



HAL
open science

Modeling Joint Synergies to Synthesize Realistic Movements

Matthieu Aubry, Frédéric Julliard, Sylvie Gibet

► **To cite this version:**

Matthieu Aubry, Frédéric Julliard, Sylvie Gibet. Modeling Joint Synergies to Synthesize Realistic Movements. (Revised Selected Papers), Feb 2009, Bielefeld, Germany. pp.231-242. hal-00505186

HAL Id: hal-00505186

<https://hal.science/hal-00505186v1>

Submitted on 22 Jul 2010

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.

Modeling Joint Synergies to Synthesize Realistic Movements

Matthieu Aubry¹, Frédéric Julliard¹, and Sylvie Gibet^{2,3}

¹ Université Européenne de Bretagne, LISyC/ENIB
{aubry, julliard}@enib.fr

² Université de Bretagne Sud, VALORIA

³ Centre INRIA Rennes Bretagne Atlantique
sylvie.gibet@univ-ubs.fr

Abstract. This paper presents a new method to generate arm gestures which reproduces the dynamical properties of human movements. We describe a model of synergy, defined as a coordinative structure responsible for the flexible organization of joints over time when performing a movement. We propose a generic method which incorporates this synergy model into a motion controller system based on any iterative inverse kinematics technique. We show that this method is independent of the task and can be parametrized to suit an individual using a novel learning algorithm based on a motion capture database. The method yields different models of synergies for reaching tasks that are confronted to the same set of example motions. The quantitative results obtained allow us to select the best model of synergies for reaching movements and prove that our method is independent of the inverse kinematic technique used for the motion controller.

Key words: Virtual Humanoids, Movement Synthesis, Synergy, Reaching Gesture, Joint Synergies, Movement Learning

1 Introduction

Designing virtual characters that generate human-like gestures is still a major challenge, the research domain ranging from interactive ergonomics or virtual entertainment, to human gesture studies for different performances (sport, motor disabilities, etc.). One of the main issues that researchers are trying to resolve is how the human performer coordinates and controls a complex musculo-skeleton system, with many degrees of freedom (DoFs). The rotations applied by the muscles on the DoFs controlling the different joints - shoulder, elbow, wrist - are combined in order to perform a smooth movement. When groups of muscles, corresponding to one or several DoFs, cooperate in this way, they form a synergy [1, 2].

We follow here the assumption that synergies can be modeled in a flexible and dynamic fashion in order to produce a reaching movement, according to the task to perform (reaching different target points) and to physical characteristics of

the individual. One of the basic problems underlying this notion of synergy, from a kinematic point of view, is the amount of use of each DoF within the formation of a movement [2]. Moreover, we may consider that synergies dynamically evolve through time. Therefore the temporal characteristics of the synergies along the movement are a key issue that need to be investigated.

In this paper, we present a new approach to imitate human reaching gestures, based on the modeling of these synergies. Here, the notion of synergy expresses both the spatial interrelation between joints and the dynamical patterns acting on each joint during the course of the movement. Our synergy model can be included into any inverse kinematics (IK) technique, which iteratively computes the posture of the arm system from the specification of goals defined as reaching targets in the 3D space. The originality of our approach relies in the incorporation into the sensorimotor model of an explicit joint synergy function which can be parametrized from captured movements. The goal is to finely reproduce the gesture while preserving the natural characteristics of the real movements. This approach is compared to other classical IK methods, and quantitatively analyzed for a set of reaching movements.

The rest of this paper is organized as follows: in section 2, we briefly discuss related works, and then continue with the mainstay of this paper; in section 3, we describe our method for controlling motion and modeling joint synergies; experiments and results are presented in section 5 for a set of reaching movements; and lastly we conclude by a discussion and give future perspectives for this type of work.

2 Related Work

Over the years, different methods aiming to produce natural gestures reproducing characteristics of real movements have been proposed. We try to highlight to what extent these works may explain the underlying synergies when performing movements.

Learning model of motion from examples has been an active and productive area of research. In these approaches, the main objective consists of synthesizing new sequences of movements from existing ones. Some approaches use Hidden Markov models [3], or use a Linear Dynamic System to learn the style of training motions [4]. Gibet et al. [5] use the Nadarada Watson estimate to learn local transformations of the Jacobian. Rose et al. [6] and Grochow et al. [7] propose respectively a Radial Basis Function model and a Scaled Gaussian Process Latent Variable model to learn inverse kinematics from human poses. Wang et al. [8] learn a probabilistic mapping between poses and style variables, using a parametric Gaussian mixture model. Ong et al. [9] propose a novel modular neural network architecture for learning inverse kinematics. Other approaches determine physical constraints by using Non Linear optimization techniques [10], or clustering models [11], learned from captured motion. Chai et al. [12] learn a statistical dynamical model from motion capture data and use this model to generate new motions from a variety of user-defined constraints. These approaches

produce natural-looking and physically plausible motions, and have proven to be effective for a large variety of human motions. Some of them result in the generation of new motions by interpolating or extrapolating the learned parameters. But, for these new motions, the physical laws of motion are not necessarily guaranteed. Moreover, these learning models do not explicitly incorporate the modeling of synergies.

Numerous inverse kinematics (IK) techniques have also been developed to produce realistic motions from the specification of geometrical user-defined constraints. Among these techniques, traditional IK solvers use numerical optimization methods, such as Jacobian pseudo inverse, which implicitly capture the correlation between joints when performing a movement. In order to avoid non realistic situations, constraints may be introduced, such as physical-based constraints [13], shape constraints [14], or ergonomic constraints [15]. These methods rely on the validity of the constraints for a given task and usually do not take into account features specific to an individual.

In order to take advantage of both methods, example-based IK methods are an effective alternative. They combine goal-directed and data-driven methods, thus relying on real motions with an explicit model for solving IK. Komura et al. [16] extract joints' weights from captured data and re-use them during synthesis to reproduce the synergies. Some recent studies perform inverse kinematics in low-dimensional space, such as latent spaces [17], and [18], thus including within the solver the linear combination between the DoF. In these cases, synergies are explicitly specified within the motion. However, these methods make the assumption that the inter-relation between the joints within the synergies may be represented by linear functions. Moreover, they require the passage of one posture to another, and do not take into account the dynamical effect of the synergies over time when performing a movement.

Our approach is also related to example-based IK methods. But in contrast to the above methods, we propose to model the dynamical effect of synergies over time. In the line of previous work [19], based on a sensorimotor optimization loop, we propose a novel approach which uses a motion capture database and a learning technique that automatically computes the parameters of the synergy model.

3 Proposed Model

3.1 Inverse Kinematics Formulation

In this section, we consider the control of a kinematic chain \mathcal{C} composed of n joints linking segments of different lengths, each segment representing a bone of the virtual character. Each joint is characterized by 1, 2 or 3 rotations, defining several DoFs. The set of rotations can be represented by the state vector $\Theta = \{\theta_1, \dots, \theta_n\}$.

We define the forward kinematic operator \mathcal{H} which computes the end effector situation \mathcal{X} of the chain \mathcal{C} , given its state Θ . Usually, the situation is determined

by a position and an orientation.

$$\mathcal{X} = \mathcal{H}(\Theta)$$

The IK problem can be defined as a method that determines a vector Θ so that the end effector is in the desired situation \mathcal{X}_d .

$$\Theta = \mathcal{H}^{-1}(\mathcal{X}_d)$$

As the human arm is a complex and redundant system, the kinematic chain \mathcal{C} cannot be inverted in general. Most of previous work solves the IK problem with local linearization methods. In this case, the IK formulation determines small variations of the posture from small variations of the end effector situation, thus trying to converge towards the desired situation. Different iterative methods can be used to solve IK. Whereas the pseudo-inverse methods can be expressed as:

$$\Delta\Theta = \lambda J(\Theta)^+ \Delta X$$

where $J(\Theta)^+$ is the pseudo-inverse of the Jacobian of \mathcal{C} corresponding to the state Θ , and λ is a scalar changing the rate of convergence. This pseudo-inverse can be obtained, for example, using the Singular Value Decomposition (SVD) of the Jacobian.

Another method: J^T , uses the transpose of the Jacobian $J(\Theta)$:

$$\Delta\Theta = \lambda J^T(\Theta) \Delta X$$

presents smaller convergence rates in the general case. In our study, we also define a slightly derived method of the J^T (called NJ^T) which computes the normalized variations of the rotations. The iterative algorithm resulting from the numerical solutions of IK can be represented in fig. 1, where \mathcal{O} is the optimized function (J^+ , SVD, J^T , NJ^T), \mathcal{I} the integrated function, and \mathcal{H} the forward kinematic transformation.

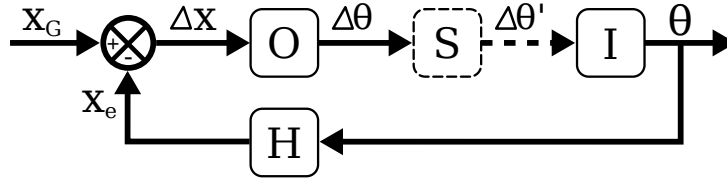


Fig. 1. Iterative method used to solve inverse kinematics.

Other optimization functions may be used, thus improving the convergence rate towards the solution, and avoiding singularities [13, 20]. Whereas these

methods propose optimized solutions to the problem of IK, they do not guarantee the natural quality of the movement through time. We will introduce below a synergy model which is compatible with a traditional IK formulation, and dynamically modifies the temporal evolution of the synthesized motion, according to characteristics of human arm movements.

3.2 Modeling Synergies

In our approach, synergies can be represented by dynamical functions interacting within a set of joints. More precisely, we define synergies as a combination of gains applied to each of the n DoFs and a temporal function. By modifying the level of involvement of the DoFs along the motion, we may completely control the velocity and the shape of the motion.

We define a synergy transformation as a function $\mathcal{S}(t, \Theta)$ of time and joint angular values. The calculation of the synergy gains is achieved according to the following equation:

$$\Delta\theta'_i = \mathcal{S}(t, \Theta) \cdot \Delta\theta = \phi_i(t) \cdot (M[\Theta] \cdot \Delta\theta)_i$$

where the function $\phi_i(t)$ and the matrix $M[\Theta]$ represent respectively the temporal part and the spatial part of the synergy function.

The synergy function $\mathcal{S}(t, \Theta)$ is thus parametrized, the parameters depending on the subject physical characteristics, as well as on the task. We will consider in our simulation a parameter vector p with a dimension m which may vary with \mathcal{S} ; its values are constant over the movement and is noted:

$$p = \{p_1, \dots, p_m\}$$

By using a global learning approach, based on the comparison between simulated movements and captured ones, our methodology consists in determining the generic synergy model for specific tasks and the set of parameters corresponding to an individual.

3.3 Learning Joint Synergies Parameters

We define a meta-heuristic based on simulated annealing which requires an optimization function \mathcal{O} , a synergy model \mathcal{S} and captured movements as training reference. A pseudo-code version of our meta-heuristic is given in algorithm 1. The principle of our meta-heuristic is to generate a random population of parameters (l. 1): $population = \{p_1, \dots, p_u\}$ and to make them evolve randomly by applying small variations (l. 4) on each parameter. The parameters are then evaluated for each training motion (l. 6-10), by setting the controller in the same initial posture and by simulating the motion for a task similar to the training motion's one. The distance between the training and the synthesized motion is then computed. If the evolutionary algorithm reduces this distance, the vector of parameters is kept otherwise previous values are restored (l. 11-16). These steps are repeated until the average score of the population is stabilized (l. 18) and finally the best parameter's vector is returned (l. 19).

Algorithm 1 Meta-heuristic for determining the best parameters p

```
1:  $population \leftarrow GenerateRandom()$ 
2: repeat
3:   for all  $parameters \in population$  do
4:      $parameters.evolve()$ 
5:      $score \leftarrow 0$ 
6:     for all  $mocap \in Trainings$  do
7:        $controller.setPosture(mocap.initialPosture)$ 
8:        $simulatedMotion = controller.synthesize(mocap.task)$ 
9:        $score \leftarrow score + distance(simulatedMotion, mocap.motion)$ 
10:    end for
11:     $deltaScore \leftarrow parameters.previousScore - score$ 
12:    if  $deltaScore < 0$  then
13:       $parameters.keep()$ 
14:    else
15:       $parameters.restore()$ 
16:    end if
17:  end for
18: until  $population.isStabilized()$ 
19:  $result \leftarrow population.findBest()$ 
```

3.4 Metric for Evaluating Synthesized Movements

The naturality of the synthesized motion is evaluated by superposing two virtual characters and playing captured and synthesized movements for the same task and comparing the two motions over time. In this case an error metric is computed by defining a distance between the two motions, based on the mean distance between postures along the motion. The calculation is stopped at the end of the shortest motion:

$$d(\mathcal{M}_A, \mathcal{M}_B) = \sum_{i=1}^{mini} \frac{dist(\mathcal{P}_{A,i}, \mathcal{P}_{B,i})}{mini}, mini = \min(length(\mathcal{M}_A), length(\mathcal{M}_B))$$

with \mathcal{M}_A motion A , $\mathcal{P}_{A,i}$ posture from motion A at frame i and $length(\mathcal{M}_A)$ the number of frames in motion A .

Such a distance gives a measure which takes into account both the velocity and the shape of the movement. First of all, by using the frame by frame comparison over time, we ensure that the movements' velocities can be compared. If both movements have a different frame rate, one of the movements has to be interpolated to match the other's frame rate. Secondly, the deformed shape of the movements is computed by using the average Cartesian distance between joints. This distance between two postures \mathcal{P}_A and \mathcal{P}_B is given by equation:

$$dist(\mathcal{P}_A, \mathcal{P}_B) = \left(\sum_{j=1}^l \frac{\|\mathcal{X}_{A,j} - \mathcal{X}_{B,j}\|}{l+1} \right) + \frac{\|\mathcal{X}_{A,e} - \mathcal{X}_{B,e}\|}{l+1}$$

where the kinematic chain \mathcal{C} is composed of l joints, the position of the j -th joint in global space for the posture \mathcal{P}_A is noted $\mathcal{X}_{A,i}$ and the position of the end effector is noted $\mathcal{X}_{A,e}$.

4 Experiments and Results

In our experiments, 14 reaching movements have been recorded for one subject. This one is seated on a chair and targets from different colors are uniformly distributed in front of him (cf fig. 2). He has been asked, from the rest posture, to reach a specific target and then come back to the same rest posture. To avoid the bias of finding the target, he has to locate the target before starting the motion. The rest posture has been chosen by the subject and is used as the starting position for each motion.

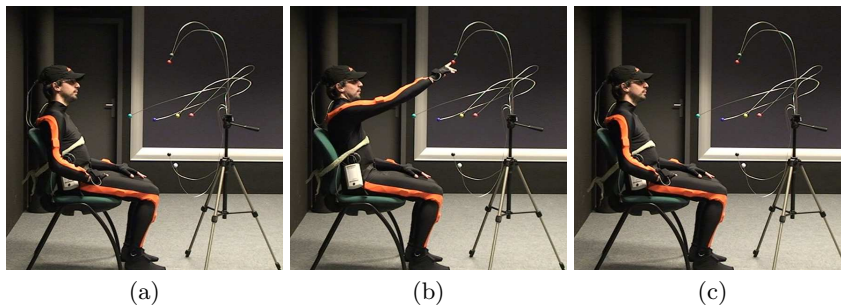


Fig. 2. A subject is seated, his back supported by the chair. Color targets are located in front of him. From its rest posture (a), he is asked to reach a specific target (b) and then come back to the same rest posture (c).

It should be noted that a Cartesian distance instead of an angular one allowed us to use two characters with the same joints but different DoFs. In fact, captured motions are recorded on 4 joints with a total of 12 DoFs, while the synthesis model has 4 joints but only 7 DoFs.

Three different analysis are performed. First, we test different models of synergies to find the best compromise between the number of parameters and the quality of the synthesized motions. Next, different optimization functions are implemented in order to evaluate the genericity of our model. And finally, some characteristics of the synthesized motion are compared to captured motions.

First of all, we define different gains models and compare their performances. These models used for reaching tasks are composed of gains and sigmoids as temporal functions. In particular we evaluate the influence of using same or different gains and sigmoids functions for each DoFs.

For the gains, we use diagonal matrix, one with the same gain for each DoF (M_A) and one with a different gain for each DoF (M_B). While M_A requires only

one parameter, M_B will require as many parameters as DoFs. Those matrix are defined by :

$$M_A = \begin{pmatrix} g_1 & 0 \\ & \ddots \\ 0 & g_1 \end{pmatrix}, M_B = \begin{pmatrix} g_1 & 0 \\ & \ddots \\ 0 & g_n \end{pmatrix}$$

We also defined three different time functions inspired from work of Gibet et al. [19]: one neutral and two using sigmoids $sig(t) = \frac{1}{1+e^t}$. These time functions are defined for each DoF by:

$$\phi_{A,i} = 1, \quad \phi_{B,i} = sig(-s.t + o), \quad \phi_{C,i} = sig(-s_i.t + o_i)$$

where ϕ_B uses the same parameters s (slope) and o (offset) for each DoF while ϕ_C uses one for each DoFs.

Therefore, four synergy functions are distinguished:

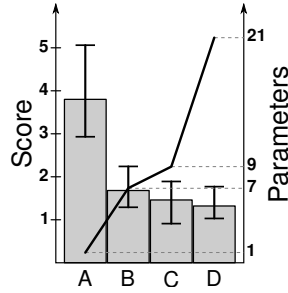
- $\mathcal{S}_A(t, \Theta)$ using M_A and ϕ_A ; $p = \{g_1\}$
- $\mathcal{S}_B(t, \Theta)$ using M_B and ϕ_A ; $p = \{g_1, \dots, g_n\}$
- $\mathcal{S}_C(t, \Theta)$ using M_B and ϕ_B ; $p = \{g_1, \dots, g_n, o, s\}$
- $\mathcal{S}_D(t, \Theta)$ using M_B and ϕ_C ; $p = \{g_1, \dots, g_n, o_1, \dots, o_n, s_1, \dots, s_n\}$

Taking one by one the 14 training motions, we obtained 14 sets of parameters associated to 14 distances for each model. To reduce the field of exploration of the meta-heuristic, we used $g_i \in [-5; 5]$, $s_i \in [1; 10]$ and $o_i \in [0; 20]$. These values, obtained by experimentation, also reduced the computational cost. The results in figure 3 show the average, minimum and maximum distances obtained for each model.

These results clearly show the necessity to separate the gains applied to the DoFs. In fact, a reduction of 55% of the average distance is obtained if we move from \mathcal{S}_A to \mathcal{S}_B . Next, the insertion of a sigmoid in \mathcal{S}_C allows us to reduce of 13% more, and separating the sigmoid on each DoF gives another reduction of 10%. In addition, whereas \mathcal{S}_D gives the best results over the whole set of motions (1.35), \mathcal{S}_C gives the minimum distance between two motions (0.90). This can be explained by the number of parameters used by \mathcal{S}_D (21) which is more than twice \mathcal{S}_C 's one (9). In fact, our meta-heuristic is more efficient on a smaller number of parameters.

For the rest of our experiments, we decide to choose the model \mathcal{S}_C , which is a compromise between obtaining a good average result and the number of parameters.

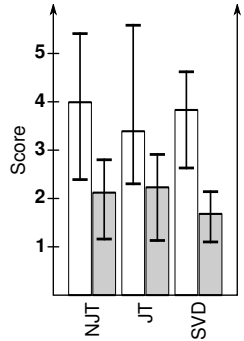
In order to test the genericity of our model according to optimization functions, we have implemented three IK algorithms: the transposed Jacobian (J^T), the normalized transposed Jacobian (NJ^T) and the pseudoinverse Jacobian using SVD. Each optimization function is tested for each of the 14 motions, once with \mathcal{S}_A (raw controller) and once using \mathcal{S}_C (model selected from previous experiment). This experiment allowed us to compare the average distance between motion synthesized with the raw controller and with the controller enhanced by our synergy model.



Model	$dim(p)$	Distance		
		avg.	min.	max.
\mathcal{S}_A	1	3.83	2.92	5.10
\mathcal{S}_B	7	1.71	1.28	2.28
\mathcal{S}_C	9	1.49	0.90	1.93
\mathcal{S}_D	21	1.35	1.02	1.81

Fig. 3. Average, minimum and maximum distance for the different synergy functions \mathcal{S}_A , \mathcal{S}_B , \mathcal{S}_C , and \mathcal{S}_D .

Results are displayed in figure 4. They show that for each optimization function, our synergy model decreases the distance between synthesized and captured motions. In fact, incorporating \mathcal{S}_C , we improve the results of 34% for J^T , of 66% for SVD, and of 47% for NJ^T . Furthermore, the average distance is lower when using the SVD method (1.71 versus 2.15 and 2.26), but we can observe that results achieved with J^T and NJ^T can be as good as the SVD's ones (only 6% difference between the minimum values).



Function	Distance		
	avg.	min.	max.
NJ^T	4.02	2.38	5.45
J^T	3.42	2.32	5.65
SVD	3.86	2.62	4.66

With \mathcal{S}_A (in white)

Function	Distance		
	avg.	min.	max.
NJ^T	2.15	1.15	2.84
J^T	2.26	1.12	2.95
SVD	1.71	1.09	2.18

Using \mathcal{S}_C (in yellow)

Fig. 4. Average, minimum and maximum distance between training motion and synthesized motion with \mathcal{S}_A (in white) or with \mathcal{S}_C (in gray).

Finally, the realism of the movements is evaluated for the different synergy models. As many researchers have demonstrated that the velocity profiles of simple reaching arm movements are approximately bell-shaped [19], we compare the velocities of synthesized and captured trajectories during the movement execution.

The hand velocities for recorded and synthesized movements, using a raw controller \mathcal{S}_A and a \mathcal{S}_C controller is displayed in figure 5. These curves clearly highlight the improvement made by the synergy model. Practically, the trajectories of the synthesized movements produced with \mathcal{S}_C are very close to the captured ones, whereas trajectories produced by the raw controllers do not match at all the bell-shaped curve.

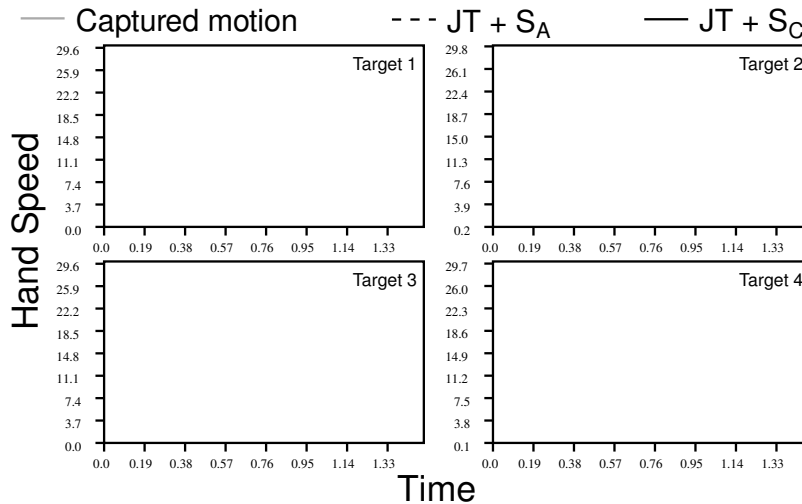


Fig. 5. Hand speed through time for captured motion (plain gray), synthesized motion with a raw controller (dotted black), and synthesized motion with the synergy model \mathcal{S}_C (plain black) for four different targets.

We also produced videos that make possible to compare a recorded task with the corresponding synthesized task with and without the synergy model (cf. fig. 6). While we can see that the use of \mathcal{S}_C instead of \mathcal{S}_A allowed to the synthesized motion to match the captured one both in speed and shape, these animations still need to be perceptually evaluated.

5 Discussion and perspectives

In this paper, we present a novel approach to finely reproduce reaching gestures by preserving natural human characteristics for an individual. The originality of the approach relies on the notion of joint synergy which is explicitly included into the model as a spatio-temporal function, parametrized from motion capture training data.

In order to precisely tune up our synergetic model, we used a preliminary database built from 14 motions obtained from a single person. We then tried to find a sufficiently small relevant set of synergetic parameters able to maximize the naturality of motions. For estimating this naturality, a motion metric is defined as an average Cartesian distance between joint positions, computed on the whole movement. This study reveals that pointing gestures can be naturally synthesized by the use of only one constant gain per DoF and one shared sigmoid.

The interchangeability of the optimization function, regardless of the synergetic function, has then been carried out by testing different numerical inversion methods as optimization functions. A quantitative analysis of the synthesized

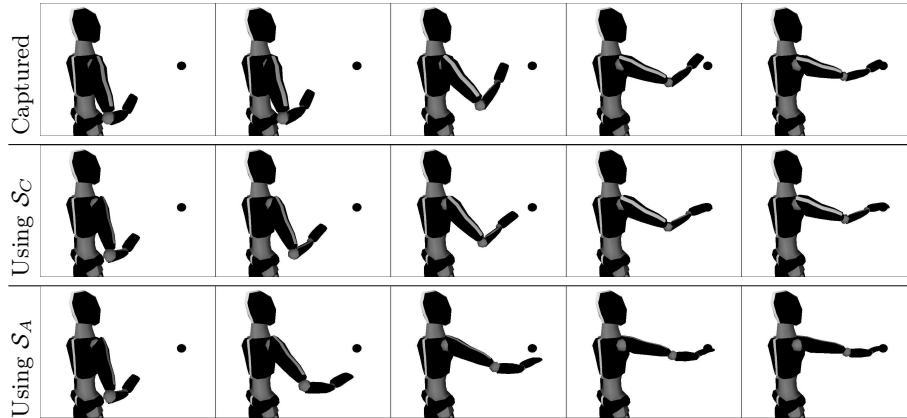


Fig. 6. Superposition of captured motion and synthesized motions both with S_A and with S_C as a synergy model. Motion synthesized with S_C match the captured one while motion synthesized with S_A does not (motion is quicker and elbow is moved too high).

motions has been performed. Generated motions reveal that the specific hand's velocity bell-shaped profile is close to the velocity of real motions, only when the synergetic function is employed.

Our experience has been successfully extended to a largest set of captured motions (192) from two different persons. Results highlight the ability of the model to learn individual joint synergies regardless of the number of captured data. Due to the genericity of our approach, we may incorporate synergy mechanisms into a wide range of inverse kinematics motor controllers. The meta-heuristic also proves to be fully independent of the task, of the synergetic function, and of the number of training data.

The main limitation of the current model concerns its generalization to the synthesis of any kind of gesture: parameters are learned motion by motion and it is obviously inconceivable to learn the whole accessible space. However, we observed that for a given set of parameters learned from a particular original target, it is possible to synthesize new motions, from nearby targets, whose characteristics are relatively closed to the features of the original motion. The parameters' space thus contains continuous regions that we still have to identify in order to define the applicability domain for a set of synergetic parameters. Moreover, this perspective should highlight the possibility to efficiently map each partition of the accessible space with a valid set of parameters, thus lowering the number of required training motions.

References

1. N.A. Bernstein. *The co-ordination and regulation of movements*. Oxford, New York: Pergamon Press, 1967.

2. M. L. Latash. *Synergy*. Oxford University Press, 2008.
3. M. Brand and A. Hertzmann. Style machines. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183 – 192. ACM Press/Addison-Wesley Publishing Co., 2000.
4. Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 465 – 472. ACM, 2002.
5. S. Gibet and P.F. Marteau. Expressive gesture animation based on non parametric learning of sensory-motor models. In *Computer Animation and Social Agents*, pages 465 – 472. ACM, 2003.
6. C. F. Rose, P.-P. J. Sloan, and M. F. Cohen. Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum*, 20:239 – 250, 2001.
7. K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522 – 531, 2004.
8. Y. Wang, Z. Liu, and L. Zhou. Key-styling: learning motion style for real-time synthesis of 3d animation. *Computer Animation and Virtual Worlds*, 17(3-4):229 – 237, June 2006.
9. E.-J. Ong and A. Hilton. Learnt inverse kinematics for animation synthesis. *Graphical Models*, 68(5-6):472–483, 2006.
10. C. Karen Liu, Aaron Hertzmann, and Zoran Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph.*, 24(3):1071 – 1081, 2005.
11. E. Oyama, A. Agah, K. F. MacDorman, T. Maeda, and S. Tachi. A modular neural network architecture for inverse kinematics model learning. *Neurocomputing*, 38-40:797 – 805, 2001.
12. J. Chai and J. K. Hodgins. Constraint-based motion optimization using a statistical dynamic model. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 8. ACM, 2007.
13. P. Baerlocher and R. Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20(6):402 – 417, 2004.
14. B. Le Callennec and R. Boulic. Interactive motion deformation with prioritized constraints. *Graphical Models*, 68(2):175 – 193, March 2006.
15. F. Yang, L. Ding, C. Yang, and X. Yuan. An algorithm for simulating human arm movement considering the comfort level. *Simulation Modelling Practice and Theory*, 13(5):437 – 449, 2005.
16. T. Komura, A. Kuroda, S. Kudoh, C.-L. Tai, and Y. Shinagawa. An inverse kinematics method for 3d figures with motion data. In *Computer Graphics International*, pages 266 – 271, 2003.
17. D. Raunhardt and R. Boulic. Motion constraint. *The Visual Computer*, 25(5):509 – 518, May 2009.
18. M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Reveret. Motion compression using principal geodesics analysis. *Eurographics*, 28(2):to appear, april 2009.
19. S. Gibet and P. F. Marteau. A self-organized model for the control, planning and learning of nonlinear multi-dimensional systems using a sensory feedback. *Applied Intelligence*, 4(4):337 – 349, 1994.
20. S. R. Buss and J.-S. Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3):37–49, January 2005.