HAL
open science

Kinematic driven by distances

Thibaut Le Naour, Nicolas Courty, Sylvie Gibet

To cite this version:

Thibaut Le Naour, Nicolas Courty, Sylvie Gibet. Kinematic driven by distances. 2018. hal-01838784

HAL Id: hal-01838784

https://hal.science/hal-01838784v1

Preprint submitted on 13 Jul 2018

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.

Kinematic driven by distances

1st Thibaut Le Naour
IRISA-UBS
University of Bretagne Sud
56000 Vannes, France
tlenaour@gmail.com

2nd Nicolas Courty
IRISA-UBS
University of Bretagne Sud
56000 Vannes, France
courty@univ-ubs.fr

3rd Sylvie Gibet
IRISA-UBS
University of Bretagne Sud
56000 Vannes, France
City, Country
sylvie.gibet@univ-ubs.fr

***Abstract*—This paper deals with a new method for controlling kinematically articulated systems. Unlike traditional approaches, we pose the problem in a new distance space. We propose a novel distance-based formalization of kinematic control, and an optimization method for solving a set of classical for motion adaptation and inverse kinematics problems. The originality of the method lies also in the possibility to introduce distance constraints with priorities. The approach is validated by a set of experiments using several classical kinematic operations applied to motion control of articulated figures, and compared to other approaches by means of stability, convergence and performance issues.**

***Index Terms*—Character Animation, Kinematic, Inverse kinematic, Motion editing, Geometrical constraints**

I. Introduction

Traditionally, a character's animation is represented by a series of skeletons over time. A skeleton is a hierarchy of joints between which distances are fixed. In inverse kinematics problem, the aim is to move an effector (typically the end of the chain) to a point $\boldsymbol{x}$. The animated chain is represented by a vector $\boldsymbol{q}$ of positions and orientations of the joints which describes the skeleton. In a direct kinematic problem, for a chain with n joints, we define the function $f : SO(3)^n \rightarrow \mathbb{R}^{3n}$ such that :

$$\boldsymbol{x} = f(\boldsymbol{q}). \tag{1}$$

Inversely, in a problem of inverse kinematic, we obtain :

$$\boldsymbol{q} = f^{-1}(\boldsymbol{x}), \tag{2}$$

where f is the set of transformations to apply to the chain $\boldsymbol{q}$ to move the effector to bmx. We note some observations about the classic inverse kiniematic :

- the expression of the task is given by the definition of a point $\boldsymbol{x}$ in the space to reach ;
- the previous equation 2 is highly non linear ;
- the systems are generally redundant in relation to the problem to be solved. Indeed, the number of constraints is often considerably lower than the number of unknowns ;
- although treated and solved in the literature, inverse kinematics is limited concerning its resolution in the case of multiple effectors. As stated, the problem reaches limits in terms of complexity to satisfy several target constraints. Thus, to be accomplished, the task is rarely expressed in a global way on the whole skeleton but rather introduced by a set of tasks between which priorities must be expressed.

In our approach, our aim is to characterize a posture not by $\boldsymbol{q}$, but by a set of distances. In this case, the motion of an articulated system is driven by the specification of distance relationships between points of the system. By characterizing motion in the distance space, the skeleton chain is represented by a vector $\boldsymbol{w}$ describing a set of positions associated with these distances. Considering n joints associated with m distances, we can express the problem of direct kinematic with a new function $h : \mathbb{R}^m \rightarrow \mathbb{R}^{3n}$:

$$\boldsymbol{x} = h(\boldsymbol{w}). \tag{3}$$

Inversely, in a problem of inverse kinematic, we use the notation :

$$\boldsymbol{w} = h^{-1}(\boldsymbol{x}). \tag{4}$$

Advantages of the method. As the traditional inverse kinematic, the formulated problem is highly non linear. Our approach will therefore be a variational approach. However, the advantage of the proposed formulation is related to **the manner of defining tasks through distances**. We will see through a set of tests that this new way of expressing tasks greatly simplifies the work dedicated to animation control, notably by providing an implicit specification of contacts (in opposition to an explicit spatial definition of contact zones). The latest studies in this domain [HK09], [HKT10], shows the interest of the community for this problem. Moreover, our system allows to animate all of the skeleton in an unified way, so the the resolution of one or more tasks simultaneously is simple. Finally, we will see that through a decoupling of the invariant information (distances which represent fixed lengths of the skeleton segments) and the information which change over time (the other distances), we give priorities between constraints and thus give a style to the animation.

Outline of this paper. First, we define the problem from the expression of the distances (section 3). In this context, a skeleton is characterized by a set of points, each point expressing itself in relation to the others in terms of distances. Then, we propose a method to solve the problem of direct kinematic and highlight how with this spatial representation of the skeleton, we formulate the problem of inverse kinematic (sections 4 et 5). Finally, we propose a set of tests to validate our method (section 6).

II. Related work

Inverse kinematic and applications Inverse kinematic techniques have been widely studied in recent decades, in a graphic context [Wel93], [ZB94] but especially robotic [WE84], [NH86]. These methods give rise to powerful tools, both to generate new motions, and also as components of more specific methods: adaptation of motions to new constraints [Gle98], [CK00], [MBBT00], [KMA05], [CB06], [HRE*08], reconstitution of postures based on virtual markers [JiCVGHP10], [HSP11], or the reconstruction of motions partially acquired from capture methods [GMHP04], [AL11]. The best known approaches have been the subject of comparative studies as part of the reconstruction of different classes of movement [UPBS08].

However, the difficulty to constraint this kind of problem and the redundancy of the considered articulated chain (due to the hierarchical imbrication of several joints), implies the use of secondary tasks which over-constraint the problem [BMT96], [YN03], [BB04], [SK05]. Among these constraints, we can note the angular limits [CMA01], the control of the balance of the character [BMT96]. Recent studies focused on the imbrication of several levels of constraints [BB04], [CB06], [SK05] using priority levels. Here, the difficulty is to propose a system which couples some antagonist tasks without having *a priori* ideas on the global realization of the instructions.

From the point of view of the resolution of the inverse problem, most of the time, the existing algorithms are based on a descent of gradient exploiting an inversion of a Jacobian numerically evaluated. One of the most effective methods consists in calculating the Moore-Penrose pseudo-inverse matrix associated with a dumping function [Mac90]. However, there are several alternatives to this operation: sensorimotor approaches [GM94], probabilistic approaches [GMHP04] or Bayesian inference [CA08]. Note also the possibility of using information collected from movements captured used as reference poses in the processes of optimisation [GMHP04], [CBT07], [TWC*09], [RB09]. These approaches do not work directly in the space of the poses but rather in latent spaces (obtained by methods as the principal components analysis), which induces an over-constrained problem and gives in an implicit way the correlations which can exist between certain articulations. Note here that in the proposed method, the use of reference poses, defined as a set of distances to be respected, plays a similar role to these given-based approaches.

Other methods propose to solve the problem of inverse kinematics by using, not the space of orientations but that of positions (Euclidean space). Thus, the FABRIK [AL11] method proposes a new algorithm of inverse kinematics, by traversing the chain forwards and backwards in a finite number of iterations. The main interest of this approach lies in the possibility of controlling multi-effector systems but also in its speed of execution. This method, which is expressed as a sequence of local geometric constructions, does not guarantee the optimality of the solution obtained.

In the method called "Particle IK Solver" citehecker08, the character is represented by a set of 3D particles (joint positions). The proposed method is a method of iterative resolution under constraints, which is based on a set of ad hoc preconditions and operates in two distinct phases applied successively to the different parts of the body to be animated. This problem decomposition allows local control and provides flexible and customizable solutions that preserve the style of movement. Ho et al [HKT10] are also interested in a skeleton and mesh representation in the form of a point cloud. But contrary to the two previous approaches, the problem of adaptation of movement (in particular morphological adaptation) is formalised as a global problem of spatio-temporal optimisation. The position variations are calculated from the minimization of a sum of energy quantities (deformation, acceleration, energy stress). Finally, a new IK resolution method, based on distance constraints is proposed in robotics [HR06]. This geometric method consists in parameterizing the system by inter-joints distances, and in reformulating the problem from the resolution of a system of linear inequalities.

Similar to previous approaches, our method represents the animated character from a set of Cartesian positions. As in [AL11] and [HR06], our method allows to manage multiple effectors and graph with cycles. However, our method differs essentially in the IK resolution method. It is an iterative, global, and non-linear optimization process. A set of distances (intra-skeleton and distances to the environment) is separated into two distinct subsets of constraints (primary and secondary) that are directly exploited in the optimization process. In addition, secondary constraints are projected into the kernel of the main task, which is an interesting contribution of the method.

Distances, topologies The originality of our method is based on the use of distance information as a relevant representation space from the point of view of the definition of inverse kinematics tasks. This formulation is inspired by existing work in the field of computational biology, and more particularly in the calculation of molecular conformations [Hav98]. In this context, only the information of distances between the different atoms of the molecules is known (in particular thanks to magnetic resonance techniques). This information allows to deduce the global topology of the molecule. Many methods exist. Either, they are based on a progressive construction of the molecule [DW03], or on operational research methods such as separation and evaluation [MLLT10]. A recent state of the art of these methods is given in [LLMM10]. Note here that we differentiate our study from these studies through the use of a human skeleton, where the number of rigid distances between two joints is much less than the number of couples of joints. In these studies, the general trend is rather reversed in the case of molecules (highly over-constrained problem), this requires more complex optimization procedures integrating uncertainty on distance knowledge.

Geometrical Methods Other approaches, such as spectral analysis of the metric matrix of the point cloud [SS85], are used to determine the coordinates of the points. However, this

method requires knowledge of all the exact distances (respect for triangular inequalities). Finally, the methods introduced in the field of computer-aided design propose solving sytems with geometric constraints such as the distances between two points or the angle between two vectors. The resolution of such systems has given rise to several methods presented by Jermann et al [JMS07]. We note that those using gradient descents (Newton, by homotopy), have the advantages of being fast and give approximate results in the absence of a solution.

III. Kinematic driven by distances : formalization

A. Distance definitions

Our study focuses on the animation of a skeleton in the metric space. Each joint is geometrically represented by a point in $\mathbb{R}^3$. By using distances, we want to describe the 'gap' of a joint with the others joints of the skeleton or with points of the space. More precisely, a skeleton is represented by a graph $G = (\boldsymbol{V}, E, d)$ where V is the set of points (center of joints), E the set of edges which link the points of $\boldsymbol{V}$ and d the set of distances of $\boldsymbol{V} \times \boldsymbol{V} \to \mathbb{R}_+$. We note that d_{ij} is the Euclidean distance between the point $\boldsymbol{x}_i$ and the point $\boldsymbol{x}_j$, for all $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$. We define the spatial structure of this graph by the set of equations :

$$\forall (\boldsymbol{x}_i, \boldsymbol{x}_j) \in E, \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = d_{ij}, \tag{5}$$

where $\|.\|$ is the Euclidean norm. To simplify the notation, we will use in the remain of the paper indifferently the notation $\boldsymbol{x}_i$ to represent the point of V or the vector $\overrightarrow{O\boldsymbol{x}_i}$.

Our problem can be formulated as follows : from a sufficient set of distances, the aim is to find the positions of the points (and therefore joints) in the space $\mathbb{R}^3$.

Each point of V is potentially associated with a number of distances, which depends of the topology of the graph at this point. Note that to infer a point $\boldsymbol{x}_i$ in the space, it is sufficient to dispose of four known points which are distinct two by two, associated with their respective distances with the point $\boldsymbol{x}_i$. This means that in some cases we can simplify the problem posed in equation (5).

Numerically, solve such equations is hard. As we saw before, there are several methods. The methods using a spectral approach respond well at this problem but are practical only when all distances are known and accurate (respect of triangular inequalities). We choose to use optimization methods that enable to have close results in the cases where the distances are not accurate, or to reduce the space of solutions if all of distances are not known.

B. Formulation par un problme de minimisation

To solve the previous problem using optimization methods, we reformulate the equation (5) by a minimization problem of energy W such that :

$$W = arg \min_{\boldsymbol{X}} \sum_{E} (\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - d_{ij}^2)^2, \tag{6}$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ is the vector of unknown to determine, with $n = |V|$ and $m = |E|$.

Note that in the equation (6), we perform a quadratic penalization of the error. The advantage is to guarantee a positive error for each equation, which implies that several errors can not cancel each other. This point improve the stability of our algorithm.

We note also that the equation (6) is a system of equation non linear.

We will call in the remain of this paper *constraints*, the rules that the system has to follow We will call in the following article *constraints*, the rules that must satisfy feasible solutions of the problem. Among these rules, the preservation of distance 5 (constraint of distance) or the equality between a point and an other ($g(\boldsymbol{x}) = \boldsymbol{x}_T$). Apply to a skeleton, the aim of the problem is to associate for each joint a sufficient count of constraints.

In order to describe more precisely the optimization method, we discuss in the next section only about the distance constraints concerning the points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$. These constraints can be written by a system of equations F of m objectives functions where each function takes equal distance between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as :

$$f_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - d_{ij}^2)^2 = 0. \tag{7}$$

Thus, we want to solve the system of equations $F(\boldsymbol{X}) = 0$, where $F = (f_{11}, ..., f_{1n}, ..., f_{mn})^T$ is the vector of objectives functions and $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)^T$ is the vector of unknown variable.

IV. Methods of optimization dedicated to the resolution of system of equations

The system of equations being not linear so we use a gradient descent approach as the Newton method. This method has the advantage to be fast and sufficient stable for our problem. Indeed, this technique introduce small change from a posture to another between each iteration of the algorithm. This is an advantage to produce animation which requires a temporal continuity. Newton's formula applied to our problem is written:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \boldsymbol{J}_F(\boldsymbol{X}_k)^{-1} F(\boldsymbol{X}_k), \tag{8}$$

where $\boldsymbol{X}_k$ is the vector $\boldsymbol{X}$ obtained at the iteration of the optimization k and $\boldsymbol{J}_F(\boldsymbol{X}_k)$ is the Jacobian matrix of $F(\boldsymbol{X}_k)$ for the point $\boldsymbol{X}_k$.

The Jacobian matrix $\boldsymbol{J}$ stacks the partial derivatives of all constraints of the problem F. The size of $\boldsymbol{J}$ is $m \times n$ ($m \times n$ constraints for $n \times 3$ unknown) :

$$\boldsymbol{J}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \begin{pmatrix} \frac{\partial f_{11}}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial f_{1n}}{\partial \boldsymbol{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{m1}}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial f_{mn}}{\partial \boldsymbol{x}_n} \end{pmatrix}. \tag{9}$$

with $\frac{\partial f_{ij}}{\partial \boldsymbol{x}_j} = (\frac{\partial f_{ij}}{\partial x_j} \frac{\partial f_{ij}}{\partial y_j} \frac{\partial f_{ij}}{\partial z_j})^T$, x_j, y_j and z_j being the coordinates of the point $\boldsymbol{x}_j$.

We note that the previous equation gives us a general definition of the problem and hat it does not necessarily lead to a number of functions equal to the number of unknown

variables. Indeed, the size of the system depends on the number of constraints given by the user. So we can obtain under-determined problems similar to standard inverse kinematics, with more unknowns than constraints to respect. In contrast, our method simply adds distance constraints reducing the space of solution, until obtaining over-constrained problems (more constraints than unknowns) . Mathematically, these situations lead to Jacobian matrices that are not necessarily square , or of full rank . We use the pseudo-inverse matrix (or inverse of Moore-Penrose) $\boldsymbol{J}^{+}$ which is a generalization of the inverse matrix characterizing any matrix $m \times n (m > n)$ such that $\boldsymbol{J}^{+} = (\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J})^{-\boldsymbol{1}}\boldsymbol{J}^{\boldsymbol{T}}$. The calculation of the pseudo-inverse is performed by the method of singular values decomposition (SVD), and the evaluation of the Jacobian matrix is performed through the analytical expression of the gradient of the distance constraint between two points. So for a matrix element $\boldsymbol{J}$ evaluated at points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in $\mathbb{R}^3$, the gradient is written:

$$\frac{\partial f_{ij}}{\partial \boldsymbol{x}_j}(\boldsymbol{x}_i, \boldsymbol{x}_j) = -4.(\boldsymbol{x}_i - \boldsymbol{x}_j).f_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{10}$$

V. Motion adaptation and inverse kinematic

To solve problems of pure inverse kinematics, the model described above is fully satisfactory. If we consider an articulated chain fixed at one end, knowing the initial posture and a target to be reached by the other end, it is enough to give the system the constraints of distance between the adjacent joints of the skeleton.

The problem thus formulated allows us to replay a movement or to solve the inverse kinematics by minimizing iteratively at each key position of the animation the various constraints of distance. In order to adapt a movement, it is necessary to introduce new concepts and improve the previous optimization method. We can differentiate two types of distance constraints (figure reffig:constraintsHardSoft):

- **Hard constraints:** these are the constraints that must be imperatively respected. For example, they may be invariant distances over time which are the distances between adjacent joints in the skeletal hierarchy. They correspond to the segments. These are also the target constraints of inverse kinematics, such as the contact of a foot with the ground represented by a zero distance between the foot and a plane. For these constraints the functions must be null at the end of the iteration of the optimization loop.
- **Soft constraints:** these are the other distances in the graph. In the context of direct kinematics, they change over time and characterize the animation.

The introduction of these constraints is motivated by the need to give priority to certain constraints over others. In the context of motion adaptation, we rely on a reference motion calculated in the space of distances to which we add a task via hard constraints. This induces a conflict between the constraints: the length of the segments as well as the task added are constraints to be satisfied in priority

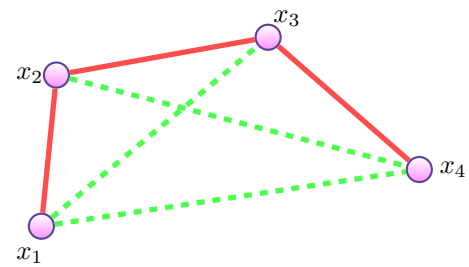

Fig. 1. The dotted segments represent the soft distance constraints while the other segments represent the hard distance constraints.

(hard constraints), whereas the distances which characterize the spatial positioning of the joints of the reference posture constitute properties, that we want to preserve at best (soft constraints). Thus, the system defined is finally over-constrained. We therefore propose a modification of the Newton algorithm which consists in adding a secondary task (soft constraints) which is projected in the solutions space of the main task (hard constraints). Similarly to the formulation proposed by Boulic et al [CB06], this allows incorporating hard and soft constraints into the same algorithm, by taking account of the priorities between constraints. Our algorithm becomes:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \boldsymbol{J}_0^{+} F(\boldsymbol{X}_k) + (\boldsymbol{I} - \boldsymbol{J}_0^{+}\boldsymbol{J}_0)\boldsymbol{J}_1^{+} G(\boldsymbol{X}_k) \tag{11}$$

whre $\boldsymbol{X}_k$ is the vector of positions of joints to determined. To simplify the notations, $\boldsymbol{J}_{F_0}^{+}(\boldsymbol{X}_k)$ and $\boldsymbol{J}_{F_1}^{+}(\boldsymbol{X}_k)$ are written $\boldsymbol{J}_0^{+}$ and $\boldsymbol{J}_1^{+}$. $F(\boldsymbol{X}_k)$ and $G(\boldsymbol{X}_k)$ respectively represent the objective functions associated with hard and soft constraints, and $\boldsymbol{J}_0^{+}$ and $\boldsymbol{J}_1^{+}$ are the pseudo-inverse of the associated Jacobians. $(\boldsymbol{I} - \boldsymbol{J}_0^{+}\boldsymbol{J}_0)$ is an operator of projection that guaranteesthe realization of $G(\boldsymbol{X}_k)$ will not affect the realization of $F(\boldsymbol{X}_k)$.

VI. Results

In order to validate our method, we test it on several case studies. Thus, we highlight the convergence of the method and the importance of defining the tasks to be accomplished. First of all we are interested in the task of motion tracking which has for objective to replay an animation. In a second step, we experiment our method on inverse kinematic situations, notably by comparing the results with those obtained by a traditional kinematic method. Finally we test our method on a motion adaptation experiment. Our tests are all performed in real time (10 frames per second for the slowest animation). They are performed by a single process with a processor running at 2.3 GHz. We tested and used two mathematical libraries ([Dav06]), [GJ*10]) to validate our results.

A. Replay

In order to validate the ability of our method to calculate an animation in distance space, our first test is an example of replay, which represents the reproduction of a S source motion (captured data) to a C target motion for identical skeletons. We define four points two by two distinct in 3d space (canonical

basis and origin). For each time step, after calculating the joint positions of S, we extract for each skeletal joint S the distances with these four points. Finally we incorporate these distances into our system. More precisely, for a skeleton with n joints, we define a problem of $4n$ constraints for $3n$ unknown. Our test is performed on a walking motion stored in *bvh* format from the CMU database.

In order to validate our method, we evaluate the convergence of the algorithm, and we show (figure 3) the real-time follow-up of the target animation. The figure 2 clearly shows the success of convergence. More precisely, we can deduce that the minimization of errors for each constraint at the end of 10 iterations is equal to zero (remember that the minimization was squared). This results correspond to zero distance between the joints of S and C, and visually by an identical follow-up of the animation C on S (figure 3).

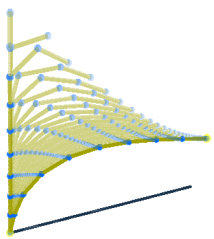

Fig. 2. Convergence results by Newton iteration (transition from the reference pose to the first pose of the movement)

Fig. 3. Illustration of motion reproduction by tracking. Left figure: reference posture (left), initial posture (right). Right figure: the reference movement (left) and the calculated movement (right).

This experiment also shows the robustness of the method: indeed, as illustrated in figure 3, the optimization algorithm can adapt to large variations in initial values (posture very different from the reference posture). Character animation has the advantage of having a reference pause (bind pose) that introduces an initial vector into the iteration of Newton already very close to the solution.

B. Inverse kinematic

1) Rich a target with an articulated chain: Our second test (figure 4) concerns a simple example of inverse kinematics. The goal is that the extremity of an articulated chain bme_1 reaches a target bmc, the other extremity bme_2 being attached to another point bmp. For this experiment, we construct an articulated chain consisting of 10 joints, each of them spaced with the same distance from the adjacent joint. The problem is: we give to the system the 9 constraints specific to the articulated chain plus 2 constraints: the distances between bme_1 and bmc and between bme_2 and bmp are zero. The question here is to highlight the good respect of the constraints (11 constraints for 3×10 unknown).

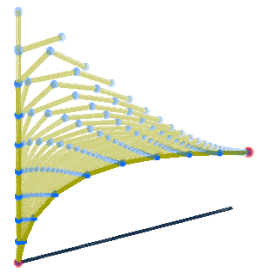

Fig. 4. Illustration of the articulated arm test. The added constraints are illustrated by the red spheres.

The results show first the success of the experiment with constraints respected (the error is zero after 18 iterations). Beyond the respect of these constraints, the experience shows that our method is able, illustrated on a simple example, to solve the same tasks as those solved by the traditional inverse kinematics.

However, our method has the disadvantage of being slower than standard inverse kinematics methods. Indeed, as illustrated in the figure 5, we compared the average calculation times of an iteration relatively to the number of articulations of a chain for our method and for the standard inverse kinematics. The time differences between the methods are explained by the fact that our remote based on the Jacobian matrix presents with this implementation, a dimension higher than the classical Jacobian matrix (orientation based. This results induce a longer inversion calculation time. Note however that our matrix is strongly sparse and that we can plan, to accelerate the inversion of this type of matrix, to use methods such as the one proposed by M. Berry [Ber92]. The acceleration of our method in this context is implemented and discussed in the section 6.3. Also note some irregularities in the distance-based curve, due to variations in processor load.

In the case of standard inverse kinematics, singularity problems lead to the use of a damping matrix. The results obtained are highly dependent on the correct estimation of the coefficients in this matrix. It is interesting to note that our method gives a regular configuration of the chain without damping parameter (see figure 6). This is explained by the fact that the minimization is calculated in the space of distances, which is equivalent to expressing variations across the Jacobian along the axis between two constrained points, unlike the standard inverse kinematics which applies these variations on the 3 axes of rotation of each joint.

2) Application of the contact between a hand and a sphere: Through this experience, we want to highlight the method's

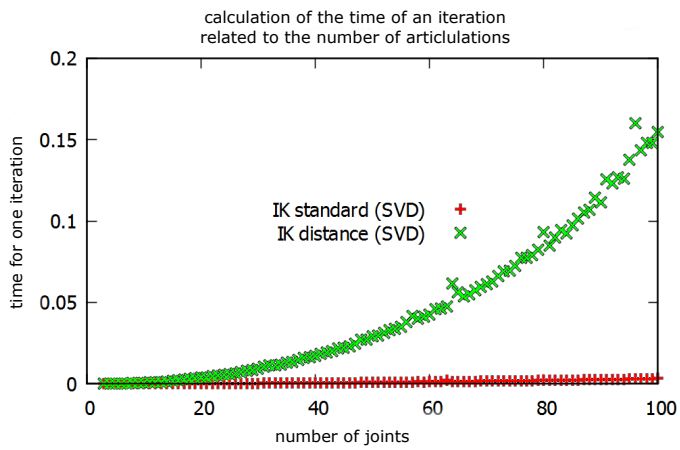

Fig. 5. Average calculation time of an iteration relatively to the number of joints for distance-based methods (in green) and traditional inverse kinematics methods (in red).

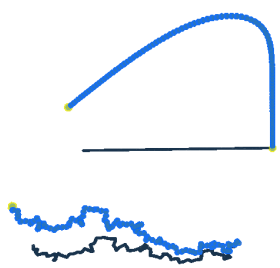

Fig. 6. Illustration for two identical chain. Bottom: calculated by the standard inverse kinematics with an arbitrary damping factor. Above: calculated by our method.

ability to simply express tasks in the form of distances, as well as to implicitly manage a large number of constraints associated with a certain level of priority.

With traditional inverse kinematics, it is also possible to specify complex tasks in the form of multiple goals to be achieved, characterized by geometric constraints of different kinds [PZB90]. This type of approach can be broken down into two phases:

- First, the constraints must be converted into goals to be achieved,
- then to solve the inverse kinematic in the angular coordinate space.

Our method does not need the conversion phase. It allows to specify the task directly in the form of a set of properties to respect. These properties are implicitly integrated in the Jacobian matrix, and the optimization algorithm seeks to find a convergent solution in the Cartesian coordinate space.

To illustrate this approach, we implemented an experiment (see figure 7) involving the right hand of a skeleton that reaches a sphere. This action is defined from the simple specification of a set of distances, without knowing precisely the points to reach. The experiment is performed with a skeleton including detailed hands. Starting from a key reference posture, the objective is to bring all the joints of the right hand to touch the sphere placed in the 3D scene. Beyond the lengths of the segments, we considered hard constraints that express the task through the contacts between the 26 joints of the hand and the sphere. Thus, any articulation of the hand in contact with the sphere leads to a distance constraint that is expressed as an objective function of the same type as (7).

This example shows the advantage of incorporating priorities between constraints, through hard and soft constraints (secondary tasks). Without priorities between constraints, the problem is over-constraint, with the risk that there will be no solution. It is thus possible, by adding or removing constraints, to influence the solutions to the inversion problem. If, for example, the secondary task is removed, it gives the system more latitude to reach the target, but can lead to a potentially unrealistic hand posture.

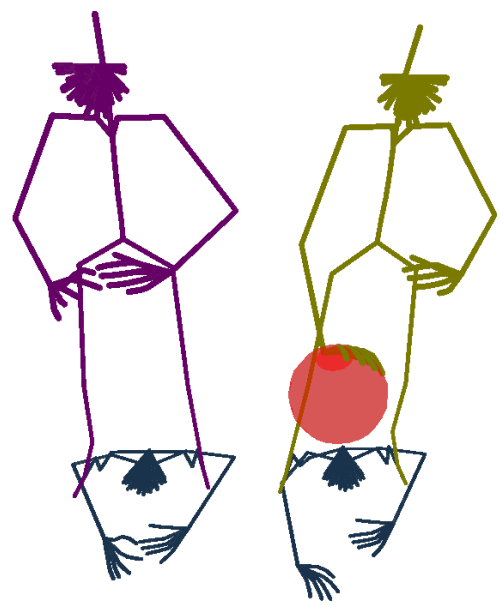

Fig. 7. Sphre atteindre avec la main droite. A gauche la pose de rfrence. A droite le mouvement compos.

C. Motion edition (target to be reached during walking)

Here we highlight the method's ability to modify original movements, by introducing additional tasks. We also show two phases in the animation process. The first concerns the achievement of the objective (positioning of a constraint), the second the relaxation of the constraint which is accompanied by the return of the skeleton towards the initial posture. This experimentation raises the problem of competition between tasks, which requires priority management between constraints.

For this example, we use the same walking motion as in the motion tracking experiment. During his walk, we give as

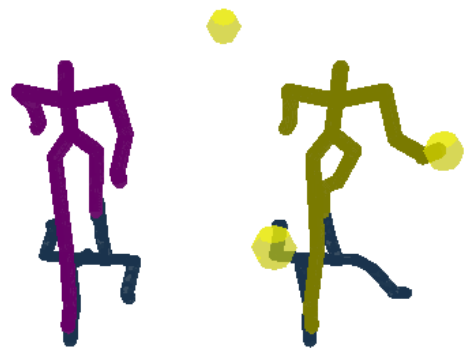

Fig. 8. Illustration de l'adaptation de mouvement. A droite le mouvement rfrence, gauche le nouveau mouvement.

directive to the character to reach successively by the right hand, the left hand and again the right targets that are on his way. Using the same constraints as in the first experiment, we consider the segment lengths as hard constraints and the others as soft constraints (secondary task). The objective of reaching targets during animation is expressed by the temporary addition of hard constraints that take priority over the secondary task.

The figure 9 highlights the conflict between the constraints through an increase in the error of the secondary task. this happens when the targets are reached and the good respect of the hard constraints throughout the animation.

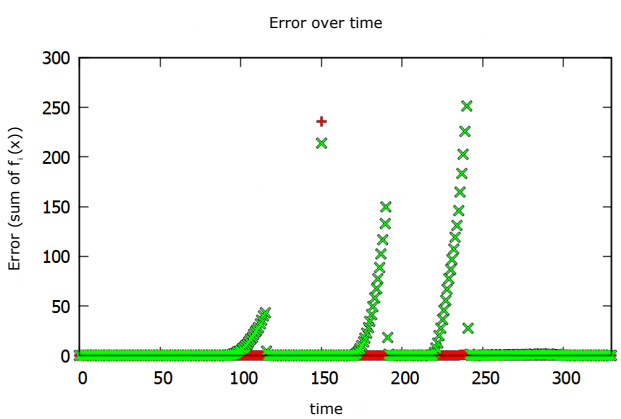

Fig. 9. Evolution of the errors of the main and secondary tasks during the animation. In blue, the error generated by the main task. In red, the one for the secondary task.

Numerical stability First, it is important to note that, contrary to the standard inverse kinematics where the orientation of a joint is dependent on all the parent joints, working on a cloud of points makes it possible to express a point in relation to two neighbours in the simplest case and rarely to more than four or five points for a skeleton (six for a regular mesh). This implies a very low dependence between the points and therefore a more local and more specific distribution of the error related to the objective function. This makes our algorithm more stable. Note, however, that this method leads to a slower convergence. Moreover, over-constraining the problem leads to the possibility of no solution and can therefore lead to a non-continuity between two iterations.

Finally, we added a parameter to Newton's iteration to limit the involvement of the secondary task at the end of the iteration, the secondary task thus serving as a guide towards a solution during the iteration.

Performance The soft constraints are all expressed in relation to four static reference points (which are not part of the unknown vector), which also means that there is no dependence between the three coordinates of a point with all the other coordinates of the unknown vector. This induces that our Jacobian matrix of soft constraints is a diagonal matrix by blocks. However the inverse of such matrix is a diagonal matrix by blocks of the inverses of the blocks. We can thus simplify and finally accelerate the inversion speed of our jacobian by calculating independently the pseudo-inverts of each sub-matrix. If bmx_a, bmx_b, bmx_c and bmx_d are the reference points and bmx_i is an unknown point to be determined based on these four reference points, the expression of the Jacobian becomes:

$$\boldsymbol{J}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \begin{pmatrix} \boldsymbol{J}_1 & 0 & \ldots & 0 \\ 0 & \boldsymbol{J}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{J}_n \end{pmatrix}, \tag{12}$$

avec les matrices diagonales (4×3)

$$\boldsymbol{J}_i(\boldsymbol{x}_i, \boldsymbol{x}_a, \boldsymbol{x}_b, \boldsymbol{x}_c, \boldsymbol{x}_d) = 4 \begin{pmatrix} f_{ia}(\boldsymbol{x}_i, \boldsymbol{x}_a)(\boldsymbol{x}_i - \boldsymbol{x}_a)^T \\ f_{ib}(\boldsymbol{x}_i, \boldsymbol{x}_b)(\boldsymbol{x}_i - \boldsymbol{x}_b)^T \\ f_{ic}(\boldsymbol{x}_i, \boldsymbol{x}_c)(\boldsymbol{x}_i - \boldsymbol{x}_c)^T \\ f_{id}(\boldsymbol{x}_i, \boldsymbol{x}_d)(\boldsymbol{x}_i - \boldsymbol{x}_d)^T \end{pmatrix}. \tag{13}$$

This optimization allows us to obtain significant results: for 31 points to be determined (93 unknown) associated with 124 (4×31) soft constraints and 30 hard constraints, we obtain a rate of 110 frames per second in the optimized case for 51 frames per second in the non-optimized case.

D. Application to a graph with cycles

The last experiment proposes to extend the solution to larger problems such as graphs containing cycles. For this category of applications we used a graph consisting of twenty vertices, which corresponds to twenty hard distances corresponding to the segments between adjacent vertices. We also added two hard distances representing the contact between two targets and two vertices of our graph. Finally, we contrasted two experiments with and without soft constraints in order to visualize the importance of these in the spatial deformation of our graph. As illustrated by the figure 10, the method works well for cycle graphs and allows us to contrast the results with and without soft constraints. This demonstration highlights the fact that through the manner of defining soft constraints, in addition to guaranteeing reference states during the manipulation, it is possible for us to give a style to the deformation of the chain. In particular it is possible to define a style relative to the rigidity of the structure considered.

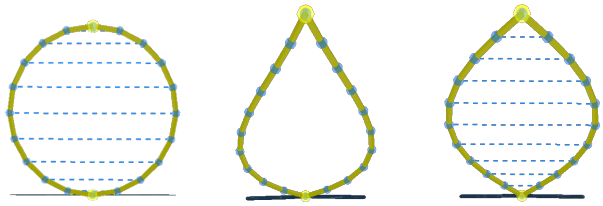

Fig. 10. Application to a cycle graph. Without soft constraint (left figure) and with soft constraints (right figure) : these constraints are represented by the blue dotted lines

VII. Discussion et perspectives

The results obtained demonstrate the ability of the distance-based method to solve kinematic motion control problems. This for diverse tasks from simple tracking of motion captured to solving reverse kinematics problems and modifying motion. The originality of the method lies essentially in the possibility of defining tasks in the form of a set of geometric constraints that express in a simple and relevant way the function to be performed (for example adapting the hand to the object to be grasped). These constraints are expressed in particular in the form of distances which characterize the proximity of the points of the skeleton (distances between joints), or from effectors to effectors or to other points of the environment. Thus, the formulation of the motion control that we propose makes it possible to implicitly take into account a set of constraints (including those related to the task). Incorporating these constraints into optimization methods provides optimal and consistent solutions that are relatively robust with regard to initial conditions and tasks. The general method is extended, in order to separate the contribution of hard distances (main task) and soft distances (secondary task), allowing to introduce a form of priority between constraints. In addition, projecting the solutions of the secondary task onto the kernel of the main task ensures that the main task is respected while reducing the space of the solutions.

The method easily performs motion adaptation tasks from captured data. However, limitations of the method are observed when the specified constraints are too different from those of the original motion, which can lead to a non-convergence of the algorithm, or even to instabilities. The method presented does not deal with the temporal aspects of animations either. Significant improvements would be possible by coupling to spatial functions temporal functions modelling joint synergies[AJG09]. The method could also be extended to handle point clouds from meshes or captured data.

References

[AL11] Aristidou A., Lasenby J.: Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models. Vol. 73*, Num. 5 (2011), 243 – 260.

[BB04] Baerlocher P., Boulic R.: An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer. Vol. 20*, Num. 6 (2004), 402–417.

[Ber92] Berry M. W.: Large scale sparse singular value computations. *International Journal of Supercomputer Applications. Vol. 6* (1992), 13–49.

[BMT96] Boulic R., Mas R., Thalmann D.: A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics. Vol. 20*, Num. 5 (1996).

[CA08] Courty N., Arnaud E.: Inverse kinematics using sequential Monte Carlo methods. In *International Conference on Articulated Motion and Deformable Object* (Port d'andratx, Espagne, 2008), pp. 1–11.

[CB06] Callennec B. L., Boulic R.: Interactive motion deformation with prioritized constraints. *Graphical Models. Vol. 68*, Num. 2 (2006), 175–193.

[CBT07] Carvalho S., Boulic R., Thalmann D.: Interactive Low-Dimensional Human Motion Synthesis by Combining Motion Models and PIK. *Computer Animation & Virtual Worlds. Vol. 18* (2007). Special Issue of Computer Animation and Social Agents (CASA2007), to appear.

[CK00] Choi K., Ko H.: Online motion retargetting. *Journal of Visualization and Computer Animation. Vol. 11* (2000), 223–235.

[CMA01] Courty N., Marchand E., Arnaldi B.: Through-the-eyes control of a virtual humanod. In *Proc. of Computer Animation 2001* (Seoul, South Korea, novembre 2001), pp. 74–83.

[Dav06] Davies R. B.: Newmat c++ matrix library. http://www.robertnz.net/nm_intro.htm, 2006.

[DW03] Dong Q., Wu Z.: A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization. Vol. 26* (July 2003), 321–333.

[GJ*10] Guennebaud G., Jacob B., et al.: Eigen v3. http://eigen.tuxfamily.org, 2010.

[Gle98] Gleicher M.: Retargetting motion to new characters. In *Proc. of the 25th annual conference on Computer graphics and interactive techniques (Siggraph)* (1998), pp. 33–42.

[GM94] Gibet S., Marteau P.: A self-organised model for the control, planning and learning of nonlinear multivariable systems using a sensori- feedback. *Journal of Applied Intelligence. Vol. 4* (1994), 337–349.

[GMHP04] Grochow K., Martin S., Hertzmann A., Popovic Z.: Style-based inverse kinematics. *ACM Tra. on Graphics (Proc. SIGGRAPH). Vol. 23*, Num. 3 (aot 2004), 522–531.

[Hav98] Havel T. F.: Distance geometry: Theory, algorithms, and chemical applications. *Encyclopedia of Computational Chemistry. Vol. 120* (1998).

[HK09] Ho E. S., Komura T.: Character motion synthesis by topology coordinates. In *Computer Graphics Forum* (2009), vol. 28, pp. 299–308.

[HKT10] Ho E. S., Komura T., Tai C. L.: Spatial relationship preserving character motion adaptation. *ACM Transactions on Graphics (TOG). Vol. 29*, Num. 4 (2010), 1–8.

[HR06] Han L., Rudolph L.: Inverse kinematics for a serial chain with joints under distance constraints. In *Proc. of Robotics, Science, and Systems* (2006).

[HRE*08] Hecker C., Raabe B., Enslow R. W., DeWeese J., Maynard J., van Prooijen K.: Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG). Vol. 27*, Num. 3 (2008), 1–11.

[HSP11] Hauberg S., Steenstrup-Pedersen K.: Predicting articulated human motion from spatial processes. *International Journal of Computer Vision. Vol. 94* (2011), 317–334.

[JiCVGHP10] Jaume-i Capó A., Varona J., González-Hidalgo M., Perales F.: Adding image constraints to inverse kinematics for human motion capture. *EURASIP J. Adv. Signal Process. Vol. 2010* (January 2010), 4:1–4:13.

[JMS07] Jermann C., Michelucci D., Schreck P.: Modélisation géométrique par contraintes. In *Informatique graphique, modélisation géométrique et animation*, Bechmann D., Péroche B., (Eds.). Hermés Science, 2007, pp. 185–210.

[KMA05] Kulpa R., Multon F., Arnaldi B.: Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum, Eurographics 2005 special issue. Vol. 24*, Num. 3 (2005), 343–352.

[LLMM10] Liberti L., Lavor C., Mucherino A., Maculan N.: Molecular distance geometry methods: from continuous to

discrete. *International Transactions in Operational Research. Vol. 18* (2010), 33–51.

[Mac90] MACIEJEWSKI A.: Dealing with the ill-conditioned equations of motion for articulated figures. *IEEE Computer Graphics and Applications. Vol. 10*, Num. 3 (1990), 63–71.

[MBBT00] MONZANI J.-S., BAERLOCHER P., BOULIC R., THALMANN D.: Using an intermediate skeleton and inverse kinematics for motion retargetting. *Computer Graphics Forum. Vol. 19*, Num. 3 (2000). ISSN 1067-7055.

[MLLT10] MUCHERINO A., LAVOR C., LIBERTI L., TALBI E.-G.: A parallel version of the branch & prune algorithm for the molecular distance geometry problem. In *AICCSA* (2010), pp. 1–6.

[NH86] NAKAMURA Y., HANAFUSA H.: Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measures and Control. Vol. 108* (septembre 1986), 163–171.

[PZB90] PHILLIPS C., ZHAO J., BADLER N.: Interactive real-time articulated figure manipulation using multiple kinematic constraints. In *Proc. of the 1990 symposium on Interactive 3D graphics (I3D'90)* (1990), ACM, New York, USA, pp. 245–250.

[RB09] RAUNHARDT D., BOULIC R.: Motion constraint. *The Visual Computer. Vol. 25*, Num. 5 (May 2009), 509 – 518.

[SK05] SENTIS L., KHATIB O.: Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics. Vol. 2*, Num. 4 (2005).

[SS85] SIPPL M. J., SCHERAGA H. A.: Solution of the embedding problem and decomposition of symmetric matrices. *Proceedings of the National Academy of Sciences. Vol. 82*, Num. 8 (1985), 2197–2201.

[TWC*09] TOURNIER M., WU X., COURTY N., ARNAUD E., REVERET L.: Motion compression using Principal Geodesic Analysis. *Computer Graphics Forum. Vol. 28*, Num. 2 (mars 2009).

[UPBS08] UNZUETA L., PEINADO M., BOULIC R., SUESCUN .: Full-body performance animation with sequential inverse kinematics. *Graphical Models. Vol. 70*, Num. 5 (2008), 87–104.

[WE84] WOLOVICH W. A., ELLIOT H.: A computational technique for inverse kinematics. In *Proc. of 23rd IEEE Conf. on Decision and Control* (1984), pp. 1359–1363.

[Wel93] WELMAN C.: *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation*. Master's thesis, Simon Frasier University, septembre 1993.

[YN03] YAMANE K., NAKAMURA Y.: Natural motion animation through constraining and deconstraining at will. *IEEE Tra. on Visualization and Computer Graphics. Vol. 09*, Num. 3 (2003), 352–360.

[ZB94] ZHAO J., BADLER N.: Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Tra. on Graphics (Proc. SIGGRAPH). Vol. 13*, Num. 4 (1994), 313–336.