiology community which lead to the conclusion that these mappings are learned by biological organisms rather than pre-programmed. For the design of artificial life systems, the biological plausibility of the involved mechanisms is not really considered as an issue. Nevertheless, we aim at designing virtual actors that have adaptive, predictive and generic capabilities. Moreover, the behavior of our virtual actors is goal driven rather than just being purely reactive. Their behavior is adaptive, in the sense that it can be automatically adjusted to suit unanticipated variations in their unstructured environments.

Another argument in favor of learning is that, when modeling complex artificial systems with many degrees of freedom, it may be difficult to express the analytical differential equations that describe the dynamics or the kinematics of the system. Furthermore, these equations lead to numerical simulations with high computational cost and the design of control strategies for such systems is not a simple task. In this context, learning part of the control strategy from the observation of the system behavior is an appealing and efficient approach. In this paper we present a non parametric learning method for the control of sensory motor systems. The learning algorithm is used to animate with expressivity an anthropomorphic hand-arm system in multi-point reaching tasks.

2. Background

Previous work in learning motion control can be divided in two main classes, depending on the motivations: the first class concerns work which provides new insights into motor control. This kind of work may improve the understanding via simulation of hypothetical strategies that the Central Nervous System uses to control limb movements. The second class concerns the design of artificial systems that mimic biological behaviors.

Within the first class of work, numerous approaches integrating learning mechanisms have been developed to control sensorimotor systems. Among them, several significant contributions highlight two main approaches: those which are looking for an *a priori* analogy with biological systems (identification of functions of the cerebellum) [1-3], and the others which are looking for an *a posteriori* analogy with biological systems [4-5].

In the second class of work, the problem of learning motion control is encompassed by the highly developed field of adaptive control. The two fields have similar goals but differ in scope and execution. A traditional adaptive controller tends to adjust a small number of parameters to achieve optimal performance. In contrast, a typical "intelligent" motion controller tends to output the control signals directly from a neural network, or similar device. A much larger number of parameters are adjusted by learning. The last approach is essentially developed in the field of autonomous robots, which must learn about their unstructured environments without continuous human guidance. A lot of research has been done on using neural network control [6] (see section 5.1).

Motion learning has recently raised interest in the computer animation community for autonomous virtual characters acting in varying environments. In the field of dynamic locomotion control, Van de Panne proposed an original neural network approach - Sensor-Actuator Networks [7] and the graphs of postures [8] - to optimize the parameters of controllers whose structure operates as rhythmic generators. Terzopoulos and al. developed a realistic aquatic locomotion method in which the learning of the optimal swimming mode is realized though an optimization process operating on Fourier coefficients [9]. These authors also demonstrate how neural network emulators can learn physical models by observing the dynamic state transitions produced by such models in action [10]. Evolutionist theories, essentially based on the use of genetic algorithms have also raised a great interest for learning animated motion [11]. To arrive at biomimetic autonomous agents situated in realistic virtual worlds, much effort should be carried out in learning both controllers and direct systems.

Our approach tries to understand the sensorimotor transformations involved in motion control and to learn the different mappings characterizing the system to control and the controller itself.

3. Controlling Sensory-Motor Systems

Any motor system can be characterized in a state (or phase) space where the state of the system is supposed to be completely determined at any time by a point in this space. Time evolution of the system results in the development of a trajectory in the state space.

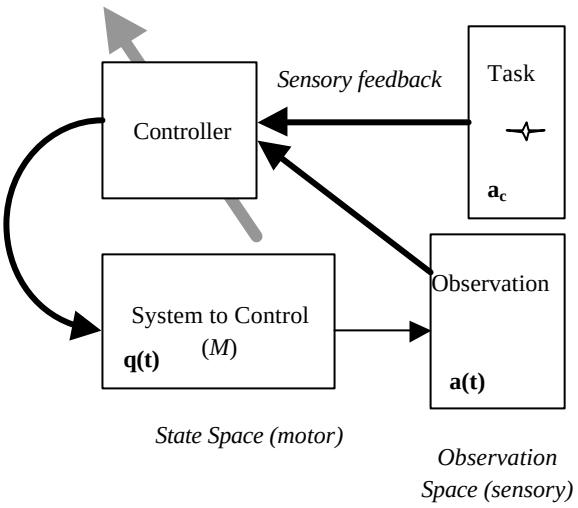

Figure 1. System controlled with sensory feedback

For sensory-motor systems, the state is observed through sensory signals (observable outputs), as shown in figure 1. Thus, the track of the evolution within the state space results in the development of a trajectory within a sensory space. The mapping *M* between sensory outputs and state characterizations is generally highly non linear and projective (the dimensionality of the state space is higher than the dimensionality of the sensory space). When the system is controlled using a sensory feedback, the task affected to the system is specified in a space homogeneous to the sensory space labeled task space in figure 1. The error signal measured between sensory outputs and task inputs is finally used as a feedback to update the state of the system. The projective transforms *M* are generally well-defined functions, with a redundancy in the articulated systems characterized by an excess of degrees of freedom, e.g. the transformation between the input and the output is characterized by a many-to-one transform. Thus, the same sensory outputs may be observed for numerous

different states of the system. Consequently, forward mapping M is well-defined while the inversion of M is an ill-posed problem. Control strategies of Sensory Motor Systems (SMS) require some how either the knowledge of the forward mapping M or the knowledge of its pseudo inverse: M^{-1}. Both are closely related to the structure of the SMS itself.

4. Gradient Sensory Motor Model (GSMM)

Inverse kinematics can be regarded as a nonlinear optimization problem based on the minimization of a scalar potential function defined by :

$$E(q) = \left(M(q) - a_c\right)^T \left(M(q) - a_c\right)$$

where $M(q)$ denotes the forward kinematics map from State space to Observation space and a_c denotes the desired sensory output expressed in the Task space. Such approaches have been used in computer graphics as a numerical iterative approach to solve inverse kinematics IK [12-13].

Another approach has been proposed to solve IK which try to understand the nature of the sensorimotor transformations involved in pointing tasks [14]. This approach has been also used for motion planning [15].

Our GSMM model uses a gradient-based algorithm in a sensorimotor closed-loop transformation which integrates neurophysiological elements. This model has proved to control articulated chains and produce motion that globally respects human motion laws [16]. It has been used in a modular architecture to generate expressive communicative and Sign Language Gestures [17-18] or coordinated juggling motion [19].

In such a model, the update of the state q is computed on the basis of the error E between the current endpoint a and the task specification a_c (goal), according to a gradient descent strategy.

$$\frac{\partial q}{\partial t} = -g(E(a, a_c, t)).\frac{\nabla_q(E(a, a_c, t))}{\left\|\nabla_q(E(a, a_c, t))\right\|}$$

$$= -g(E(a, a_c, t)).\frac{(\frac{\partial M}{\partial q}).(M(q) - a_c)}{\left\|(\frac{\partial M}{\partial q}).(M(q) - a_c)\right\|} \quad (1)$$

$(\frac{\partial M}{\partial q})$ is the Jacobian matrix of the operator M, g is a gain function and $\tilde{N}_q$ the gradient operator.

In order to ensure the stability of the system and to generate damped behaviors, a nonlinear function g and a second order filter have been introduced. The nonlinear function g has a sigmoid shape: the gain of this function increases significantly when the error between the observable position and the reference target position goes towards zero. Stability and asymptotic properties of such a model have already been studied in [16]. For GSMM, the direct mapping M required to compute the gradient of the error E as shown in equation (1). To implement this model, all coefficients of the Jacobian matrix $(\frac{\partial M}{\partial q})$ should be known for all values of the state vector q. These coefficients depend directly on the structure of the articulated chain to be controlled. Furthermore, for any articulated chain, a specific Jacobian matrix should be calculated.

5. Learning Sensory Motor mappings in GSMM

The previous requirement for an analytical knowledge of the sensory-motor mapping restricts the potential use of GSMM to well defined articulated systems. One way to overcome such limitation is to introduce a learning scheme, a functionality that most of biological systems have implemented.

5.1. Non Parametric Learning v.s. Parametric Learning

Two distinct and competing approaches are available when facing the problem of learning non linear transforms (NLT) and in particular non linear mappings involved in multi-joint control systems: parametric learning (PL) and non parametric learning (NPL) (Cf. [20] for a pioneer and detailed synthesis on PL and NPL, and [21] for a more recent review of PL and NPL models with biological relevance arguments regarding internal sensory-motor maps). The fundamental difference between PL and NPL is that PL addresses the learning essentially globally while NPL addresses it much more locally. In other words, PL methods try to learn non linear transforms over their whole domain of validity. This means that if a change in the environment occurs locally, it will potentially affect the learning process every where in the definition domain of the transform. Conversely, NPL learns the properties of the transform in the neighborhood of each point of interest within the definition domain of the transform. Thus, a local update in the learning process does not affect the rest of the learned definition domain. Multi layer Perceptron [22-23] are instances of the PL class with synaptic weights as parameters, while Probabilistic Networks or General regression Neural networks [24-25] are instances of the NPL class.

Biological relevance can be found for the two kinds of approaches [26]. Nevertheless, local characteristics of NPL is undoubtedly a great advantage when addressing incremental learning in variable environments, since the local modification resulting from any change does not affect the overall structure of the non linear transform already learned.

5.2. Learning GSMM maps using General Regression Neural Network(GRNN)

To estimate the normalized gradient of the error, the following map f is defined:

$$d\hat{q} = f(q, da) \qquad (2)$$

where da is the 3D directional vector towards the task a_c specified in the sensory space, q the vector of the state variable. $d\hat{q}$ is the estimated normalized modification within the state space that will minimize the error between a (the mapping of q in the sensory space) and the task specification a_c. The different mappings are illustrated in figure 2.

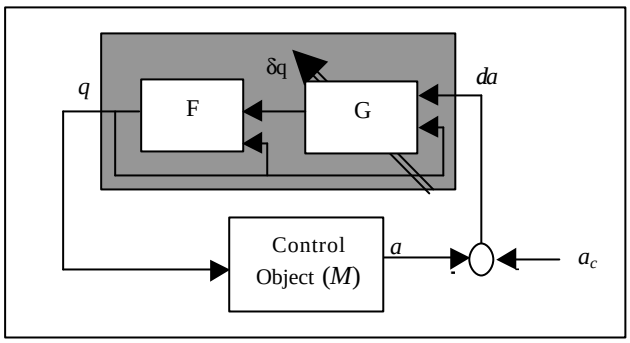

Figure 2. Structure of the ASMM (Adaptive Sensory Motor Model) model.

Following GRNN memory based approach, the calculation of the map f is approximated through a variable gaussian kernel density estimator as explained below:

Given a set of N learning samples, $\{(q_i, dq_i, da_i)\}_{i=1...N}$, the state update $d\hat{q}$ that minimizes the error signal calculated from a current state q and a 3D normalized directional vector da is estimated as the conditionnal expectation of dq given x:

$$d\hat{q} = \frac{\sum_{i=1}^{N} K(x_i, x) dq_i}{C}, \qquad (3)$$

where $x=[q, da]$, $x_i=[q_i, da_i]$, C is a normalizing factor, and K a variable gaussian kernel:

$$K(x_i, x) \approx \exp\left(-\frac{(x - x_i) W (x - x_i)^T}{s} \right) \qquad (4)$$

W is a weighting diagonal matrix used to balance the weighting of sensory information,(da) with motor information (q), s is a parameter that scales the local density in the state space and in the sensory space: if the density is low, s is increased and conversely, if the density is high, s is lowered.

5.3. Naive GRNN learning algorithm

s is selected empirically, since an optimum value cannot be determined from a set of observations.

<u>Initialization</u>: select a small value ε, an integer value N and set i to 0 (e can be a function of N).

1) Select randomly a state vector q, position the multi-joint system according to q, and observe the corresponding sensory outputs a.
2) Select a small normalized change dq, position the multi-joint system according to $(q + dq)$, and observe the change in sensory outputs da.
3) Calculate $d\hat{q}$ using $x=[q, da]^T$ according to equations (3) and (4).
4) If $||d\hat{q} - dq|| > e$, save the association $([q, da], dq)$ as a new learning sample (ξ_i , dq_i), create a corresponding neuron and increment i.
5) If $i<N$, loop in 1), stop otherwise

5.4. Implementation issues

In estimating the expectation of the state update (dq) given x, the computations of distances in the d-dimensional x vectors space are required (d = dimension of the state space + dimension of the sensory space). When summing gaussian kernels (eq. (3) and (4)), only the ξ_i vectors belonging to the neighborhood of ξ are retained. To speed up the computation process, a kd-tree [27] for identifying neighborhoods in logarithmic time with N can be advantageously used. (The kd-tree representation of the stored data leads to reconsider the architecture of GRNN to implement similar neighborhood search).

6. Results

Figure 3. Visualization of the animated character

6.1. The animation system

The 3D models are achieved with OpenGL libraries. A typical representation of such models is shown in figure 3. Graphical 3D output is generated at a rate of 8 frames per second.

6.2. Geometrical models of the Hand-Arm System

The arm is composed of three joints with seven degrees of freedom as shown in figure 4.

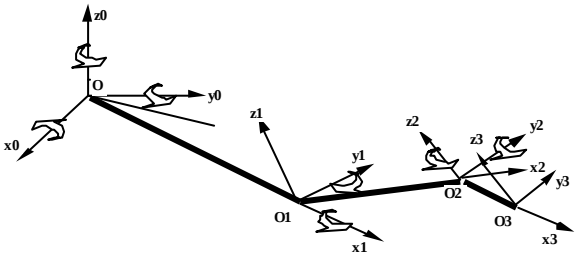

Figure 4. The arm model: the arm is modeled with three joints and seven dof.

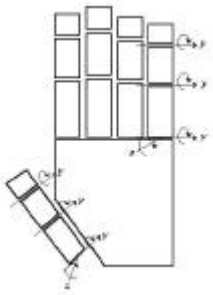

Figure 5. The hand model: each finger is modeled with three joints and four dof.

The arm is composed of three joints with seven degrees of freedom as shown in figure 4. The hand is modeled by a set of five articulated chains, each one having 4 dof [17].

6.3. ASMM experimentation

To experiment the learning capability of the above presented NPL technique, ASMM method has been experimented on the above hand-arm system.

The arm is controlled both by a Gradient-based model (GSMM) and a learning-based model (ASMM) where the gradient descent strategy is estimated using learning of the non linear mapping f characterized in the previous section. The hand is controlled by five direct angular-driven GSMM.

To illustrate the flexibility and the generic characteristics of ASMM, a simple experiment is carried out on a simulated arm system performing multi-points tracking and reaching tasks.

For increasing values of the number of learning samples N, five learning runs are carried out. For each of these runs, two hundred and fifty 3D spatial target positions and initial conditions have been selected randomly. For these 250 conditions, the error rate (number of cases where the arm is not able to reach the target), the average of the time to target (number of iterations of the sensory motor loop) and the average of the residual distance to target when errors occur are calculated.

The experimental settings for this test are the following:

- A target is considered to be reached when the residual distance between the arm end-point and the target is below 0.05.
- The size of s is selected such that at least 40 neighbors can be provided to evaluate dq.
- W is such that $w_{ii} = 1$ if i identifies a state variable in q , $w_{ii} = .005$ if i identifies a 3D coordinate of the arm end-point, $w_{ij} = 0$ otherwise.

The results of this test are reported in figures 6,7 and 8. Clearly, for 60000 learning samples, the map f is apparently well modeled, since the residual error rate is low (about 5%) and very few improvements are gained when increasing N. Furthermore, the average time to target is more or less asymptotically reached for $N=80000$, while the average residual distance to target when an error occurs increases slowly.

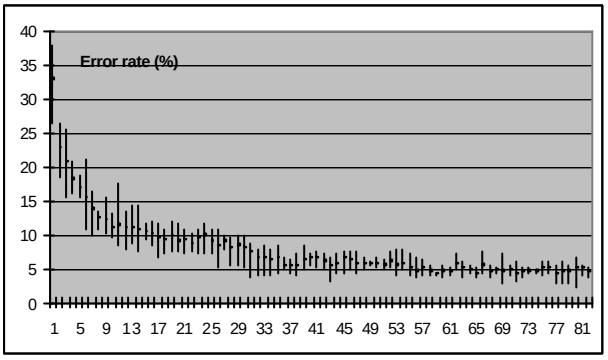

Figure 6. The maximum, minimum, and average error values evaluated on the 5 learning runs are presented for N varying from 1000 samples to 85000 samples.

The first interesting result is that *N* can be chosen quite low for acceptable performances. It is commonly accepted that for estimating a multivariate function with *10* variables (e.g. *7* degrees of freedom and 3D coordinates) using a kernel density estimator requires above *800,000* samples adequately selected [28]. *N=80,000* seems to be sufficient for the considered task. One reasonable explanation is that the sensorimotor loop performs a time average over successive gradient values that compensate small errors due to the coarse estimation. A rough gradient mapping estimation is consequently suitable for the reaching task that is addressed.

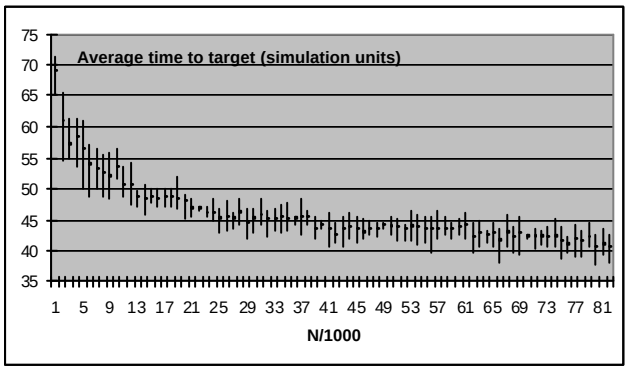

Figure 7. The time average to target as a function of N. The maximum, minimum and average values of the 5 learning runs are shown.

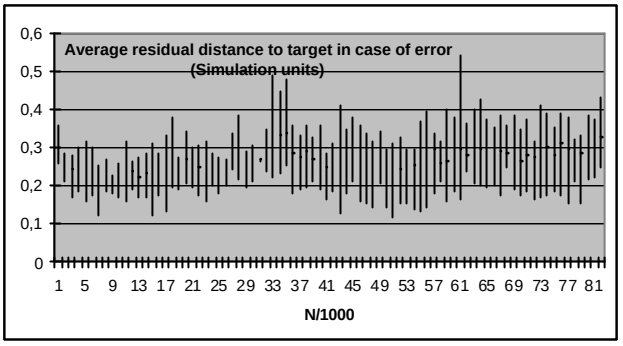

Figure 8. The average of the residual distance to target is presented as a function of N. The maximum, minimum and average values of the 5 learning runs are shown.

To evaluate how well the learned map performs, the *5%* residual error rate should be compared to the performance of a sensory motor loop that incorporates a true gradient operator, namely a gradient value that is analytically calculated. Surprisingly, the error rate for this configuration is about *12.5%*. Note that this result is met by ASMM for *N @ 9000* learning samples, which is quite small. This figure is significantly worse than the one exhibited by ASMM when *N* is above 60 000 learning samples. We cannot propose yet a definitive explanation for this result. Nevertheless, one can conjecture that the noise induced by the coarse gradient map involved in ASMM allows to escape from local minima of the error function *E*. If this can be proved, one can view ASMM as an efficient alternative method to minimize multivariate non linear quadratic functions.

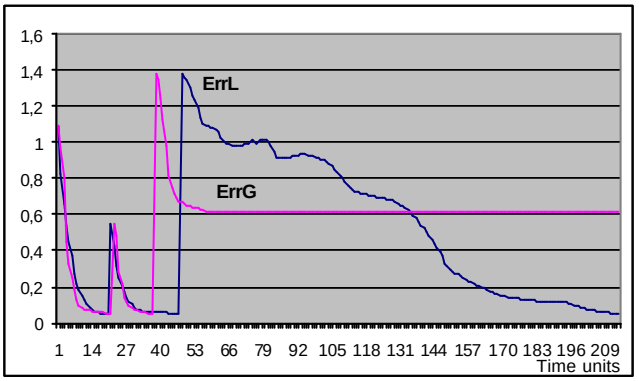

Figure 9. Error function E for ASMM (ErrL) and for GSMM (ErrG). The third target is reached by ASMM while GSMM falls into a local minimum.

Another experiment has been realized in which three successive targets and initial conditions have been

selected. The experiment runs for $N=85000$ samples. Figure 9 illustrates the similar evolution with time of the function E for the ASMM (ErrL) and for the GSMM (ErrG). The third target cannot be reached by the GSMM that falls into a local minimum, while the ASMM can find a path towards this target, escaping from local minima.

This is shown on the ErrL signal that is not monotonically decreasing and can be locally increased for ASMM, while ErrG signal is necessarily monotonically decreasing for GSMM.

7. A modular ASMM-based architecture for expressive gesture animation

The architecture consists of several interacting sub-systems, as shown in figure 10. A motor programs module provides spatio-temporal discrete targets used to drive the control and animation system. These targets are distributed through a real-time scheduler to appropriate learning-based models (ASMM), each one associated to a specific articulated chain. Note that this architecture is similar to the one proposed previously [18], except that the GSMM models are replaced by ASMM models.

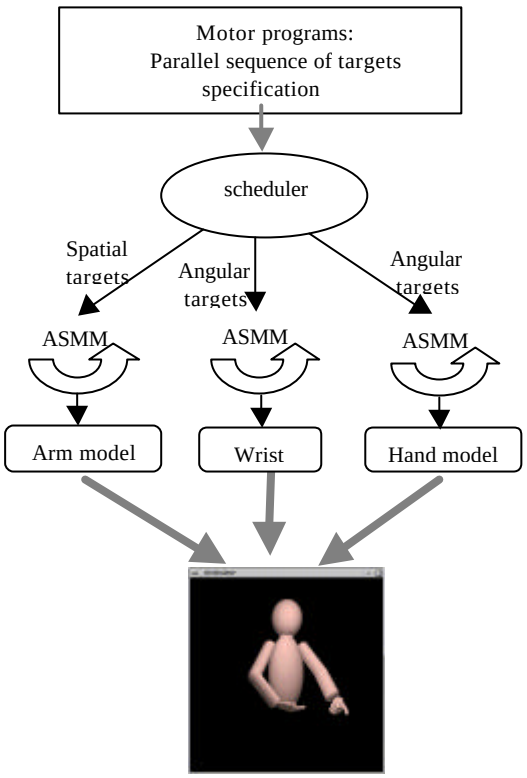

Figure 10. Control architecture of one hand-arm system

This system generates smooth pointing gestures and more complex expressive gestures involving co-articulation and synchronization mechanisms (figure 11).

One of the main interest of the learning approach is that it is more capable of escaping from local minima than the gradient-based approach. In the near future, we plan to learn motor programs using similar learning techniques on the basis of discrete observation data characterizing higher-level tasks, and to connect the system to a planning module.

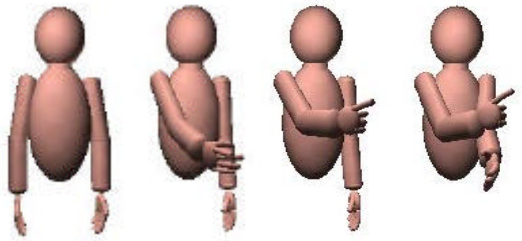

Figure 11. Snapshots sequence for a reaching task (right-hand, then left hand) specified as a 3D position, an orientation for the palm, and a configuration target for the set of fingers.

8. Conclusion

This paper presents an adaptive model of learning motion control for animating articulated systems. A Sensory-motor control model (GSMM), combining non linear sensory motor transforms is first described to solve in an iterative fashion the inverse kinematics problem according to a gradient descent strategy.

From this model, the learning of sensory to motor maps involved in sensory motor controlled systems has been addressed at the light of non parametric learning GRNN approaches, based on a variable kernel density estimator and the use of a kd-tree architecture to simulate neuron activation according to a near neighbor search. Despite the apparent high memory requirement needed by this kind of estimator, the proposed learning scheme is efficient when used to control multi-articulated chains with seven degrees of freedom. This result is obtained even if the number of learning samples is significantly below commonly accepted statistical figures. Nevertheless, the number of degrees of freedom cannot be too high in order to cope with the memory needs of the model. Since the knowledge of the analytical equations that describe the mechanical or kinematics system is not required, the above described ASMM model is generic and can be easily adapted to various kinds of sensory motor systems. The biological plausibility of ASMM-like models is supported by neuro-physiological arguments that can be found in the

literature. Finally, ASMM seems to show better performances than the sensory motor control model which integrates the true gradient operator. The residual learning noise in ASMM allows effectively to escape local minima of the error signal to minimize. This property potentially positions the ASMM as an efficient alternative method to minimize multivariate quadratic functions, according to a pseudo steepest descent strategy. The adaptive and generic properties of ASMM allow to handle the animation of a hand-arm system. As GSMM, these models can be easily combined in a modular architecture, and learned to generate coordination gestures. This offers some promising perspectives for the animation of virtual actors with biomimetic behaviors. Moreover, the ASMM method could be extended to learn both the controller and the direct system.

References

[1] Kawato M., Maeda Y., Uno Y., Suzuki R.. Trajectory Formation of Arm Movement by Cascade Neural Network Model Based on Minimum Torque Criterion. *Biological Cybernetics*, vol. 62, 1990, pp. 275-288.

[2] Wolpert D.M., Miall R.C., Kawato M. Internal models in the cerebellum. Trends in Cognitive Science, vol. 2, n°9, 338-347, 1998.

[3] Spoelstra J., Schweighofer N., Arbib M.A. Cerebellar learning of accurate predictive control for fast reaching movements. Biological Cybernetics, 82, 321-333, 2000.

[4] Bullock D., Grossberg S., Guenther F.H. A Self-Organizing Neural Model of Motor Equivalent Reaching and Tool Use by a Multijoint Arm. *Journal of Cognitive Neuroscience*, vol. 54, 1993, pp. 408-435.

[5] Jordan M.I. Computational motor control. In M. S. Gazzaniga (Eds.), *The cognitive neurosciences. Cambridge*, MA: MIT Press, 1995 pp. 587-609.

[6] Werbos P.J. An overview of neural networks for control. IEEE Control Systems Magazine, January 1991.

[7] Van de Panne M., Fiume E. Sensor-actuator networks. Proceedings of ACM SIGGRAPH'93, ACM Computer Graphics, août 1993, pp. 335-342.

[8] Van de Panne M., Kim R. & Fiume E. Virtual wind-up toys for animation. Graphics Interface'94. Banff, Alberta, Canada, May 16-17, 1994.

[9] Terzopoulos D., Tu X. & Grzeszczuk R. Artificial fishes with autonomous locomotion, perception, behavior and learning in a simulated physical world. In: Artificial Life IV Workshop, MIT Press, 1994.

[10] Grzeszczuk R. & Terzopoulos D & Hinton G. NeuroAnimator: Fast neural network emulation and control of physics-based models. In M. Cohen, ed., Proceedings of ACM SIGGRAPH 98, July 1998.

[11] Sim K. Evolving virtual creatures. *Proceedings of ACM SIGGRAPG'94*, Pp.15-22, 1994.

[12] Zhao J. & Badler N.I. Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures. ACM Transactions on Graphics, vol. 13, N°. 4, October 1994, pp. 313-336.

[13] Tolani D., Goswami A. & Badler N.I. Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. Graphical Models 62, pp. 353-388., 2000.

[14] Soechting J.K. & Flanders M. Errors in pointing are due to approximations in sensorimotor transformations. Journal of Neurophusiology, vol. 62, n° 2, pp. 582-594, 1989.

[15] Koga Y., Kondo K., Kuffner J. & Latombe J.C. Planning motions with intentions. Proceedings of SIGGRAPH'94, pp. 395-408, 1994.

[16] Gibet S. and Marteau P.F., A Self-Organized Model for the Control, Planning and Learning of Nonlinear Multi-Dimensional Systems Using a Sensory Feedback, *Journal of Applied Intelligence*, Vol.4, 1994, pp. 337-349.

[17] Lebourque T., Gibet S. High Level Specification and Control of Communication Gestures: the GESSYCA System, *Computer Animation'99*, Geneva, Switzerlan, 1999.

[18] Gibet S. Lebourque T., Marteau P.F. High Level Specification and Animation of Communicative Gestures. *Journal of Visual Languages and Computing*, 12, 657-687, 2001.

[19] Julliard F., Gibet S., RML: A specialized Parallel Language for 3D Motion Control Specification., *In International Applied Informatics Conference, Parallel and Distributed Processing Symposium*, Innsbruck, Austria, pp. 39-45, 2001.

[20] Duda, R. O., & Hart, P. E. Pattern classification and scene analysis. *New York, NY: Wiley*, 1973.

[21] Schaal, S. Nonparametric regression for learning nonlinear transformations. In: Ritter, H., Holland, O., Möhl, B. (eds.). *Prerational Intelligence in Strategies*, High-Level Processes and Collective Behavior. Kluwer Academic, 2001.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Learning Representation by Back-Propagating Errors. *Nature* 323:533-536.

[23] Bishop, C.M. Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK, 1995.

[24] D.F. Specht, A General Regression Neural Network IEEE Trans. Neural Networks, Vol. 2, No. 6, p568-576, Nov. 1991.

[25] D.F. Specht, Probabilistic Neural Networks, Neural Networks, 3, 1990, 109-118.

[26] Churchland, P. S. and SeSejnowski, T. J, *The computational brain*, MA:MIT Press, 1992.

[27] Friedman, J.H., Bentley J.L. and Finkel R.A., "an algorithm for finding best matches in logarithmic expected time", *ACM Trans. Math. Software*, 3, 209-226, 1977.

[28] Silverman B.W., *Density Estimation for Statistics and Data Analysis*, Monographs on Statistics and applied Probability 26, third edition, Chapman and Hall Ed., 1992.