Sequential Monte Carlo Inverse Kinematics

Nicolas Courty, Élise Arnaud

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Sequential Monte Carlo Inverse Kinematics

Nicolas Courty — Elise Arnaud

N° 6426

December 2007

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6426--FR+ENG

Sequential Monte Carlo Inverse Kinematics

Nicolas Courty*, Elise Arnaud†

Thème COG — Systèmes cognitifs
Projet Perception

Rapport de recherche n° 6426 — December 2007 — 21 pages

Abstract: In this paper we propose a new and original approach to solve the Inverse Kinematics problem. Our approach has the advantages to avoid the classical pitfalls of numerical inversion methods such as singularities, accept arbitrary types of constraints and exhibit a linear complexity with respect to degrees of freedom which makes it far more efficient for articulated figures with a high number of degrees of freedom. Our framework is based on Sequential Monte Carlo Methods that were initially designed to filter highly non-linear, non-Gaussian dynamic systems. They are used here in an online motion control algorithm that allows to integrate motion priors. Along with practical results that show the effectiveness and convenience of our method, we also describe potential follow-ups for our work.

Key-words: inverse kinematics, sequential Monte Carlo method, aticulated motion, motion control

* Université de Bretagne Sud
† Université Joseph Fourier, Inria Rhône-Alpes

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Cinématique inverse par approche séquentielle de Monte Carlo

Résumé : Ce rapport présente une méthode originale au problème de cinématique inverse. Les avantages de notre approche sont (i) d'éviter les singularités numériques des méthodes classiques, (ii) de pouvoir prendre en compte tous types de contraintes, et (iii) d'être de compléxité linéaire par rapport au nombre de degrés de liberté de la chaîne articulaire considérée. Cette dernière caractéristique rend la méthode très efficace pour les chaînes de grande dimension. Notre approche s'appuie sur une méthode séquentielle de Monte Carlo, initialement conçue pour répondre au problème de filtrage non linéaire et non gaussien. Ici, cette méthode est mise en œuvre pour définir un algorithme de contrôle de mouvement en ligne, permettant la prise en compte de modèle de mouvement *a priori*. Les résultats obtenus démontrent la simplicité et l'efficacité de notre algorithme.

Mots-clés : cinématique inverse, méthode séquentielle de Monte Carlo, mouvement articulé, contrôle du mouvement

1 Introduction

Given a kinematic chain described by a fixed number of segments linked by joint angles, the forward and inverse kinematics problems can be derived. The first one amounts to computing the pose of the figure given the values of the joint angles. The second one is the process of determining the parameters of the kinematic chain in order to achieve a desired configuration. The latter has been extensively studied in computer animation due to its huge number of applications, such as connecting characters to the virtual world (feet landing on top of terrain, or hands lining up with doorknobs), as well as in robotics, where manipulator arms are commanded in terms of joint velocities.

While the forward problem has a unique solution, the inverse problem does not in the general case. Because of this, in the inverse problem, one needs to make explicit any available *a priori* information on the model parameters. Traditional ways of solving inverse kinematics rely on analytical or numerical methods, where some constraints are added to choose one of the solutions. Carrying out such methods becomes rapidly difficult for a large number of model parameters.

A very general theory to solve inverse problems is obtained when using a probabilistic point of view, where the *a priori* information on the model parameters is represented by a probability distribution over the model space (i.e. the state space). This *a priori* probability distribution is transformed into the posterior probability distribution, by incorporating a theory (relating the model parameters to some observable parameters) and the actual result of the observations (with their uncertainties). The probabilistic formulation of the inverse problem requires a resolution in terms of **samples** of the posterior probability distribution in the model space. This, in particular, means that the solution of an inverse problem is not a model but a collection of models (that are consistent with both the data and the *a priori* information). The generation of this collection of possible figures can be accomplished by means of an efficient Monte Carlo method.

Following this direction of work, we propose to solve the inverse kinematics problem using a Monte Carlo approach. Traditional inverse problem solvers use non-sequential Monte Carlo optimization tools such as the Metropolis algorithm [22]. In this paper, we present how **sequential** Monte Carlo methods (SMCM) can be advantageously used for inverse kinematics. The inverse kinematics is then re-formulated in a filtering framework. This allows us to derive a simple and efficient algorithm that can be seen as a filter whose state is the entire complex articulated figure. The sequential aspect of the procedure is one of its keypoints. The algorithm produces a complete motion, from the initial position to the target position as a result of the optimization procedure where each intermediate pose corresponds to an optimization step. The motion velocity is directly dependent on the filter parameters, making the algorithm flexible. The produced motions may then be used by an animator or an interactive animation system.

The contributions of our method to the domain of articulated character control are threefold:

- our method does not require any explicit numerical inversion method to solve the control problem,
- any type of constraints can be added to the system in a simple and intuitive manner provided that an *evaluation function* can be provided (no derivation *wrt.* articular space is required),
- this method can be implemented in a few lines of codes and tested easily without the needs of complex optimization algorithms.

Let us remark that another strong motivation for this work comes from motor neuroscience where Körding and Wolpert [16] have recently highlighted the bayesian nature of sensori-motor learning and the role of uncertainty in the realization of a motor task. Though those aspects still have to be clarified, we believe that a statistical approach of motion control can integrate those aspects in a successful way.

The remainder of the paper is organized as follows. Related works are described in section 2. The theoretical background on inverse kinematics is recalled in section 3. In section 4, we propose the Bayesian formulation of inverse kinematics and do an analogy with the filtering problem formulation. We make an overview of the existing solutions with specific attention to sequential Monte Carlo methods. In section 5, we will focus on the specific statistical model we propose. The last section before concluding presents results and possible extensions to our methodology.

2 Background

Creating realistic and plausible motions remains an open challenge within the animation community. Pure synthesis models like inverse kinematics [25, 24, 27] have been well studied in the literature and used as support in several other problems such as motion reconstruction (in presence of missing markers) or retargetting [7, 18, 4]. Nevertheless, the fact that most of the time its solution is undetermined implies that several constraints need to be added to the produced motion [3, 26, 2, 20]. 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 (see next section). In a recent work, Grochow *et al.* [14] propose to use motion capture data to constrain the result of an inverse kinematics algorithm. Though the results are quite interesting, this solution requires an example motion to be available, which is not always the case. We can draw a parallel with these types of works and those performing inverse kinematics in low dimensional space such as PCA space [5].

To the best of our knowledge, no sequential Monte Carlo methods have been applied for inverse kinematics. A similar approach in the use of a filtering framework to animate avatars is presented in [21]. This work addresses the motion-editing problem using a non linear Kalman Filter (the Unscented Kalman Filter). Their motion retargetting filter uses motion capture data to compute a predictive figure that is corrected using kinematic and dynamic constraints. Our methodology differs from this work in two major ways. First, we

do not rely on motion capture data to produce new motions (in that sense our method is not data-driven). Second, even if the Unscented Kalman Filter is an interesting solution, SMCM has proved it can overcome its performance for highly non linear multi-modal inference [12]. Recently, Chai and colleagues [6] have proposed the use of a statistical dynamic model in a constraint-based motion optimization framework. Their dynamic model is trained over a motion database and then used as a prior in a Maximum A Posterior framework. The main difference with our technique is that the motion is a result of a global optimization process, whereas our model is online and does not necessarily require any prior.

It is also interesting to note that sequential Monte Carlo methods have been used to address the markerless motion capture problem in the computer vision community. The key papers in that field are [11] that designs an algorithm in a multi-camera setting, and [9, 10] that estimate the articulated human motion from monocular data. As in this paper, those works propose the use of sequential Monte Carlo methods to generate plausible human figures, but evaluate the relevance of each figure using image features.

3 Theoretical background on Inverse Kinematics

In this section we consider a kinematic chain $\mathcal{C}$ composed of n joints and defined by the length of its different segments $\{l_1, \ldots, l_n\}$. $\mathcal{C}$ is parameterized by the following rotation vector $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\} \in SO(3)^n$ (which defines the *articular space*).

It is possible to define the forward kinematic operator $\mathbf{H}$ that computes the configuration of the end effector of the chain. Usually this configuration $\mathbf{P}$ is defined by a position and an orientation, *i.e.* $\mathbf{P} \in SE(3)$ (*task space*):

$$\mathbf{H} : \mathbb{R}^3 \times SO(3)^n \longmapsto SE(3) \tag{1}$$

$$\{\mathbf{r}_1, \mathbf{q}_1, \ldots, \mathbf{q}_n\} \longmapsto \prod_{1}^{n} \mathbf{M}_i(l_i, \mathbf{q_i}) = \mathbf{P} \tag{2}$$

where $\mathbf{r}_1$ is the root position of the chain and $\mathbf{M}_i(l_i, \mathbf{q_i})$ the homogeneous transformation matrix representing the rotation of a segment of length l_i by the quaternion $\mathbf{q}_i$. For clarity purposes this operation will be summarized by:

$$\mathbf{P} = \mathbf{H}(\mathbf{Q}) \tag{3}$$

Problem Statement The goal of inverse kinematics technics is to find a vector $\mathbf{q}_i$ such that $\mathbf{P}$ is equal to a given desired configuration $\mathbf{P}_d$. This problem amounts to the following non-linear inverse problem which does not always have a unique solution and is not always well-behaved:

$$\mathbf{Q} = \mathbf{H}^{-1}(\mathbf{P}_d) \tag{4}$$

Numerical resolution Most of the previous works on inverse kinematics solve equation (4) by using a local linearization method which amounts to converging to the solution by computing small variations $\dot{\mathbf{Q}}$ in the articular space that ensure the regulation from $\mathbf{P}$ to $\mathbf{P}_d$:

$$\dot{\mathbf{Q}} = -\lambda \mathbf{J}_{\mathbf{Q}}^{+}(\mathbf{P} - \mathbf{P}_d) \tag{5}$$

where $\mathbf{J}_{\mathbf{Q}}^{+}$ is the pseudo inverse of the Jacobian of $\mathcal{C}$ evaluated around the configuration $\mathbf{Q}$, and λ a scalar which sets the rate of convergence. Evaluation of the Jacobian (matrix of partial derivatives $\{\frac{\partial P_j}{\partial \mathbf{q}_i}\}$) is usually done with finite difference methods. The computation of the pseudo-inverse is one of the critical parts of the inverse kinematics. Some works [25, 24, 13] have explored the possibility to use the transpose of the Jacobian $\mathbf{J}^t$. As this solution relies on the assumption of the convexity of $\mathbf{H}$ (which is far from being the case), it has proved to be less efficient in the general case and exhibits smaller convergence rates. One of the most common techniques relies on the Singular Value Decomposition (SVD) which has the advantage of being robust to ill-posed problem since singularities can be detected and treated during the process. It is also current to consider a slightly modified version of the pseudo-inverse that guaranties that singularities are avoided; it is referred to as the damped or singularity robust (SR) pseudo-inverse [19, 17, 26].

Adding constraints Since the number of degrees of freedom in our kinematic chain is (most of the time) greater than the size of the task space, the number of solutions is usually infinite. With the pseudo-inverse approach the minimal norm solution is chosen by the algorithm, but it can be useful to control with additive constraints the choice of this solution. This operation is possible thanks to the use of projection operator $(\mathbb{I} - \mathbf{J}^{+}\mathbf{J})$ that allows us to project a constraint (or secondary task) on the null-space of $\mathbf{J}$. The new solution is given by:

$$\dot{\mathbf{Q}} = \mathbf{J}_{\mathbf{Q}}^{+}\dot{\mathbf{P}} + (\mathbb{I} - \mathbf{J}^{+}\mathbf{J})\frac{\partial h}{\partial \mathbf{q}_i} \tag{6}$$

where $\mathbb{I}$ is a $n \times n$ identity matrix and h is usually expressed as a generic cost function that needs to be analytically derived *wrt.* articular parameters. This formulation ensures that the secondary task will have no effects on the regulation from $\mathbf{P}$ to $\mathbf{P}_d$. This secondary task has been extensively used, notably for enforcing joint limits [8] or control the position of the center of mass [3] for instance. Recent works deal with adding several levels of constraints to the system [2, 4, 20]. In this particular theoretical framework (also called Prioritized inverse kinematics), the difficulty is to balance and order the different constraints (which is usually performed manually) without *a priori* knowledge on how the global task will be realized.

4 Inference problem and SMCM

4.1 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 prior, skeleton constraints (i.e. joint limits) and kinematic constraints. We denote $\mathbf{x} = \mathbf{x}_{0:M} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_M\}$ the sequence of poses from the initial pose of the chain $\mathbf{x}_0$ to its final pose $\mathbf{x}_M$ satisfying the kinematic constraints. The goal is to infer the most likely trajectory $\hat{\mathbf{x}}$ given the set of kinematic constraints $\mathbf{z}$. We have:

$$\hat{\mathbf{x}} \quad = \quad \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}) = \arg\max_{\mathbf{x}} \frac{p(\mathbf{z}|\mathbf{x})\ p(\mathbf{x})}{p(\mathbf{z})} \tag{7}$$

where $p(\mathbf{z})$ is a normalizing constant. The involved components are the motion prior $p(\mathbf{x})$ and the constraint likelihood $p(\mathbf{z}|\mathbf{x})$. The motion prior carries the *a priori* knowledges about the intrinsic nature of the motion, as well as biomechanical constraints ; whereas the likelihood calculation gives an evaluation on how good is the pose with respect to the kinematic constraints that have to be satisfied.

Two main families of methods are used to solve this inference problem. The first one relies on a direct estimation of the *maximum a posteriori* $\hat{\mathbf{x}}$ through an optimization procedure of $-\log(p(\mathbf{z}|\mathbf{x})\ p(\mathbf{x}))$. An example of this approach has been recently used for constraint-based motion optimization [6]. Monte Carlo methods are the second family. They approximate the probability density itself $p(\mathbf{x}|\mathbf{z})$ and *then* estimate $\hat{\mathbf{x}}$ through the use of *maximum a posteriori* or *minimum mean square error* estimates. Traditionally solvers from these two families are non-sequential, that make them improper for an efficient on-line animator use.

In this paper, we propose to use a *sequential* Monte Carlo technique. The resulting algorithm has the advantage of being at the same time easy to implement, robust and adapted to online practice. The formulation (7) has to be modified to suit a sequential approximation of $p(\mathbf{x}|\mathbf{z})$. Let suppose that constraints $\mathbf{z}$ can be decomposed into a set of constraints $\mathbf{z}_{0:M}$; each $\mathbf{z}_k$ has to be satisfied at point of time k. Typically, this can be translated into a progressive hardening of the kinematic constraints. Then $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ is expressed using $p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})$ $(k \leq M)$:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{0:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{0:k-1})\ d\mathbf{x}_k}, \tag{8}$$

where

$$p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})\ d\mathbf{x}_{k-1} \tag{9}$$

To derive this expression, one has to suppose the *hidden state process* $\mathbf{x}_{0:M}$ to be Markovian. The new involved components are : the motion prior, now described as an *evolution prior* $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and an *instantaneous constraint likelihood* $p(\mathbf{z}_k|\mathbf{x}_k)$. Those two densities define the *model* of the system.

Doing an analogy between constraints and observations, we can recognize here a filtering problem. The filtering recursion (8 - 9) yields closed-form expressions only for specific cases. The most well-known case is the Kalman filter for linear Gaussian models. Non optimal extensions of the Kalman filter, based on a Gaussian approximation of the filtering distribution (Extended Kalman filter, Unscented Kalman filter [23], [21]), have been devised for non linear systems. In the general multi-modal case, such an approximation is not satisfactory. For general non-Gaussian models, the recent development of sequential Monte Carlo approaches [1, 12] has lead to new efficient algorithms. Before commenting upon the specific model we propose for inverse kinematics, we describe these methods in the next subsection.

4.2 Sequential Monte Carlo methods

The idea behind sequential Monte Carlo algorithms is very simple. These techniques propose to implement recursively an approximation of the sought density $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ (called the filtering distribution). This approximation consists in a finite weighted sum of N Diracs centered on hypothesized locations in the state space – called particles – of the initial system $\mathbf{x}_0$. At each particle $\mathbf{x}_k^{(i)}$ $(i = 1 : N)$ is assigned a weight $w_k^{(i)}$ describing its relevance. This approximation can be formulated with the following expression:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) \approx \sum_{i=1:N} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k). \tag{10}$$

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 $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1:N}$. At each time instant (or iteration), the algorithm can be decomposed into three steps :

1. *exploration of the state space:* The set of new particles $\{\mathbf{x}_k^{(i)}\}_{i=1:N}$ is drawn from an approximation of the true distribution $p(\mathbf{x}_k|\mathbf{z}_{0:k})$, called the *importance function* and denoted $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})$. The closer the approximation to the true distribution, the more efficient the filter.

2. *evaluation of particles relevance using the observations (i.e. calculation of the new importance weights):* The importance weights $w_k^{(i)}$ account for the deviation w.r.t. the unknown true distribution. To maintain a consistent sample, the importance weights are updated according to a recursive evaluation as the new measurement $\mathbf{z}_k$ becomes available:
$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})\ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})}, \qquad \sum_{i=1:N} w_k^{(i)} = 1. \tag{11}$$

3. *mutation/selection of the particles:* From time to time, it is necessary to perform a resampling step. This procedure aims at removing particles with weak normalized

weights, and multiplying particles associated to strong weights, as soon as the number of significant particles is too small. Consequently, resampled particles tend to be concentrated in areas where important features exist.

These three steps (sampling / calculation of the importance weights / resampling) constitute the general framework of sequential Monte Carlo filter. Then, different instances of this general algorithm can be defined according to the choice of the importance function and/or the choice of the resampling strategy (see [12]). In particular, the simple method we use is built with the following rules: (a) to set the importance function to the evolution law, i.e. $\pi(\mathbf{x}_k|\mathbf{x}^{(i)}_{0:k-1}, \mathbf{z}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}^{(i)}_{k-1})$; (b) this implies the calculation of the weights using $w^{(i)}_k \propto w^{(i)}_{k-1}\ p(\mathbf{z}_k|\mathbf{x}^{(i)}_k)$. The application of this algorithm for inverse kinematics problem is described in the next section.

5 SMCM for Inverse Kinematics

In this section we present the inverse kinematics filter. After defining our notations in a first part, the second part describes the design of motion prior and likelihood when modeling the inverse kinematics problem. The last part is dedicated to the corresponding algorithm.

5.1 Notations

Let consider a kinematic chain parameterized by a vector of rotations. Each rotation, expressed as a unitary quaternion, corresponds to one joint and may have one, two or three degrees of freedom. Quaternions lives on the hypersphere S^3. We denote $\phi(\mathbf{q}\ ; \mathbf{m}, \Sigma)$ the Gaussian quaternionic density of variable $\mathbf{q}$. This density is called QuTem distribution in [15]. 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 [15].

To each joint is associated a quaternion and its QuTem distribution. The covariance matrix of this QuTem distribution designs the kinematic properties of the joint (number of degrees of freedom). This is depicted figure 1 where realizations on S^3 for three different covariance values of this distribution are shown. For instance, modeling a one degree of freedom (DOF) joint amounts to consider only one possible axis of rotation (and its opposite). This property is modeled by a diagonal covariance matrix with only one non-zero eigenvalue (Figure 1.c). Generalizing this idea, a 2 DOF joint will exhibit a diagonal covariance matrix two non-zero eigenvalues and full ball-and-socket joint will have three non-zero eigenvalues. This system allows to model for instance one DOF joint with a small variations allowed on the remaining DOF (just as in the human body where the biomechanical nature of the joints allow this), provided that the two other eigenvalues are much more smaller (the example of Figure 1.c is a good illustration of this). Appendix A details how to sample from the QuTem distribution.

Supposing that each quaternion of the kinematic chain follows a Qutem distribution, the distribution of the quaternion vector $\mathbf{Q}$ is denoted $\Phi(\mathbf{Q}\ ; \mathbf{M}, \Sigma)$. We assume that this

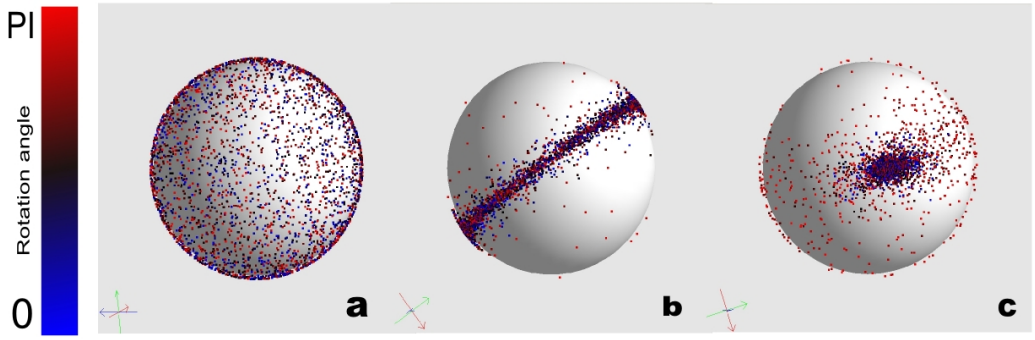


Figure 1: **Quaternion distribution** In this figure the equivalent representation axis–angle of a quaternion is adopted. Points on S^2 represent a rotation axis while the varying color stands for the rotation angle along the axis; 1000 samples over a QuTem distribution on S^3 with (a) $\sigma_1 = \sigma_2 = \sigma_3 = 1$ (3 DOF joint) (b) $\sigma_1 = \sigma_2 = 1, \sigma_3 = 0.05$ (2 DOF joint) (c) $\sigma_1 = 1, \sigma_2 = 0.1, \sigma_3 = 0.05$ (1 DOF joint)

last distribution defines a Gaussian distribution over the pose space $SO(3)^n$. $\mathbf{M}$ and Σ are deduced from the QuTem parameters.

5.2 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 the joint limits. One may also want to fix other constraints such as balance constraints for instance. As said before, we propose to reformulate this problem in a filtering framework. The rotation vector is now seen as a 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.

We choose to simply set the state vector $\mathbf{x}_k$ of the filter as the rotation vector $\mathbf{Q}_k$. The motion trajectory $\mathbf{x}_{0:M} = \mathbf{Q}_{0:M}$ will be the result of our algorithm, under the assumption that the optimization iteration time k also corresponds to the motion decomposition time. The sets of various constraints are taken into account in the design of the evolution prior and likelihood.

Evolution prior The evolution prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ carries the *a priori* knowledge about the intrinsic nature of the motion, as well as biomechanical constraints. As in the kinematic framework no *a priori* motion has to be verified, our model is simply a random walk model under the condition that the new sampled pose $\mathbf{x}_k$ enforces the joint limits of the skeleton. We propose the following general design:

$$\boxed{p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \text{random walk} \setminus \mathbf{x}_k \text{ enforces joint limits}}$$

This is equivalent to assume that the configuration remains constant along time. Angular displacements are only supported by a Gaussian noise model. This leads to:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \Phi(\mathbf{x}_k\ ;\mathbf{x}_{k-1}\ ,\ \Sigma_{\mathbf{x}}) \quad \backslash\mathbf{x}_k \text{ enforces joint limits} \tag{12}$$

The covariance matrix $\Sigma_{\mathbf{x}}$ contains the kinematic properties of each joint as explained in section 5.1. The following method is applied to sample from (12). Once a sample is drawn from Φ, an accept/reject procedure is applied to satisfied the condition on the joint limits: while sampling over the current configuration, if a joint does not enforce its corresponding limit, a new orientation value is sampled. This rejection/acceptance process guarantees that no impossible configurations will be considered. A drawback of this method is that it is not totally efficient (and may lead to the worst, but highly improbable, case of endless rejection). Therefore, it is reasonable to envisage other types of distribution (like a quaternionic log-normal), but this option has been left for future works.

Likelihood The likelihood calculation $p(\mathbf{z}_k|\mathbf{x}_k)$ gives an evaluation on how good is the configuration with respect to the kinematic constraints that have to be satisfied. It is designed as:

$$\boxed{p(\mathbf{z}_k|\mathbf{x}_k) = \exp(-|\text{distance to task}|_{\Sigma_{\mathbf{z}}})\ \prod \text{other constraints}}$$

where $|d|_\Sigma$ is the Mahalanobis distance $d^t\Sigma^{-1}d$, and $\Sigma_{\mathbf{z}}$ is the covariance of the noise.

As an example, If a unique kinematic constraint is imposed, the likelihood of a given state $\mathbf{x}_k$ is evaluated by calculating the distance between the end effector configuration – computed using the forward kinematic operator $\mathbf{H}$ described in equation (1) – and the desired configuration $\mathbf{P}_d$. The likelihood model is therefore:

$$p(\mathbf{z}_k|\mathbf{x}_k) \propto \exp\ \ (-|d(\mathbf{P}_d\ ,\ \mathbf{H}(\mathbf{x}_k))|_{\Sigma_{\mathbf{z}}}) \tag{13}$$

where $d(.,.)$ is the distance function in the task space.

Other constraints may be added into 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:

$$p(\mathbf{z}_k|\mathbf{x}_k) \propto \exp(-|\text{distance to task}|_{\Sigma_{\mathbf{z}}}).\prod_i \exp(-|C_i|_{\Sigma_{\mathbf{i}}}) \tag{14}$$

Examples of constraints and their corresponding cost function are given in the results section 6. Let us finally note here that setting the amplitude of the noises with respect to each constraint can be seen as a discrimination between *important* and *optional* constraints, which is related in a sense to the prioritization of constraints in traditional inverse kinematics.

5.3 Algorithm

A synopsis of the inverse kinematics filter based on the simple sequential Monte Carlo algorithm is described by algorithm 1. It computes the set of figures from its initial position $\mathbf{x}_0$ to its target position at $k = M$.

If we apply directly the simple method to the defined model, and use the set of Maximum *a posteriori* estimates as the final configuration sequences, the result is not satisfactory. Indeed, one drawback of SMCM is the jitter of the final estimate whose effect is a strongly noisy trajectory of the avatar. An online smoothing strategy is then added. Just after the calculation of the maximum *a posteriori* (MAP) estimate, we consider an additional step in the algorithm. Its goal is to calculate a new estimate $\hat{\mathbf{x}}_k$ using the MAP estimate $\bar{\mathbf{x}}_k$ with a Kalman-like smoothing algorithm on $SO(3)$:

$$\hat{\mathbf{x}}_k = \text{interpolate}\left(\hat{\mathbf{x}}_{k-1}, \bar{\mathbf{x}}_k, \alpha\right), \tag{15}$$

interpolate$(.,.,.)$ is an interpolation function which performs linear interpolation between cartesian data of the model and spherical linear interpolation (*slerp*) between rotation data. In our experiments we found that setting this value $\alpha =\sim 0.1$ provided good results. It has though to be clear that this smoothing strategy may induce a small delay in the realization of the task (if one consider a tracking task for instance).

6 Results

All the following examples were computed on a standard laptop and can be found in the accompanying video.

6.1 Simple positioning task

This first example is a simple positioning task with a model of arm constituted of 4 segments with 4 pivot articulation (3 rotational DOFs). The target is a point in the $3D$ space. Figure 2.a shows the initial and the final configuration at convergence. Figure 2.b describes the convergence of our algorithm to the solution (distance to target) along several trials. One can note that although this convergence differs in the different cases, the overall shape of the convergence does not change. It is still not clear for us how general this result can be. Figure 2.c is a plot of the $3D$ trajectories of the end effector of the arm. One can observe that here again the global shape of the trajectory is not modified from a trial to another. Those trajectories were obtained without the smoothing process described in the previous section, thus illustrating the noisy rough output of the filter without a smoothing process. Figure 2.d shows the evolution of particles along the first frames of the animation. As their respective weights are color coded, one can observe the importance sampling process which discriminates between particles.

We then applied a similar task to a chain composed of 90 DOFs (30 segments with 3 rotational degrees of freedom each). The center of mass of this chain was constrained to lie

Algorithm 1 Inverse Kinematics filter

- Estimation of the configuration sequence $\{\hat{\mathbf{x}}_k\}_{k=1:T}$

---

initialization :
for $i = 1, ..., N$, draw $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, $w_0^{(i)} = \frac{1}{N}$.

for $k = 1, ..., T$

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

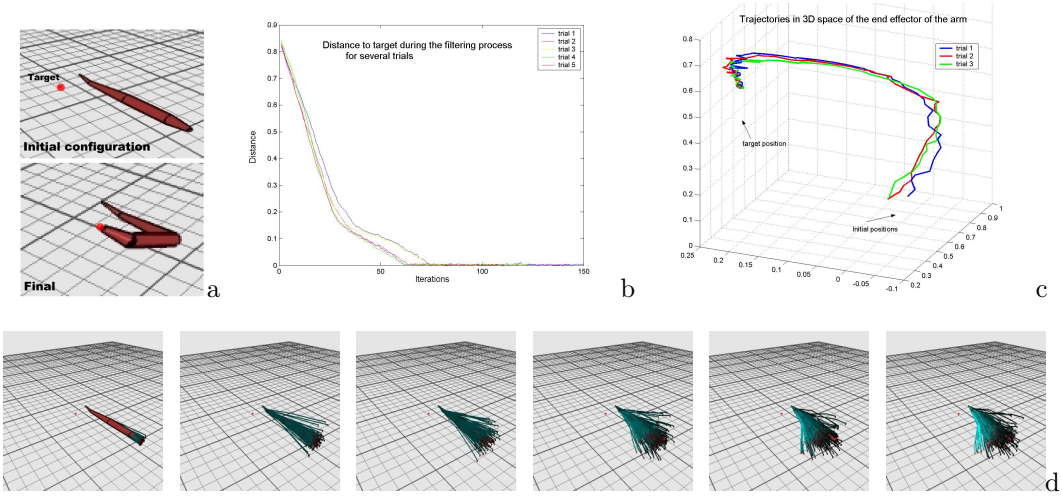


Figure 2: **Convergence of the algorithm** (a) 3D view of the articular chain and the target (b) Distance to target. One can observe that the nature of the convergence does not change along the different trials (c) Trajectories of the end effector in the $3D$ space. **Importance sampling** along 6 consecutive iterations. (d) Each particle is displayed as colored linked segments. This color is a function of the weights: the greener the greater the weight

on a vertical line passing through the root of the chain. We used here another constraint expressed as the following energy function C whose goal is to smooth the overall aspect of the chain (it can be seen as a regularization function):

$$C = \sum_{i=1}^{N} \sum_{j=-k}^{k} ||Log(\mathbf{q}_i^{-1}\mathbf{q}_{i+j})|| \tag{16}$$

This function amounts to minimizing the differences (using geodesical distance) between successive rotations in the chain. This example ran at an approximate speed of 60 frames per second. Figure 3 shows the initial and final configuration.

6.2 Computational performances

From a computational point of view, there are two major questions that arise in our framework: what is a good number of particles for the simulation to be correct and how does this method compare to classical numeric inversion schemes? In Figure 4. We investigated this first issue in the following setup: given a kinematic chain whose number of degrees of freedom is parameterizable, we performed a simple positioning task one thousand times, and reported the percentage of success. In this case, the task was considered successful if

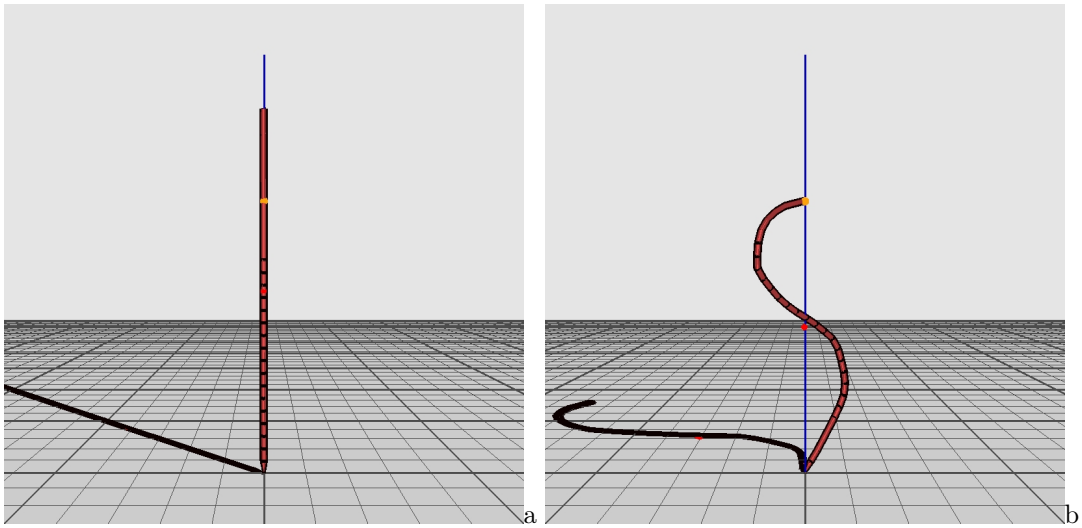
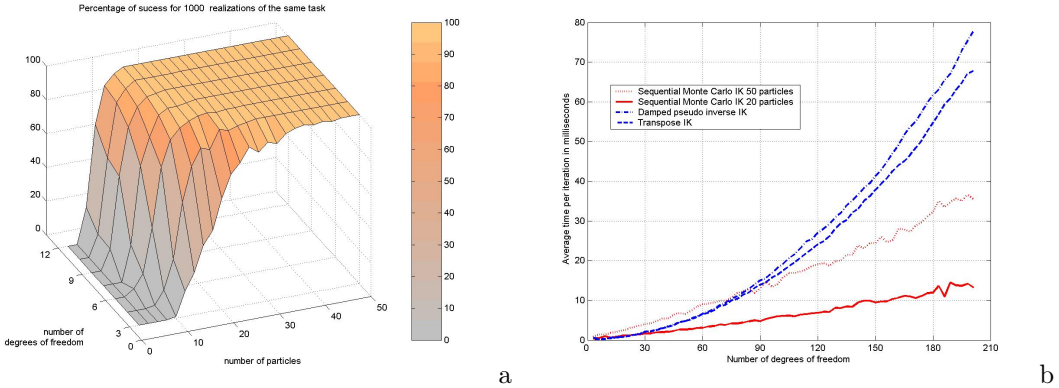


Figure 3: **Chain example**. This chain is composed of 30 segments with 3 rotational degrees of freedom each (a) Initial configuration (b) final configuration.

the distance between the end effector and the target fell down below a certain threshold within a given number of iterations (in our case this threshold was 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 trials is almost always 100 percent, which gives one an idea of the minimal number of particles needed for a fairly robust utilization. Another interesting issue which is rather counter-intuitive is that this minimal percentage diminishes with the number of degrees of freedom (10 particles for almost 100 percent of success for 12 degrees of freedom). One can explain this strange behavior by the fact that when some degrees of freedom are added to the system, redundancy also increases, and the size of the solution state space as well, so that particles are more likely to be solutions of the task.

In Figure 4.b we compared the average time per iteration of two numerical IK solutions (the Jacobian transpose method and the damped pseudo-inverse methods as proposed in [19]) and our method parameterized with 50 and 20 particles. In both numerical methods the Jacobian was evaluated with a finite difference scheme (which reveals to be 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 a structure with a high number of DOFs. One can also notice the noisy aspect of the performance curves of the Sequential Monte Carlo methods. This is due to the random nature of the acceptance/rejection technique that ensures the enforcing of joint limits.



a b

Figure 4: **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. One can can effectively see the linear nature of the complexity of our method (instead of exponential with numeric IK).

6.3 Human figure

In this example we consider a complete human figure with 40 joints that were designed to respect predefined kinematic properties (number of degrees of freedom and joint limits). Snapshots of a resulting animation are shown in Figure 5. For this example, we added to the state space the root position (the pelvis in our case) so that the whole figure can move in the $3D$ space. For the cartesian coordinates of this link, an additive Gaussian noise was applied (conversely to the multiplicative quaternionic noise presented in the previous section). The feet were constrained to stay on the floor, while the left and the right arms were given two different targets. It is interesting to notice in the produced animation how the motions of root of the body contribute to the solution.

6.4 Hand animation

In this example the considered chain is a forearm with a hand. The elbow, as well as the wrist, have been given 3 DOFs. The fingers are constituted of 3 segments. The basis of each fingers has 2 DOFs allowing abduction/adduction and flexion/extension. The remaining joints are set to have only one degree of freedom. In this animation, each fingertip is given a target (empirically determined). Two sets of targets are chained together during the animation. Figure 6 shows images from this animation. In order to increase the realism of the produced animation, we added the biomechanical constraint linking the last two joints of each fingers except the thumb:

$$\theta_{last} = \frac{2}{3}\theta_{previous}$$

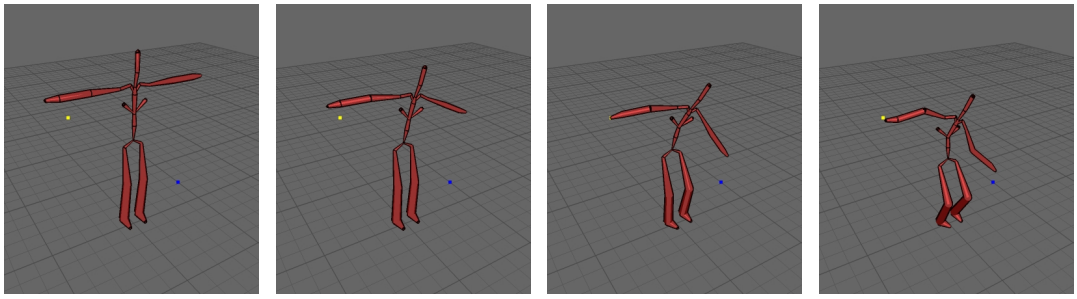


Figure 5: **human figure animation** In this animation, feet are constrained to lie on the floor, the right hand is linked with the yellow dot while the left arm has the blue dot as target. Notice how the knees bend for the task to be achieved

where θ stands for the flexion/extension angle. At this point, let us note the difficulty of handling such a kinematic configuration and the previous constraint in a classical numerical inverse kinematics scheme where this problem would be decomposed into several problems (corresponding to several distinct linear chains) with likely conflicting solutions. Conversely with our framework this problem is treated as a global optimization problem.

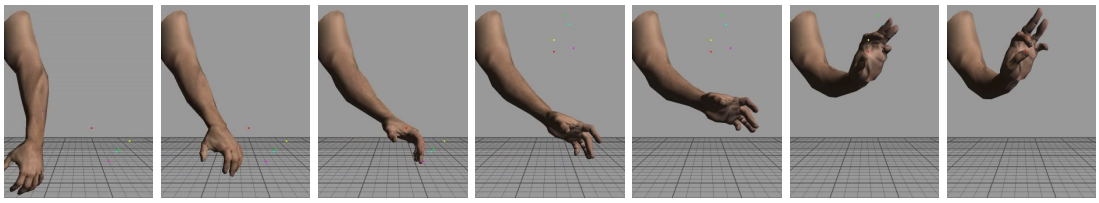


Figure 6: **Hand animation** In this animation the fingers were given a target position represented as colored dots in the images. The two strips correspond to two different tasks that were chained along the animation.

6.5 Limitations

The different experiments of our framework have shown the following limitations:

- Though the type of the resolution for the inverse kinematics problem is different from an analytical or numerical scheme, its intrinsic nature remains local, *i.e.* it is possible to fall into local minima. Hence, the visual quality of the solution may depend on how close the initial configuration is to the solution (conversely to global space-time optimization methods),

- The nature of the noises and their parameters impact in a significant manner on the quality of the results, especially when a lot of constraints are considered. Tuning those parameters to accomplish a given effect has revealed to be sometimes tricky,
- Finally the random nature of the resolution gives a different solution at each execution. This also means that it is possible to figure out a worst case where a solution may not be found. Practically, this has never been observed in our experiments for a reasonable amount of particles.

7 Conclusion and future work

This article has introduced the use of Sequential Monte Carlo Methods in the context of computer animation and proposed a new insight to the resolution of the inverse kinematics problem. Our inverse kinematics filter is very simple to implement and test. It runs very fast and provides a totally new and original alternative to solve this classical problem. We also think it can be successfully used in the protein loop closure problem. As experimental validations of our model we have tested several situations ranging from simple positioning task to hand animation with convincing results. Future works will be done in two main directions:

Adding constraints. Adding kinematic constraint during the exploration stage is a very interesting direction of work since it could reduce the search space, reduce the number of needed particles and increase the result quality. We are studying the use of truncated Gaussian or even more complex non-symmetric probability functions (like the log-normal distribution).

Using motion priors and more complex evolution models. In this article an extremely weak evolution prior was used by assuming a constant configuration over time. Our goal was to to show that, even with such a simple assumption, it is possible to obtain interesting results. It is clearly reasonable to envisage inserting a priori knowledge about the motion here. Heuristics models are first to be considered. We will subsequently investigate the use of motion capture data to design the evolution model (eventually using auto-regressive models as suggested in [6]) as well as physics evolution model. This could lead to efficient online motion retargetting or control algorithms.

References

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particles filters for online nonlinear / non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[2] P. Baerlocher and R. Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20(6):402–417, 2004.

[3] R. Boulic, R. Mas, and D. Thalmann. A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics*, 20(5), 1996.

[4] B. Le Callennec and R. Boulic. Interactive motion deformation with prioritized constraints. *Graphical Models*, 68(2):175–193, 2006.

[5] S. Carvalho, R. Boulic, and D. Thalmann. Interactive Low-Dimensional Human Motion Synthesis by Combining Motion Models and PIK. *Computer Animation & Virtual Worlds*, 18, 2007. Special Issue of Computer Animation and Social Agents (CASA2007), to appear.

[6] J. Chai and J.K. Hodgins. Constraint-based motion optimization using a statistical dynamic model. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 26(3), July 2007.

[7] K.J. Choi and H.S. Ko. Online motion retargetting. *Journal of Visualization and Computer Animation*, 11:223–235, 2000.

[8] N. Courty, E. Marchand, and B. Arnaldi. Through-the-eyes control of a virtual humanoïd. In *Proc. of Computer Animation 2001*, pages 74–83, Seoul, South Korea, November 2001.

[9] C.Sminchisescu and B.Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[10] C.Sminchisescu and B.Triggs. Fast mixing hyperdynamic sampling. *Image and Vision Computing (Special Issue on Selected Papers from ECCV02 Conference)*, 2006.

[11] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005.

[12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag, Series Statistics for Engineering and Information Science, 2001.

[13] S. Gibet and P.F. Marteau. A self-organised model for the control, planning and learning of nonlinear multivariable systems using a sensori- feedback. *Journal of Applied Intelligence*, 4:337–349, 1994.

[14] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 23(3):522–531, August 2004.

[15] M. P. Johnson. *Exploiting quaternions to support expressive interactive character motion*. PhD thesis, Massachusetts Institute of Technology, 2003.

[16] K.P. Körding and D.M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427:244–247, 2004.

[17] A. Maciejewski. Dealing with the ill-conditioned equations of motion for articulated figures. *IEEE Computer Graphics and Applications*, 10(3):63–71, May 1990.

[18] J-S. Monzani, P. Baerlocher, R. Boulic, and D. Thalmann. Using an intermediate skeleton and inverse kinematics for motion retargetting. *Computer Graphics Forum*, 19(3), 2000. ISSN 1067-7055.

[19] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measures and Control*, 108:163–171, September 1986.

[20] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4), 2005.

[21] S. Tak and H.-S. Ko. A physically-based motion retargeting filter. *ACM Tra. On Graphics (TOG)*, 24(1):98–117, 2005.

[22] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[23] E.A. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control*, 2000.

[24] C. Welman. Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Frasier University, September 1993.

[25] W. A. Wolovich and H. Elliot. A computational technique for inverse kinematics. In *Proc. of 23rd IEEE Conf. on Decision and Control*, pages 1359–1363, 1984.

[26] K. Yamane and Y. Nakamura. Natural motion animation through constraining and deconstraining at will. *IEEE Tra. on Visualization and Computer Graphics*, 09(3):352–360, 2003.

[27] J. Zhao and N. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 13(4):313–336, 1994.

Appendix A: Sampling Quaternions along a QuTem distribution

Here, the definition given in [15] of a QuTem density is adopted. Such a distribution can be seen as an approximation of a Gaussian distribution on S^3. The identity-mean QuTem probability density is given by:

$$\phi(\mathbf{q}; \mathbf{1}, \Sigma) = k \; exp\left(\frac{-\langle log(\mathbf{q})|\Gamma|log(\mathbf{q})\rangle}{2}\right), \tag{17}$$

where k is a normalization constant, $\mathbf{1}$ is the identity quaternion (pure real), and Γ is a symmetric matrix called the concentration matrix of the distribution (usually $\Gamma = \Sigma^{-1}$ if Σ is the covariance matrix of the distribution). $|log(\mathbf{q})\rangle$ denotes the vector corresponding to the pure quaternion $log(\mathbf{q})$ and $\langle log(\mathbf{q})|\Gamma|log(\mathbf{q})\rangle$ is the quadratic form defined by Γ. Sampling over such a distribution can be done following the next algorithm:

Algorithm 2 Sampling from $\phi(\mathbf{q}; \mu, \Sigma)$

1. $\sigma_1^2, \sigma_2^2, \sigma_3^2 \leftarrow$ eigenvalues of Σ
2. generate $x, y, z \sim \mathcal{N}(0, 1)$
3. normalize vector (x, y, z)
4. $N \leftarrow (\sigma_1^2 x, \sigma_2^2 y, \sigma_3^2 z)$
5. generate $\theta \sim \mathcal{N}(0, \sigma^2)$
6. $\mathbf{q} = \mu \exp[N\theta]$
7. **return q**

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399