Inverse kinematics using sequential Monte Carlo methods

Nicolas Courty, Élise Arnaud

Inverse Kinematics Using Sequential Monte Carlo Methods

Nicolas Courty[1] and Elise Arnaud[2]

[1] SAMSARA/VALORIA, European University of Brittany, Vannes, France
[2] Université Joseph Fourier, INRIA Rhône-Alpes, LJK, Grenoble, France

Abstract. In this paper we propose an original approach to solve the Inverse Kinematics problem. Our framework is based on Sequential Monte Carlo Methods and has the advantage to avoid the classical pitfalls of numerical inversion methods since only direct calculations are required. The resulting algorithm accepts arbitrary constraints and exhibits linear complexity with respect to the number of degrees of freedom. Hence, the proposed system is far more efficient for articulated figures with a high number of degrees of freedom.

1 Introduction

Given a kinematic chain described by a fixed number of segments linked by joints in 3D space, the forward and inverse kinematics problems can be derived. The first amounts to computing the pose of the figure given the values of the joint angles. The second is the process of determining the parameters of the kinematic chain in order to obtain a desired configuration. The latter has been extensively studied in computer animation due to its large number of applications, such as connecting characters to the virtual world, as well as in robotics, where manipulator arms are commanded in terms of joint velocities.

Classical approaches for solving inverse kinematics require numerical inversion operations that may have singularities and exhibit the classical problems encountered in numerical inversion.

In this paper, we propose to solve the problem with sequential Monte Carlo methods (SMCM), that are based on the importance sampling principle. The main advantage of using a sampling approach is that we solve the inverse kinematics using the direct kinematics, hence avoiding numerical inversion of the forward operator. To do so, we cast the problem into a hidden Markov model (HMM), whose hidden state is given by all the parameters that define the articulated figure. Hence, the state space consists of all the possible configurations of the state. The inverse kinematics is then re-formulated in a filtering framework. This allows us to derive a simple and efficient algorithm. The sequential aspect of the procedure is one of its keypoints. The algorithm produces a complete motion, satisfying all the required constraints, from the initial position to the target position as a result of an optimization procedure. Each intermediate pose corresponds to an optimization step of a sequential algorithm, not requiring any batch calculations. The contributions of our method to the domain of

F.J. Perales and R.B. Fisher (Eds.): AMDO 2008, LNCS 5098, pp. 1–10, 2008.

articulated character control are threefold: `(i)` our method does not require any explicit numerical inversion, `(ii)` any type of constraints can be added to the system in an intuitive manner, `(iii)` this method can be implemented in a few lines of codes without the need of complex optimization algorithms.

The paper is organized as follows. Next section provides a short related work on human figure animation. In section 3, we propose the Bayesian formulation of motion control, and explain how HMMs can be advantageouly used in that framework. An analogy with filtering formulation is done. In section 4, we focus on the specific statistical model we propose for inverse kinematics. Finally, results are presented in section 5.

2 Related Work

Creating realistic and plausible motions remains an open challenge within the animation community. Pure synthesis models like inverse kinematics have been well studied and used as a support in several other problems such as motion reconstruction in presence of missing markers, or retargetting [6,12,3]. Nevertheless, the fact that most of the time its solution is undetermined implies that several constraints need to be added to the produced motion [15,1,13]. The types of those constraints and the way to handle them in the resolution of the inverse problem is a critical part of the existing algorithms. Recent works propose to use motion capture data to constraint the motion [10,4,5]. Our approach uses a statistical description of the problem, similarly to [14] and [5]. This first addresses the motion-editing problem using a non linear Kalman Filter, while the second uses a learned statistical dynamic model in a constrained-based motion optimization framework. However, our methodology differs significantly from these works since it is not data-driven, and uses a sampling approach. To the best of our knowledge, no SMCM have been applied yet for inverse kinematics. Let us note that SMCM have been used to address the markerless motion capture problem in the computer vision community [8,7]. The problem is then over-constrained by image features, and SMCM are used to explore efficiently local minima. The problem adressed here is different since we aim at generating plausible motion trajectories by exploring an under-constrained state space. Finally, it is possible to outline some similarities with recent works in the domain of motion planning like [2] where a stochastic search is used to find a clear path in obstructed area. Again, our work differ from these because of the sequential aspect of our method that allows to adapt dynamically to changes in goals or environment.

3 Inference Problem and SMCM

Formulation overview. In this paper, we propose a statistical inverse kinematics solver. It is based on a Bayesian formulation of the problem, that enables us to combine motion priors, skeleton constraints (e.g. joint limits) and kinematic constraints. We denote $\mathbf{x} = \mathbf{x}_{0:F} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_F\}$ the sequence of poses

from the initial pose of the chain $\mathbf{x}_0$ to its final pose $\mathbf{x}_F$ satisfying the kinematic constraints. We denote $\mathbf{z}$ as the set of variables that we are interested to control and that takes into account the various constraints. The goal is to infer the most likely trajectory $\hat{\mathbf{x}}$ given $\mathbf{z}$. We have:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}) = \arg\max_{\mathbf{x}} \frac{p(\mathbf{z}|\mathbf{x})\ p(\mathbf{x})}{p(\mathbf{z})}, \tag{1}$$

where $p(\mathbf{z})$ is a normalizing constant. The involved components are the motion prior $p(\mathbf{x})$ and the likelihood $p(\mathbf{z}|\mathbf{x})$. The motion prior carries the *a priori* knowledge about the very nature of the motion ; whereas the likelihood $p(\mathbf{z}|\mathbf{x})$ gives an evaluation on how good the motion is with respect to the set of constraints.

In this paper, we propose to use *sequential* Monte Carlo techniques. The formulation (1) has to be modified to suit a sequential approximation of $p(\mathbf{x}|\mathbf{z})$. Let $\mathbf{z}$ be given by a desired trajectory of the controlled variables, i.e. $\mathbf{z} = \mathbf{z}_{0:F}$. Each $\mathbf{x}_k$ should satisfy all the constraints at time k. Using Bayes' theorem $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ may be expressed using $p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})$, $k \leq F$:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{0:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{0:k-1})\ d\mathbf{x}_k}, \tag{2}$$

$$\text{where}\ \ p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})\ d\mathbf{x}_{k-1}. \tag{3}$$

The new involved components are : the motion prior, now described as an *evolution prior* $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and a *likelihood* $p(\mathbf{z}_k|\mathbf{x}_k)$. Those two densities define the *model* of the system. This leads us to consider an HMM to model the motion control problem.

Sequential Monte Carlo methods. [9] The filtering recursion (2-3) does not yield closed-form expressions for general non-linear non-Gaussian models. To calculate this recursion, SMCMs propose to implement recursively an approximation of the sought filtering distribution $p(\mathbf{x}_k|\mathbf{z}_{0:k})$. This approximation consists of a finite weighted sum of N delta-Diracs centered on the hypothesized locations in the state space, called particles. These particles are instances, or copies of the system. A weight $w_k^{(i)}$ is assigned to each particle $\mathbf{x}_k^{(i)}$, $i = 1 : N$ to describe its relevance. Hence, the approximation is:

$$\hat{p}(\mathbf{x}_k|\mathbf{z}_{0:k}) = \sum_{i=1:N} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k). \tag{4}$$

Assuming that the approximation of $p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})$ is known, the recursive implementation of the filtering distribution is done by propagating the swarm of weighted particles $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1:N}$. The estimate of the state is obtained by maximizing the estimated distribution given by (4), i.e. $\hat{\mathbf{x}}_k$ is the *maximum a posteriori* (MAP) estimate. At each iteration the algorithm can be decomposed in three steps :

1. *exploration of the state space:* The set of new particles $\{\mathbf{x}_k^{(i)}\}_{i=1:N}$ is obtained using the importance sampling method. The particles are drawn from the *importance function* denoted by $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})$.
2. *calculation of the new weights:* To maintain a consistent sample, the weights are updated according to a recursive evaluation:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})\ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})}, \qquad \sum_{i=1:N} w_k^{(i)} = 1. \tag{5}$$

3. *mutation/selection of the particles:* As soon as the number of significant particles is too small, it is necessary to perform a resampling step. This procedure aims at removing particles with weak weights, and reproducing particles associated to strong weights

These three steps constitute the general framework of SMCMs. Different instances of this general algorithm can be defined. The simple method we use is built with the following rules: `(i)` the importance function coincides with the evolution law, i.e. $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$; `(ii)` this implies that the resulting weights are $w_k^{(i)} \propto w_{k-1}^{(i)}\ p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$. The application of this algorithm for the inverse kinematics problem is described in the next section.

4 SMCM for Inverse Kinematics

Notations. Let us consider a kinematic chain $\mathcal{C}$ composed of n joints. $\mathcal{C}$ is parameterized by the rotation vector $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\} \in$ the *articular space* $SO(3)^n$. Each joint of $\mathcal{C}$ is expressed as an unitary quaternion $\in S^3$. We define the forward kinematic operator $\mathbf{H}$ that computes the configuration of the end effector of the chain $\mathbf{P}$. $\mathbf{P}$ is defined by a position and an orientation, *i.e.* $\mathbf{P} \in$ the *task space* $SE(3)$. The forward kinematics equation is given as:

$$\mathbf{P} = \mathbf{H}(\mathbf{Q}) \tag{6}$$

The goal of inverse kinematics is to find a vector $\mathbf{Q}$ such that $\mathbf{P}$ is equal to a given desired configuration $\mathbf{P}_d$: $\mathbf{Q} = \mathbf{H}^{-1}(\mathbf{P}_d)$. $\mathbf{H}$ is a highly non linear operator difficult to invert.

We denote $\phi(\mathbf{q}\ ; \mathbf{m}, \mathbf{\Sigma})$ as the generalized Gaussian density for quaternion variable $\mathbf{q}$, called QuTem distribution [11]. It corresponds to the Gaussian distribution of covariance Σ in the tangent space at the quaternion mode $\mathbf{m}$ wrapped onto a hemisphere of S^3. For each joint, an associated quaternion and its QuTem distribution is defined. The covariance matrix describes the kinematic properties of the joint. For simplicity, we take a diagonal covariance matrix. A very small value of an element of the diagonal with respect to the others, may be assimilated to a degree of freedom with very small variability. This allows us to control the effective number of degrees of freedom of a given joint. The distribution of the vector of quaternions $\mathbf{Q}$ is denoted $\Phi(\mathbf{Q}\ ; \mathbf{M}, \Gamma)$. We assume that this last distribution defines a generalized Gaussian distribution over the pose space $SO(3)^n$, where n is the number of joints, and $\mathbf{M}$ and Γ are deduced from the QuTem parameters.

Model design. The goal of inverse kinematics is to estimate the value of the vector $\mathbf{Q}$ such that the resulting kinematic chain satisfies the kinematic constraints and other user-defined constraints. The rotation vector is now seen as a hidden random variable evolving in time until the final task is reached. The notation $\mathbf{Q}_k$ describes the random vector of quaternions at iteration k, corresponding to the state of the filter at time k. The notion of time refers to the iterated convergence steps toward the achievement of the goal *wrt.* the constraints.

In an operational IK system, it is desirable to be able to add several constraints to ensure particular effects. In our framework, those constraints can be handled either in the definition of the *evolution prior*, either in the *likelihood* characterization. Specifically, the nature of the kinematic structure (encoded in the Σ matrix) and hard constraints like joint limits or self-intersection avoidance are part of the sampling process (evolution prior), whereas soft or numerical constraints (such as minimizing a given function) are included in the likelihood.

Evolution prior. The evolution prior $p(\mathbf{Q}_k|\mathbf{Q}_{k-1})$ carries the *a priori* knowledge about the intrinsic nature of the motion, as well as biomechanical constraints and other binary constraints. A sample of this density is constructed by sampling from $\Phi(\mathbf{Q}_k\ ; f(\mathbf{Q}_{k-1})\ ,\ \Sigma)$, until $\mathbf{Q}_k$ is accepted, i.e. satisfies the binary constraints, including the joint limit constraints. This rejection/acceptance process guarantees that no impossible configurations will be generated. The mean of this density $f(\mathbf{Q}_{k-1})$ may describe any *a priori* knowledge on the kinematic chain motion. In case of motion control, it may be inferred from a learning phase. In this paper, we aim at demonstrating the advantages of our framework for the simple inverse kinematics problem where no *a priori* motion has to be verified. Hence, our model can be expressed as $f(\mathbf{Q}_{k-1}) \equiv \mathbf{Q}_{k-1}$. The main advantage of the algorithm proposed here is that it deals with inequality constraints in a very simple manner, using an acceptance/reject procedure.

Likelihood. The likelihood calculation $p(\mathbf{z}_k|\mathbf{Q}_k)$ gives an evaluation of how good the configuration is with respect to the desired task. If a unique kinematic constraint is imposed, the likelihood of a given state $\mathbf{Q_k}$ is evaluated by calculating the distance between the end effector configuration – obtained from equation (6) – and the desired configuration $\mathbf{P}_d$. The likelihood model used here is:

$$p(\mathbf{z}_k|\mathbf{Q}_k) \propto \exp \quad [-d_{\Sigma_\mathbf{z}}(\mathbf{P}_d\ ,\ \mathbf{H}(\mathbf{Q}_k))\ ]\,, \tag{7}$$

where $d_{\Sigma_\mathbf{z}}$ is the Mahalanobis distance with respect to the diagonal measurement noise covariance $\Sigma_\mathbf{z}$, given by:

$$d_{\Sigma_\mathbf{z}}(\mathbf{P}_d\ ,\ \mathbf{H}(\mathbf{Q}_k)) = (\mathbf{P}_d\ -\ \mathbf{H}(\mathbf{Q}_k))^t \Sigma_\mathbf{z}^{-1} (\mathbf{P}_d\ -\ \mathbf{H}(\mathbf{Q}_k)). \tag{8}$$

To guide the optimization towards the final pose, we propose to iteratively reduce the value of $\Sigma_\mathbf{z}$.

Other non-binary constraints may be added to the model. The methodology to do so is the following: each constraint has to be expressed in terms of a cost function whose value is 0 if the constraint is satisfied and large otherwise.

Supposing that j different constraints, assumed to be independent, are modeled by the cost functions $C_1 \dots C_j$, associated to noise covariances $\Sigma_1 \dots \Sigma_j$ then the likelihood is defined as:

$$p(\mathbf{z}_k|\mathbf{Q}_k) \propto \exp[-d_{\Sigma_\mathbf{z}}(\mathbf{P}_d \ , \ \mathbf{H}(\mathbf{Q}_k)) \ ] \ \prod_i \exp \ [-C_i^t \ \Sigma_i \ C_i] \tag{9}$$

Let us finally note here that setting the amplitude of the noises with respect to each constraint can be seen as a classifier between *important* and *optional* constraints. This is related in a sense to the weighted scheme of standard inverse kinematics. Enforcing strict priority levels within this framework is part of our future work.

Algorithm. A synopsis of the inverse kinematics filter is described in algorithm 1. Because of the jittered trajectories obtained if using directly the SMCM algorithm as previously described, an additional interpolation step is used to smooth the trajectory in $SO(3)$. This step is a Kalman-like interpolation step, controlled by a parameter α.

Algorithm 1. Inverse Kinematics filter

1. exploration of the state space using N copies of the system, i.e. for $i = 1...N$, draw $\mathbf{Q}_k^{(i)} \sim p(\mathbf{Q}_k|\mathbf{Q}_{k-1}^{(i)})$
2. calculation of the weights:
   - for $i = 1...N$, calculate $w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{Q}_k^{(i)})$
   - do weight normalization
3. calculation of the pose estimate:
   - obtain the first estimate as the MAP estimate, i.e. $\tilde{\mathbf{Q}}_k = \mathbf{Q}_k^{(j)}$ such as $w_{k-1}^{(j)} = \max\left(w_k^{(1)} ... w_k^{(N)}\right)$
   - obtain the final estimate as the smoothed estimate, i.e.: $\hat{\mathbf{Q}}_k = \text{interpolate}\left(\hat{\mathbf{Q}}_{k-1}, \tilde{\mathbf{Q}}_k, \alpha\right)$
4. if necessary, mutation/selection of the particles

5 Results

Simple positioning task. The first example is a simple positioning task for a kinematic model of an arm constituted of 4 segments with 4 pivot articulations, i.e. there are 3 rotational DOFs. The target is a point in $3D$ space. Figure 1.a shows the initial and the final configuration at convergence. Figure 1.b describes the convergence of our algorithm to the solution (distance to target) along several trials. Figure 1.c is a plot of the $3D$ trajectories of the end effector of the arm. Figure 1.d shows the evolution of the particles along the first frames of the animation. We then apply a similar task to a chain composed of 90 DOFs, i.e. 30 segments with 3 rotational degrees of freedom each. The center of mass of this chain is constrained to lie on a vertical line passing through the root of the chain. Another constraint is introduced to smooth the overall aspect of the

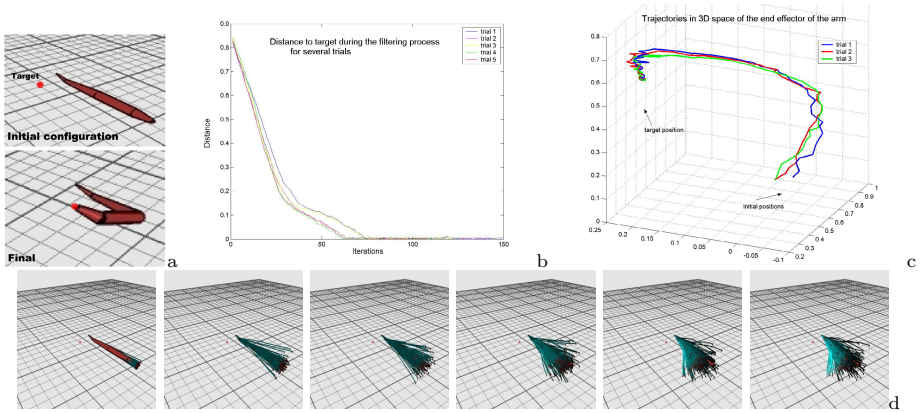

Fig. 1. Convergence of the algorithm (a) 3D view of the articular chain and the target (b) Distance to target. One can observe that the nature of the convergence does not change along the different trials (c) Trajectories of the end effector i $3D$ space. (d) **Importance sampling** along 6 iterations. Each particle is displayed as colored linked segments. This color is a function of the weights: the lighter the greater the weight.

chain. This constraint can be seen as a regularization function. It is expressed using the following energy function C:

$$C = \sum_{i=1}^{n} \sum_{j=-k}^{k} ||Log(\mathbf{q}_i^{-1}\mathbf{q}_{i+j})||. \tag{10}$$

This function aims at minimizing the differences, using geodesical distance, between successive rotations in the chain. We run this example at an approximate speed of 60 frames per second. Figure 2 shows the initial and final configurations. Let us note that this particular initial configuration stands for a typical singularity that lead numerical schemes to fail. In our case, efficiently sampling around this initial configuration provides pretty good solutions for an inferior computation time.

Computational performances. From a computational point of view, there are two major questions that arise in our framework. The first is, which is the number of particles that the simulation requires to be correct. Second, how this method compares to classical numeric inversion schemes. In figure 3, we investigate this first issue with the following setup: given a kinematic chain whose number of degrees of freedom is parameterizable, we perform a simple positioning task one thousand times, and report the percentage of success. In this case, the task is considered successful if the distance between the end effector and the target is below a certain threshold within a given number of iterations. This threshold is set to 0.001 unit for a 2 units original distance to target within 200 iterations. One can remark that with 30 particles the percentage of successful realizations is almost always 100 percent, which gives an idea of the minimal

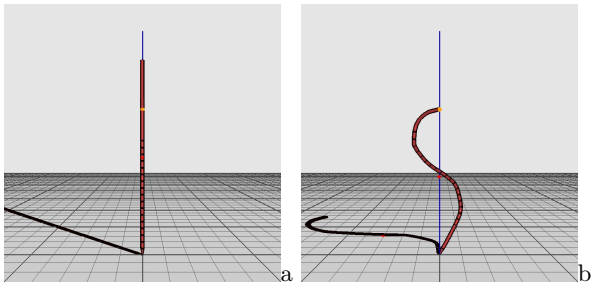

Fig. 2. Chain example. This chain is composed of 30 segments with 3 rotational degrees of freedom each (a) Initial configuration (b) final configuration. The initial configuration is a typical case where numerical inversion fails due to the singularity of the Jacobian matrix.

number of particles needed for a robust use. Another interesting issue is that this minimal percentage diminishes with the number of degrees of freedom. One may explain this behavior by the fact that, when there are more degrees of freedom, redundancy increases, as well the size of the solution state space, so that the sampled particles are more likely to be in the solution space of the task.

In Figure 3.b we compare the average time per iteration of two different numerical IK solutions. the Jacobian transpose method and the damped pseudo-inverse methods, and our method with 50 and 20 particles are compared. In both numerical methods the Jacobian is evaluated with a finite difference scheme. This scheme is very time costly at each iteration and tempers the intrinsic advantages of the Jacobian transpose method. By nature, our method leads to a linear complexity $o(kN)$ where k is the number of DOFs and N the number of particles. This makes our framework far more efficient for structures with large number of DOFs.

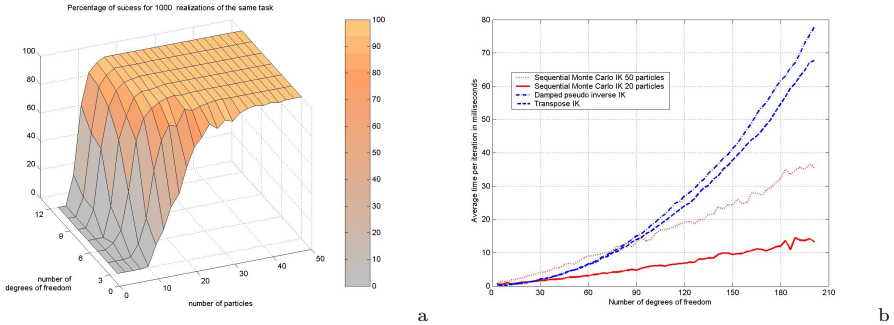

Fig. 3. Performances of our method (a) Relations between number of particles, number of degrees of freedom and task success. (b) Comparison with state-of-the-art IK methods. Note the linear nature of the complexity of our method (instead of exponential with numeric IK).

Human figure. As a first exemple, we consider a complete human figure with 40 joints. Snapshots of the resulting animation are shown in Figure 4. For this example, we add to the state the root position, that is the pelvis, so that the whole figure can move in the $3D$ space. The feet are constrained to stay on the floor, while the left and the right arms are given two different targets. The feet constraint is enforced by the acceptance/rejection strategy.

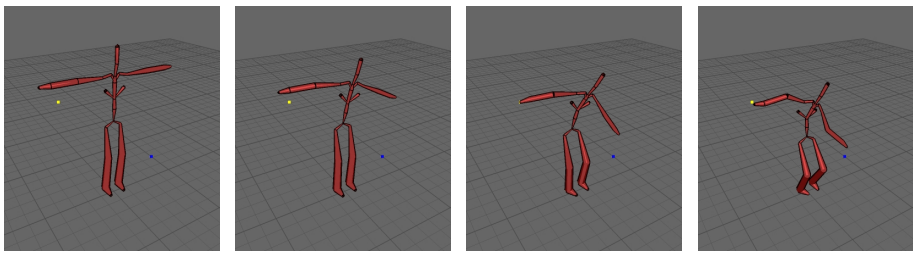

Fig. 4. Human figure animation In this animation, feet are constrained to lie on the floor, the right hand is linked with the yellow dot while the left arm has the blue dot as target. Notice how the knees bend to achieve the task.

Hand animation. As a second example, the considered chain is a forearm with a hand. In this animation, each fingertip is given a target empirically determined. Two sets of targets are chained during the animation. Figure 5 shows images from this animation. In order to increase the realism of the produced animation, we add a biomechanical constraint linking the last two joints of each fingers. Note the difficulty of handling such a kinematic configuration if a classical inverse kinematics scheme was used.

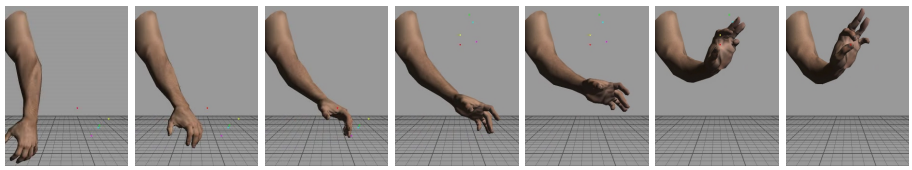

Fig. 5. Hand animation In this animation the fingers were given a target position represented as colored dots in the images

6 Conclusion and Future Work

In this article we introduced Sequential Monte Carlo Methods in the context of computer animation to solve inverse kinematics problem. We proposed a new modelization of the motion control problem in terms of hidden Markov models. Our inverse kinematics filter is very simple to implement and test. It runs very fast and provides a totally original alternative to solve this classical problem.

Only direct calculations of the kinetic model are needed. The validation of our model has been done under several different situations ranging from simple positioning tasks to hand animation, with convincing results. Future work will follow two main directions: evaluating the cost of adding constraints *wrt.* the number of needed particles, and adding more complex evolution priors (eventually learnt from motion capture data).

Acknowledgement. This work is part of the ARC Fantastik, supported by INRIA France. The authors would like to thank B. Cernuschi-Frías, L. Reveret and M. Tournier for fruitful discussions and suggestions on this work.

References

1. Baerlocher, P., Boulic, R.: An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. The Visual Computer 20(6), 402–417 (2004)
2. Bertram, D., Kuffner, J., Dillmann, R., Asfour, T.: An integrated approach to inverse kinematics and path planning for redundant manipulators. In: ICRA, pp. 1874–1879 (2006)
3. Le Callennec, B., Boulic, R.: Interactive motion deformation with prioritized constraints. Graphical Models 68(2), 175–193 (2006)
4. Carvalho, S., Boulic, R., Thalmann, D.: Interactive Low-Dimensional Human Motion Synthesis by Combining Motion Models and PIK. Computer Animation & Virtual Worlds 18(4–5), 493–503 (2007)
5. Chai, J., Hodgins, J.K.: Constraint-based motion optimization using a statistical dynamic model. ACM Tra. on Graphics (Proc. SIGGRAPH) 26(3), 686–696 (2007)
6. Choi, K.J., Ko, H.S.: Online motion retargetting. Journal of Visualization and Computer Animation 11, 223–235 (2000)
7. Sminchisescu, C., Triggs, B.: Fast mixing hyperdynamic sampling. Image and Vision Computing (2006)
8. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. International Journal of Computer Vision 61(2), 185–205 (2005)
9. Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo methods in practice. Springer, New York (2001)
10. Grochow, K., Martin, S., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. ACM Tra. on Graphics (Proc. SIGGRAPH) 23(3), 522–531 (2004)
11. Johnson, M.P.: Exploiting quaternions to support expressive interactive character motion. PhD thesis, Massachusetts Institute of Technology (2003)
12. Monzani, J.-S., Baerlocher, P., Boulic, R., Thalmann, D.: Using an intermediate skeleton and inverse kinematics for motion retargetting. Computer Graphics Forum 19(3) (2000)
13. Sentis, L., Khatib, O.: Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. Int. Journal of Humanoid Robotics 2(4) (2005)
14. Tak, S., Ko, H.-S.: A physically-based motion retargeting filter. ACM Tra. On Graphics (TOG) 24(1), 98–117 (2005)
15. Yamane, K., Nakamura, Y.: Natural motion animation through constraining and deconstraining at will. IEEE Tra. on Visualization and Computer Graphics 09(3), 352–360 (2003)