Affine trajectory deformation for redundant manipulators

Quang-Cuong Pham and Yoshihiko Nakamura
Department of Mechano-Informatics
University of Tokyo, Japan

***Abstract*—We propose a new method to smoothly deform trajectories of redundant manipulators in order to deal with unforeseen perturbations or to retarget captured motions into new environments. This method is based on the recently-developed affine deformation framework, which offers such advantages as closed-form solutions, one-step computation and no trajectory re-integration. Satisfaction of inequality constraints and dynamics optimization are seamlessly integrated into the framework. Applications of the method to interactive motion editing and motion transfer to humanoid robots are presented. Building on these developments, we offer a brief discussion of the concept of redundancy from the viewpoint of group theory.**

I. Introduction

Planning trajectories for highly redundant manipulators is challenging and time-consuming because of the large number of degrees of freedom associated with these systems [7]. As a consequence, in order to deal with unforeseen obstacles or perturbations of the target or of the system state, it is sometimes more advantageous to *deform* a previously planned trajectory rather than to re-compute entirely a new one. In motion-capture-based applications, deforming captured trajectories – e.g. to adapt them to a different environment, to retarget them to a different character [9, 2, 6], or to transfer them to a humanoid robot [15, 13] – is the only viable option, as one cannot reasonably record beforehand all the motions with the desired kinematic and dynamic properties.

A. Some desirable properties of the deformations

Our aim in the present article is to design a trajectory deformation algorithm that satisfy the three following requirements : *accuracy*, *smoothness* and *optimality*. "Accuracy" simply means that the deformed trajectory should attain the objective for which the deformation has been conceived : for instance, to reach *exactly* a new configuration, with a specified velocity, acceleration, etc.

By "smoothness", we mean that the deformed trajectory should conserve the *regularity* properties (such as being C^1, C^2 or more generally C^p) of the original trajectory. This is particularly critical for instance in the context of computer graphics to avoid motion jerkiness or, in the context of robot control, to avoid infinite torques.

By "optimality", we mean that one should have the possibility to choose a deformation that *optimizes* certain criteria, so long as this optimization does not interfere with the previous requirements of accuracy and smoothness. One typical objective can be for instance to minimize the distance between the deformed trajectory and the original one. This is obviously desirable in the case of motion retargeting [9, 2, 6] or motion transfer to humanoid robots [15, 13]. In the context of robot control, the original trajectory is often obtained through extensive computations (in order e.g. to minimize energy consumption, to avoid obstacles, to respect joint limits,...), therefore, if the deformed trajectory is "close" to the original one, one can expect the former to keep the good properties of the latter. The deformation algorithm should also leave the possibility for specifying *explicitly* other optimization objectives, such as integrated torque, energy consumption, distance to obstacles, etc.

B. Existing approaches

Two main approaches exist in the literature for handling trajectory deformations. In *spline-based* approaches, a deformation is made by altering the coefficients multiplying the basis splines [12] or by adding to the original trajectory a displacement map – which is a sum of splines [2, 6]. These modifications can furthermore be done in a coarse-to-fine manner using wavelet bases [12] or through hierarchical approximations [6].

Another approach is based on the encoding of the original trajectory by an *autonomous nonlinear dynamical system* [3, 13]. A deformation is then made by altering the coefficients multiplying the basis functions that appear in the definition of the dynamical system. This approach yields a robust execution-time behavior thanks to the autonomous nature of the dynamical system. At the same time, because of this dynamical-system representation, inequality (such as joint limits, obstacle avoidance) and equality (such as specified final velocity, acceleration, etc.) constraints at specific time instants cannot be taken into account without integrating the whole trajectory up to these time instants, which can be very costly.

The above two approaches are similar in that they make use of *exogenous* basis functions : splines in the spline-based approach, and Gaussian kernel functions in the dynamical-system-based approach. A first, pervasive, difficulty then consists of choosing the appropriate bases for a particular task. Second, adding artificial functions to a natural movement can produce undesirable behaviors, such as large undulations in the case of splines [6, 12], lack of smoothness, etc. – which call for supplementary and often costly efforts to correct.

C. Our approach

Unlike the spline-based and the dynamical-system-based approaches, the method we propose makes use of no exogenous basis function. Indeed, inspired by the recent finding that human movements were – to some extent – *invariant under*

affine transformations [1], we deform a given trajectory by applying affine transformations on parts of it. Thus, the only "basis functions" are the original joint angle time series and, as a consequence, any deformed trajectory under this scheme automatically preserves some properties of the original one. Such invariant properties may include for instance smoothness, periodicity, absence of large undulations… or more qualitatively, motion "naturalness" [17], which is a desirable feature in motion retargeting applications, yet difficult to quantify. We shall show that, despite this small functions basis, it is still possible to satisfy the requirements of "accuracy", "smoothness" and "optimality" stated earlier by leveraging the extra redundancy offered by the affine deformation framework.

More precisely, our approach is based on the affine trajectory deformation framework first developed for nonholonomic mobile robots [8]. In contrast with previous trajectory deformation methods for mobile robots [5, 10], affine-geometry-based algorithms are exact (given by closed-form expressions), can be executed in one step, and do not require any trajectory re-integration. The present manuscript focuses on manipulators with many degrees of freedom, but some of the new developments presented here (about e.g. inequality constraints or dynamics optimization) should benefit all classes of system for which affine deformations are applicable.

In section II, we present the main algorithms of affine trajectory deformation for redundant manipulators. Closed-form expressions of the deformations that optimize the closeness between the original and the deformed trajectories are given, and explicit bounds on the difference between the original and the deformed trajectories or between their derivatives are presented. Inequality constraints, dynamics optimization, and kinematics filtering are integrated into the affine deformation framework. In section III, we illustrate these results with two simple concrete examples : interactive motion editing and motion transfer to humanoid robots. In section IV, we give a characterization of *trajectory redundancy* by the group of admissible deformations, revisiting thereby the concept of kinematic redundancy and suggesting a new theoretical approach to motion planning.

II. Smooth trajectory deformations

A. Equality constraints on the deformations

1) Smoothness constraints at the deformation time instant: Consider a C^p trajectory $\boldsymbol{\theta}(t)_{t\in[0,T]}$ and an affine transformation $\mathcal{F}$ that deforms $\boldsymbol{\theta}(t)_{t\in[0,T]}$ into $\boldsymbol{\theta}'(t)_{t\in[0,T]}$ at a time instant τ, i.e.

$$\begin{aligned} \forall t < \tau \qquad & \boldsymbol{\theta}'(t) = \boldsymbol{\theta}(t) \\ \forall t \geq \tau \qquad & \boldsymbol{\theta}'(t) = \mathcal{F}(\boldsymbol{\theta}(t)). \end{aligned}$$

We say that $\mathcal{F}$ is C^p*-preserving* if the resulting $\boldsymbol{\theta}'(t)_{t\in[0,T]}$ is also C^p. From [8], we know that $\mathcal{F}$ is C^p-preserving if and only if (i) $\boldsymbol{\theta}(\tau)$ is a fixed-point of $\mathcal{F}$ and (ii) the first p derivatives of $\boldsymbol{\theta}'$ are continuous at τ. Condition (i) implies that $\mathcal{F}$ is of the form

$$\forall \boldsymbol{\phi} \quad \mathcal{F}(\boldsymbol{\phi}) = \boldsymbol{\theta}(\tau) + \mathcal{M}(\boldsymbol{\phi} - \boldsymbol{\theta}(\tau)), \tag{1}$$

where $\mathcal{M}$ is a linear map $\mathbb{R}^n \to \mathbb{R}^n$.

Next, for $i = 1 \dots p$, let us note $\mathbf{u}_i = \frac{\mathrm{d}^i \boldsymbol{\theta}}{\mathrm{d}t^i}(\tau)$ ($\mathbf{u}_1$ is the velocity vector at time τ, $\mathbf{u}_2$ is the acceleration vector, etc.) Then condition (ii) can be formulated as

$$\forall i = 1 \dots p \quad \mathcal{M}(\mathbf{u}_i) = \mathbf{u}_i. \tag{2}$$

Let us now consider the non-degenerate case when $\mathbf{u}_1, \dots, \mathbf{u}_p$ are linearly independent. In this case, it is possible to construct an orthonormal basis $\mathcal{B}$ whose first p vectors form a basis of $U = \mathrm{Span}(\mathbf{u}_1, \dots, \mathbf{u}_p)$, for instance using a Gram-Schmidt orthonormalization procedure. From equation (2), the matrix of $\mathcal{M}$ in this basis is of the form

$$\mathbf{M} = \begin{pmatrix} 1 & & & m_{1,p+1} & \cdots & m_{1,n} \\ & \ddots & & \vdots & \vdots & \vdots \\ & & 1 & m_{p,p+1} & \cdots & m_{p,n} \\ 0 & \dots & 0 & 1 + m_{p+1,p+1} & \cdots & m_{p+1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & m_{n,p+1} & \cdots & 1 + m_{n,n} \end{pmatrix},$$

which shows that the space of C^p-preserving affine transformations at τ forms a Lie subgroup of the General Affine group GA_n of dimension $n(n-p) = n^2 - pn$, parameterized by the $m_{i,j}$.

2) Constraints on the final configuration: Regarding the "accuracy" requirement, let us now characterize, within the space of C^p-preserving deformations, those that allow attaining a desired final position $\boldsymbol{\theta}_d$, i.e., such that

$$\boldsymbol{\theta}'(T) = \mathcal{F}(\boldsymbol{\theta}(T)) = \boldsymbol{\theta}_d$$

Let $(\lambda_1, \dots, \lambda_n)$ and $(\mu_1, \dots, \mu_n)$ denote respectively the coordinates of $\boldsymbol{\theta}(T) - \boldsymbol{\theta}(\tau)$ and of $\boldsymbol{\theta}_d - \boldsymbol{\theta}(\tau)$ in the basis $\mathcal{B}$. Then, the condition for the deformed trajectory to reach the desired configuration $\boldsymbol{\theta}_d$ at time T reads (see Fig. 1A for an illustration)

$$\mathbf{M} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}. \tag{3}$$

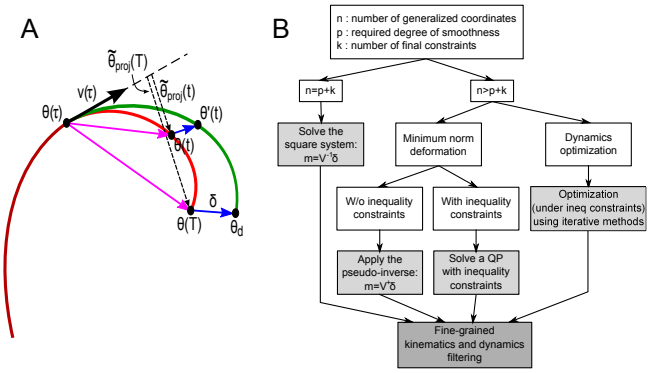

Fig. 1. **A** : Illustration for the case $n = 2$ (system of dimension 2) and $p = 1$ (C^1 continuity, i.e. continuity of the velocity, is required). **B** : Flowchart of the deformation algorithm.

Equation (3) can actually be transformed into

$$\mathbf{V}\mathbf{m} = \boldsymbol{\delta}, \tag{4}$$

where $\mathbf{V}$ is the $n \times n(n-p)$ matrix defined by

$$\mathbf{V} = \begin{pmatrix} \lambda_{p+1} & \cdots & \lambda_n & & 0 & \cdots & 0 \\ & & & & & & \\ 0 & \cdots & 0 & \ddots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & & \lambda_{p+1} & \cdots & \lambda_n \end{pmatrix}, \quad (5)$$

and $\mathbf{m}$ and $\boldsymbol{\delta}$ are the vectors of sizes $n(n-p) \times 1$ and $n \times 1$ defined by

$$\mathbf{m} = \begin{pmatrix} m_{1,p+1} \\ \vdots \\ m_{1,n} \\ \vdots \\ m_{n,p+1} \\ \vdots \\ m_{n,n} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\delta} = \begin{pmatrix} \mu_1 - \lambda_1 \\ \vdots \\ \mu_n - \lambda_n \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \vdots \\ \delta_n \end{pmatrix}.$$

Note from the above definition that $\boldsymbol{\delta}$ represents in fact the coordinates of $\boldsymbol{\theta}_d - \boldsymbol{\theta}(T)$ in the basis $\mathcal{B}$.

3) Constraints on the final derivatives: Assume for instance that one wishes to arrive at the final configuration with a desired velocity $\mathbf{v}_d$. Then, the matrix of the transformation must satisfy, in addition to equation (3), the following equation

$$\mathbf{M} \begin{pmatrix} \lambda_1^v \\ \vdots \\ \lambda_n^v \end{pmatrix} = \begin{pmatrix} \mu_1^v \\ \vdots \\ \mu_n^v \end{pmatrix}, \quad (6)$$

where $(\lambda_1^v, \ldots, \lambda_n^v)$ and $(\mu_1^v, \ldots, \mu_n^v)$ denote respectively the coordinates of $\dot{\boldsymbol{\theta}}(T)$ and of $\mathbf{v}_d$ in the basis $\mathcal{B}$.

Next equation (4) becomes

$$\mathbf{V}^g \mathbf{m} = \boldsymbol{\delta}^g, \quad (7)$$

where $\mathbf{V}^g$ is a $2n \times n(n-p)$ matrix constructed by adding below $\mathbf{V}$ a $n \times n(n-p)$ matrix similar to $\mathbf{V}$ but containing the λ_i^vs instead of the λ_i, and where $\boldsymbol{\delta}^g$ is a $2n \times 1$ vector defined by $\boldsymbol{\delta}^g = (\mu_1 - \lambda_1, \ldots, \mu_n - \lambda_n, \mu_1^v - \lambda_1^v, \ldots, \mu_n^v - \lambda_n^v)^\top$.

More generally, satisfying k final constraints ($k = 3$ for instance if one wants to constraint the final position, velocity, *and* acceleration) will give rise to a matrix $\mathbf{V}^g$ of size $kn \times n(n-p)$. From the dimensions of the matrices of the linear system (7), one can thus draw the following conclusions

- If $n^2 - pn \geq kn$ (i.e. $n \geq k + p$), then it is possible to deform $\boldsymbol{\theta}$ into a $\boldsymbol{\theta}'$ that satisfies k final constraints while guaranteeing C^p continuity.
- Furthermore, if $n > k + p$, then such a $\boldsymbol{\theta}'$ is not unique. In fact, the space of affine transformations that satisfy the above conditions constitutes a Lie subgroup of GA_n of dimension $n^2 - (k+p)n$. Within this space, one can then choose the deformations that *optimize* certain criteria, as detailed in the following sections and summarized in Fig. 1B.

B. Minimum-norm deformations

1) Closed-form solution: As stated in the Introduction, one important optimization objective for the deformed trajectory can consist of being the "closest" possible to the original trajectory. One way to achieve this is to minimize the distance between the transformation $\mathcal{F}$ and the identity transformation, which in turn can be quantified by the Frobenius distance between the matrix $\mathbf{M}$ and $\mathbf{I}$. The Frobenius distance between $\mathbf{M}$ and $\mathbf{I}$ is given by $\|\mathbf{M} - \mathbf{I}\|_F = \sqrt{\sum_{i,j} m_{ij}^2} = \|\mathbf{m}\|_2$, where $\|\cdot\|_2$ denotes the 2-norm of vectors.

On the other hand, the $\mathbf{m}$ of minimum 2-norm that satisfies equation (4) is given by

$$\mathbf{m} = \mathbf{V}^+ \boldsymbol{\delta},$$

where $\mathbf{V}^+$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{V}$ (for simplicity, we treat the case when only the final configuration is constrained).

Observe next that, since the rows of $\mathbf{V}$ are linearly independent, one can in fact compute explicitly $\mathbf{V}^+$ by

$$\mathbf{V}^+ = \mathbf{V}^\top (\mathbf{V}\mathbf{V}^\top)^{-1} = \frac{1}{S}\mathbf{V}^\top,$$

with $S = \sum_{i=p+1}^n \lambda_i^2 = \|\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(T)\|_2^2$, where $\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(T)$ is the orthogonal projection of $\boldsymbol{\theta}(T) - \boldsymbol{\theta}(\tau)$ on $U^\perp$ (see Fig.1A for an illustration). One can thus compute $\mathbf{m}$ by

$$\begin{aligned} \mathbf{m} &= \mathbf{V}^+ \boldsymbol{\delta} = \frac{1}{S}\mathbf{V}^\top \boldsymbol{\delta} \\ &= \frac{1}{S}(\lambda_{p+1}\delta_1, \ldots, \lambda_n\delta_1, \ldots, \lambda_{p+1}\delta_n, \ldots, \lambda_n\delta_n)^\top, \end{aligned}$$

which leads to the explicit form of $\mathbf{M}$ as $\mathbf{M} = \mathbf{I} + \frac{1}{S}\mathbf{W}$, where

$$\mathbf{W} = \begin{pmatrix} 0 & \cdots & 0 & \lambda_{p+1}\delta_1 & \cdots & \lambda_n\delta_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \lambda_{p+1}\delta_n & \cdots & \lambda_n\delta_n \end{pmatrix}. \quad (8)$$

2) Bound on the difference of the trajectories: Let us now compute the distance between the deformed and the original trajectories. First, observe from the form of $\mathcal{F}$ [cf. equation (1)] that, for all $t \geq \tau$

$$\begin{aligned} \boldsymbol{\theta}'(t) - \boldsymbol{\theta}(t) &= (\mathcal{M} - \mathcal{I})(\boldsymbol{\theta}(t) - \boldsymbol{\theta}(\tau)) \\ &= (\mathcal{M} - \mathcal{I})\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(t). \end{aligned} \quad (9)$$

Letting $\|\cdot\|_2$ also denote the induced 2-norm of linear maps and matrices, one has, from the explicit form (8) of $\mathcal{M}$

$$\begin{aligned} \|\mathcal{M} - \mathcal{I}\|_2 = \frac{1}{S}\|\mathbf{W}\|_2 \leq \frac{1}{S}\|\mathbf{W}\|_F &= \frac{1}{S}\sqrt{\left(\sum_{i=1}^n \delta_i^2\right)\left(\sum_{i=p+1}^n \lambda_i^2\right)} \\ = \frac{\|\boldsymbol{\delta}\|_2\sqrt{S}}{S} &= \frac{\|\boldsymbol{\theta}_d - \boldsymbol{\theta}(T)\|_2}{\sqrt{S}}. \end{aligned} \quad (10)$$

Next, from equation (9), one has

$$\begin{aligned} \|\boldsymbol{\theta}'(t) - \boldsymbol{\theta}(t)\|_2 &\leq \|\mathcal{M} - \mathcal{I}\|_2 \cdot \|\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(t)\|_2 \\ &= \frac{\|\boldsymbol{\theta}_d - \boldsymbol{\theta}(T)\|_2}{\sqrt{S}}\|\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(t)\|_2 \\ &= \|\boldsymbol{\theta}_d - \boldsymbol{\theta}(T)\|_2 \frac{\|\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(t)\|_2}{\|\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(T)\|_2}. \end{aligned} \quad (11)$$

Note that inequality (11) is in fact an equality for $t = T$. This inequality provides a bound on the distance between the deformed and the original trajectories at each time instant instant. Since $\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(t)$ and $\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(T)$ do not depend on $\boldsymbol{\theta}_d$,

one can in fact write

$$\max_{t\in[\tau,T]} \|\boldsymbol{\theta}'(t) - \boldsymbol{\theta}(t)\|_2 \leq K\|\boldsymbol{\theta}_d - \boldsymbol{\theta}(T)\|_2,$$

where K is a constant independent of $\boldsymbol{\theta}_d$ (for instance, $K = 1$ in Fig. 1A). In particular, this shows a desirable "continuity" property of our deformation algorithm : when $\boldsymbol{\theta}_d \to \boldsymbol{\theta}(T)$, one has $\boldsymbol{\theta}'(t)_{t\in[0,T]} \to \boldsymbol{\theta}(t)_{t\in[0,T]}$ in $\mathcal{L}_2$.

3) Bounds on the differences of the derivatives: One can also compute the distance between the derivatives (velocity, acceleration, etc.) of the deformed and the original trajectories in a similar way as above. One has indeed, for all levels q of differentiation

$$\forall t \in [\tau, T] \quad \frac{\mathrm{d}^q\boldsymbol{\theta}'}{\mathrm{d}t^q}(t) - \frac{\mathrm{d}^q\boldsymbol{\theta}}{\mathrm{d}t^q}(t) = (\mathcal{M} - \mathcal{I})\left(\frac{\mathrm{d}^q\boldsymbol{\theta}}{\mathrm{d}t^q}(t)\right),$$

which, together with (10), allows obtaining the following bound for all $t \in [\tau, T]$

$$\left\|\frac{\mathrm{d}^q\boldsymbol{\theta}'}{\mathrm{d}t^q}(t) - \frac{\mathrm{d}^q\boldsymbol{\theta}}{\mathrm{d}t^q}(t)\right\|_2 \leq \frac{\|\boldsymbol{\theta}_d - \boldsymbol{\theta}(T)\|_2}{\|\widetilde{\boldsymbol{\theta}}_{\mathrm{proj}}(T)\|_2}\left\|\frac{\mathrm{d}^q\boldsymbol{\theta}}{\mathrm{d}t^q}(t)\right\|_2.$$

Finally, note that the bounds on the trajectory and its derivatives obtained in the previous and this sections are concerned with the global state $\boldsymbol{\theta}'$, and not with the individual angles $\theta'_{j,j\in[1,n]}$.

C. Inequality constraints

In addition to the equality constraints of section II-A, most applications also require the satisfaction of *inequality* constraints, such as joint limits, upper-bounds on the velocity, acceleration or torque, positivity of contact forces, avoidance of obstacles, etc. In many cases, these constraints can be expressed by

$$\mathbf{A}_i\boldsymbol{\theta}'_{Can}(t_i) \leq \mathbf{b}_i, \tag{12}$$

where $t_i \in [\tau, T]$ is a specific time instant, $\mathbf{A}_i$ is a $c\times n$ matrix, $\mathbf{b}_i$ is a $c\times 1$ vector and $\boldsymbol{\theta}_{Can}(t_i)$ is the $n\times 1$ vector containing the coordinates of $\boldsymbol{\theta}(t_i)$ is the canonical basis. To enforce joints limits, one can for example choose several t_i that sample the region where the joint values are expected to be large. Note also that constraints on higher-order derivatives such as $\dot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{\theta}}\ldots$ can be treated using the formulae of section II-A3.

Next, let $\mathbf{P}$ denote the basis matrix of $\mathcal{B}$ (cf. section II-A1). Equation (4) can then be equivalently written as

$$\mathbf{V}\mathbf{m} = \mathbf{P}^{-1}(\boldsymbol{\theta}'_{Can}(t_i) - \boldsymbol{\theta}_{Can}(t_i)),$$

which yields

$$\boldsymbol{\theta}'_{Can}(t_i) = \boldsymbol{\theta}_{Can}(t_i) + \mathbf{P}\mathbf{V}_i\mathbf{m},$$

where $\mathbf{V}_i$ has been constructed as in equation (5). Equation (12) is then equivalent to

$$\mathbf{A}_i\mathbf{P}\mathbf{V}_i\mathbf{m} \leq \mathbf{b}_i - \mathbf{A}_i\boldsymbol{\theta}_{Can}(t_i).$$

One can finally construct a matrix $\mathbf{A}^g$ and a vector $\mathbf{b}^g$ by piling vertically all the $\mathbf{A}_i\mathbf{P}\mathbf{V}_i$ on one hand and all the $\mathbf{b}_i - \mathbf{A}_i\boldsymbol{\theta}_{Can}(t_i)$ on the other hand.

Now, to find the deformation of minimum-norm (cf. section II-B) while satisfying the inequalities (11), it suffices to solve the Quadratic Program

$$\min \frac{1}{2}\|\mathbf{m}\|_2^2$$

with the equality and inequality constraints

$$\mathbf{V}^g\mathbf{m} = \boldsymbol{\delta}^g, \quad \mathbf{A}^g\mathbf{m} \leq \mathbf{b}^g. \tag{13}$$

Note finally that the system of equality and inequality constraints can be fully prioritized and solved efficiently using recent algorithms [4].

As an illustration, consider a planar 3-link manipulator. The original trajectory of the end-effector is a straight line between the initial position and the final position. However, the corresponding joint angle trajectory violates several joint limit constraints. A deformed C^1 trajectory is then computed that connects the initial and final positions while respecting the constraints *and* staying close to the original trajectory, see Fig. 2.

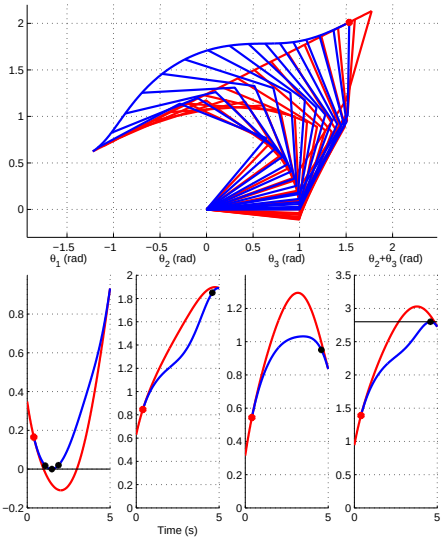

Fig. 2. Deformation under inequality constraints. The original trajectory (red) is deformed into the blue one in order to respect the constraints $\theta_1 \geq 0$ and $\theta_2 + \theta_3 \leq 2.8$ rad (black lines in the bottom plots). Note that the deformed trajectory remains C^1. The time instants when the constraint are enforced (black dots in the bottom plots) were chosen near the minimum of θ_1 and the maximum of $\theta_2 + \theta_3$ in the original trajectory.

D. Optimization of dynamic quantities

As stated in the Introduction, it is sometimes necessary to specify *explicitly* the objective of optimization, such as energy consumption or integrated square torque. However, in contrast with the minimum-norm deformation case, closed-form solutions cannot, in general, be obtained for such optimizations. One must then resort to iterative methods.

Typically, consider a cost function of the type

$$C(\boldsymbol{\theta}) = \int_0^T c(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t), \ddot{\boldsymbol{\theta}}(t))\,\mathrm{d}t.$$

Then the optimization problem can be formulated as

$$\min_{\mathcal{M}} \int_{\tau}^{T} c(\boldsymbol{\theta}(\tau) + \mathcal{M}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}(\tau)), \mathcal{M}(\dot{\boldsymbol{\theta}}(t)), \mathcal{M}(\ddot{\boldsymbol{\theta}}(t))) \, \mathrm{d}t \tag{14}$$

subject to the linear equality and inequality constraints of equation (13).

Note that, if an iterative method is used, a very convenient candidate for a initial guess is the $\mathcal{M}$ of minimum norm, which can be computed very quickly following sections II-B and II-C

As an illustration, consider again the 3-link planar manipulator of the previous section. The deformation was made here in order to minimize the integrated squared torque

$$C(\boldsymbol{\theta}) = \int_0^T \phi_1^2(t) + \phi_2^2(t) + \phi_3^2(t) \, \mathrm{d}t,$$

where ϕ_j is the torque applied at joint j. Note that the above equation can be easily put in the form of equation (14) by expressing the ϕ_j as functions of θ_j, $\dot{\theta}_j$, $\ddot{\theta}_j$ and the geometric and dynamic parameters of the system. An example is given in Fig. 3.

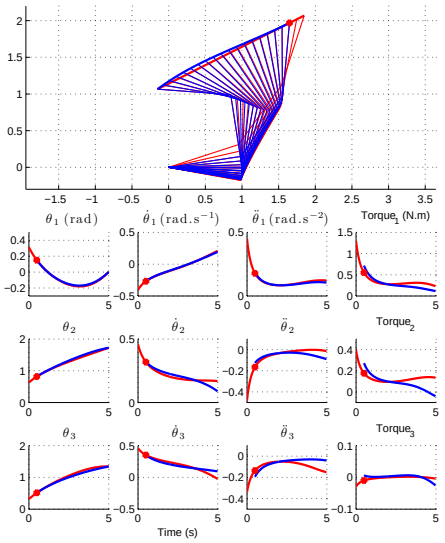

Fig. 3. Minimization of squared torque. The original trajectory (red) is deformed into a new trajectory (blue) whose integrated squared torque is smaller. Note that the deformed joint angle trajectories are C^1 but not C^2. Note also that, in agreement with classical results (see e.g. [14]), the torque-optimal trajectory of the end-effector in Cartesian space is not straight but slightly curved.

E. Fine-grained kinematics and dynamics filtering

In the previous sections, we have presented a method to find trajectory deformations that suit *global, coarse-grained,* objectives, such as reaching a desired final configuration, avoiding obstacles at specific time instants or optimizing a global trajectory cost. However, a trajectory obtained from such deformations may not satisfy *local, fine-grained,* constraints. For instance, when modifying a walking pattern, one needs to make sure that the stance foot neither leaves the ground nor slips. This is expressed by maintaining the foot position constant *throughout a continuous time interval*. It is clear that a single affine transformation cannot satisfy such a constraint that applies to a continuous time interval (neither can any other global approach, including spline- or dynamical-system-based methods [2, 6, 3, 13]).

To take into account this type of fine-grained constraints, we propose to subsequently "filter" the deformed trajectory. More precisely, consider for instance the following constraint

$$\mathbf{g}(\boldsymbol{\theta}(t)) = 0, \quad \forall t \in [t_0, t_1]. \tag{15}$$

Let us denote by $\mathbf{J_g}$ the Jacobian matrix of $\mathbf{g}$ and $N_{\mathbf{g}}(\boldsymbol{\phi})$ the nullspace of $\mathbf{J_g}$ at point $\boldsymbol{\phi}$. One first needs to make sure that the deformed trajectory $\boldsymbol{\theta}'$ satisfies the constraint at t_0 by requiring $\mathbf{g}(\boldsymbol{\theta}'(t_0)) = 0$ (and $\dot{\boldsymbol{\theta}}'(t_0) \in N_{\mathbf{g}}(\boldsymbol{\theta}'(t_0))$ if one needs C^1 continuity). This can be done by using the affine deformation method presented earlier. Next, differentiating (15) yields the constraint $\mathbf{J_g}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = 0, \ \forall t \in [t_0, t_1]$.

Let us now consider the filtered trajectory $\boldsymbol{\theta}''$ defined by

$$\begin{aligned} \forall t \leq t_0 \qquad & \boldsymbol{\theta}''(t) = \boldsymbol{\theta}'(t) \\ \forall t \in [t_0, t_1] \qquad & \dot{\boldsymbol{\theta}}''(t) = \mathrm{proj}_{N_{\mathbf{g}}(\boldsymbol{\theta}''(t))} \dot{\boldsymbol{\theta}}'(t), \end{aligned} \tag{16}$$

where proj is the operator of orthogonal projection. Equation (16) is a differential equation very simple to integrate and whose solution $\boldsymbol{\theta}''$ satisfies the constraint $\mathbf{g}$ for all $t \in [t_0, t_1]$. While in general there is no guarantee that $\boldsymbol{\theta}''$ will stay close to $\boldsymbol{\theta}'$, in practice, for very redundant manipulators and short constraint time intervals, this procedure can yield satisfying results, as illustrated in Figure 4. Finally, if needed, one can make a final affine deformation to reconnect $\boldsymbol{\theta}'(t)_{t \in (t_1, T]}$ with $\boldsymbol{\theta}''(t)_{t \in [t_0, t_1]}$ at time t_1.

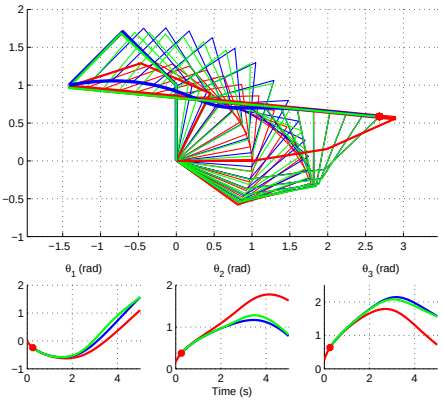

Fig. 4. Kinematics filtering to fulfill fine-grained constraints. The original trajectory (red) allows connecting the initial and final end-effector positions by a straight line in the end-effector space. The joint angle trajectory is next deformed in order to reach the final end-effector position but using a different joint angle configuration (blue trajectory, top plot). However the blue end-effector trajectory (blue thick line in the top plot) is no longer a straight line. The blue joint angle trajectory is then "kinematically filtered" into the green one whose *end-effector trajectory* is a straight line in Cartesian space, similarly to the original red trajectory, but whose *joint angle trajectory* is close to the blue one (see bottom plots). In particular, the final blue and green configurations are almost identical.

The previous development constitutes in some sense a "kinematics filter". A more sophisticate approach – the "dy-

namics filter" [15] – allows filtering a non-dynamically-consistent trajectory into a dynamically-consistent one by using a fast dynamics computation algorithm for structure-varying kinematic chains.

III. Examples of Applications

A. Interactive motion editing

In typical computer graphics applications, the system under study is so complicated that interactive manipulations are the only way to modify the motion to suit one's needs. In this perspective, a pin-and-drag algorithm that allows interactively altering a particular configuration while respecting kinematic and contact constraints using differential inverse kinematics was designed in [16]. Based on the previous development, we propose here a way to extend this algorithm to handle interactive modifications of the *entire trajectory*.

Assume that a joint angle trajectory $\boldsymbol{\theta}(t)_{t\in[0,T]}$ is given, and that we are interested in modifying the position of a certain landmark point $\mathbf{r} = \mathbf{f}(\boldsymbol{\theta})$. For instance, if $\boldsymbol{\theta}$ represents the joint angles of a human leg, $\mathbf{r}$ can represent the position of the right toe, and we want to modify $\mathbf{r}$ so that the toe passes above an obstacle at a time instant t_{obs}. This can be done as follows

1) First, "drag" the position of $\mathbf{r}(t_{\text{obs}})$ towards a higher position $\mathbf{r}_d$ and compute the corresponding $\boldsymbol{\theta}_d$ using the pin-and-drag algorithm [16] ;
2) Next, choose an appropriate $\tau < t_{\text{obs}}$ and deform $\boldsymbol{\theta}(t)_{t\in[0,T]}$ at τ to obtain $\boldsymbol{\theta}'(t)_{t\in[0,T]}$ with $\boldsymbol{\theta}'(t_{\text{obs}}) = \boldsymbol{\theta}_d$;
3) Finally, choose an appropriate $\tau' \geq t_{\text{obs}}$ and deform $\boldsymbol{\theta}'(t)_{t\in[0,T]}$ at τ' to obtain $\boldsymbol{\theta}''(t)_{t\in[0,T]}$ with $\boldsymbol{\theta}''(T) = \boldsymbol{\theta}(T)$.

Note that, at steps 2 and 3, one may also impose velocity, acceleration… constraints to guarantee C^1, C^2… continuity.

The new *trajectory* $\boldsymbol{\theta}''(t)_{t\in[0,T]}$ can be computed very quickly, even for systems with large numbers of degrees of freedom, thanks to the efficiency of the pin-and-drag and affine deformation algorithms. One can thus visualize the whole deformed trajectory *in real-time* as one drags the landmark point, which greatly enhances the editing experience. This is illustrated in Fig. 5.

B. Motion transfer to humanoid robots

We first recorded a hand waving movement (four degrees of freedom : elbow flexion, shoulder pitch, roll, yaw) of a human subject (see the first row of snapshots in Fig. 6). Optical markers were placed on the subject's shoulder, elbow and wrist, which allow reconstructing the joint angles by inverse kinematics. These joint angles were then transferred to a humanoid robot (second row of snapshots in Fig. 6). Another robot movement was obtained by smoothly deforming the joint angle trajectories in order to reach a new final arm configuration corresponding to : elbow angle $-$ 0.2rad, shoulder pitch $+$ 0.2rad, roll $+$ 0.2rad, yaw $-$ 0.2rad (third row of snapshots in Fig. 6).

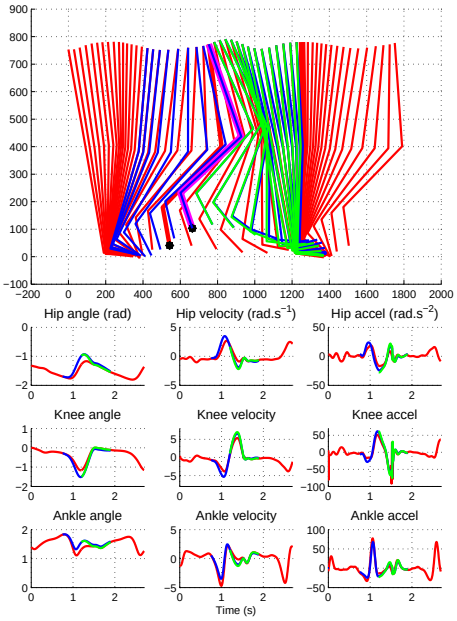

Fig. 5. Interactive editing of a human walking pattern. The original trajectory is reconstructed from motion-capture data and is plotted in red. The intermediate configuration at time t_{obs} is highlighted by bold red sticks (top plot) and the landmark point is represented by a black dot. First, differential inverse kinematics is performed to compute a desired configuration $\boldsymbol{\theta}_d$ (highlighted by bold magenta in the top plot) corresponding to a higher position $\mathbf{r}_d$ of the landmark point (black dot). Then a first deformation (blue) is made to bring the configuration at time t_{obs} towards the desired configuration $\boldsymbol{\theta}_d$. Finally a second deformation (green) is made to bring the configuration at T back to that of the original trajectory, with the original velocity, ensuring thereby C^1 continuity with the final part of the trajectory. Note that the time instant T corresponds here to the time instant of the first heel strike after t_{obs}.

The smoothness of the deformed robot joint angles and of the velocity profiles can be observed in the plots of Fig. 6. Note also that, although noisy (because of motor and sensor noises), the accelerations remained small at the global scale thanks to the C^1 continuity guaranteed by the affine deformation algorithm. Finally, it is possible to implement *online feedback control* by applying this algorithm in a reactive manner.

IV. A Characterization of Trajectory Redundancy by the Group of Admissible Deformations

Building from the previous development, we now discuss the concept of redundancy from the viewpoint of group theory.

A. Configuration and velocity redundancies

A manipulator is said to be kinematically redundant with respect to a task when more degrees of freedom than the minimum number required to execute that task are available, see e.g. [7, 11]. As in section III-A, consider the system

$$\mathbf{r} = \mathbf{f}(\boldsymbol{\theta}), \tag{17}$$

where $\mathbf{r}$ is a vector of dimension m representing the configuration of the end-effector and $\boldsymbol{\theta}$ is a vector of dimension

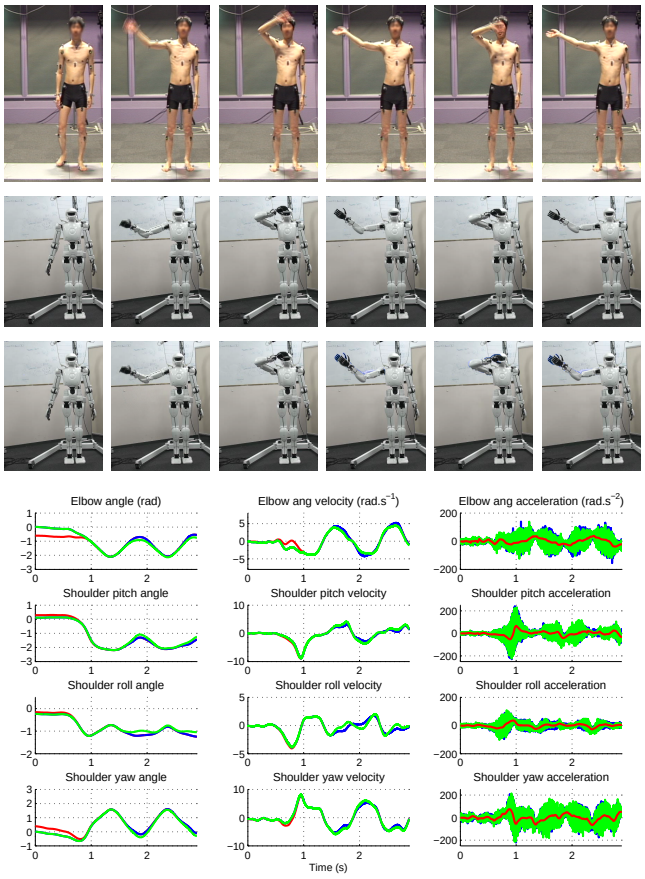

Fig. 6. Transferring a human hand waving motion to a humanoid robot. Top row: the original human movement (snapshots taken every 500ms). Second row (condition NORMAL): the joint angles obtained from inverse kinematics are transferred on the humanoid robot after making one affine deformation at $t = 1$s to align the initial posture of the motion with the standard initial posture of the robot (snapshots taken every 2s). Third row (condition MODIFIED): a second affine deformation is made at $t = 1.5$s to alter the final posture of the robot. In the last three snapshots, the robot's right arm in condition NORMAL was superimposed for comparison. Note that the robot movement speed was reduced to 1/4th of the original human speed to comply with the hardware limitations. The graphics show the joint angles and their derivatives for the original human movement (red), and for the scaled robot movements (NORMAL in blue and MODIFIED in green).

n representing the joint angles. If $n > m$, then there generally exists infinitely many $\boldsymbol{\theta}$ that correspond to a given $\mathbf{r}$, which constitutes the notion of *configuration redundancy*, see Fig. 7A.

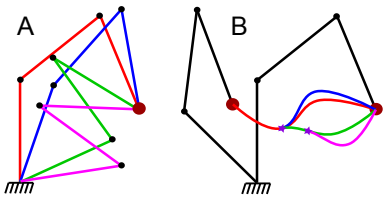

Fig. 7. Configuration redundancy (**A**) and trajectory redundancy (**B**). For simplicity, we have sketched in plot B the end-effector trajectory redundancy, but this notion applies more generally to the joint angle trajectories.

Usually, redundancy is studied from a differential viewpoint, which we call *velocity redundancy*. Differentiating (17) indeed yields $\dot{\mathbf{r}} = \mathbf{J}_{\mathbf{f}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$, where $\mathbf{J}_{\mathbf{f}} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}$ is the Jacobian matrix of $\mathbf{f}$, of dimension $m \times n$. If $n > m$ and $\mathbf{J}_{\mathbf{f}}(\boldsymbol{\theta}(t))$ is non-singular, then a given desired instantaneous velocity $\mathbf{v}_{\mathbf{r}}^d$ of the end-effector can be achieved by infinitely many different instantaneous velocities $\mathbf{v}_{\boldsymbol{\theta}}$ of the joint angles (for notational simplicity, we have dropped the time index t). More precisely, let S represent the null-space of $\mathbf{J}_{\mathbf{f}}(\boldsymbol{\theta})$ and $\mathbf{v}_{\boldsymbol{\theta}}^* = \mathbf{J}_{\mathbf{f}}(\boldsymbol{\theta})^+ \mathbf{v}_{\mathbf{r}}^d$. Then any joint angle velocity in the affine subspace $\{\mathbf{v}_{\boldsymbol{\theta}}^* + S\}$ will achieve the desired end-effector velocity $\mathbf{v}_{\mathbf{r}}^d$ [7, 11].

From a group-theoretic viewpoint, which will be handy later on, let T_S denote the space of the translations whose vectors belong to S. This set can actually be viewed as a Lie subgroup of dimension $n-m$ of the general affine group GA_n. The space of all joint angle velocities $\mathbf{v}_{\boldsymbol{\theta}}$ corresponding to a single end-effector velocity $\mathbf{v}_{\mathbf{r}}^d$ described above can then be seen as the orbit of $\mathbf{v}_{\boldsymbol{\theta}}^*$ under the action of T_S, and the "degree of velocity redundancy" of the system at $\boldsymbol{\theta}$ as the dimension of T_S as a Lie subgroup of GA_n.

B. Trajectory redundancy

The developments of sections II and III have highlighted another type of redundancy, namely *trajectory redundancy*: once a particular joint configuration $\boldsymbol{\theta}_d$ has been chosen from the many possible joint configurations that achieve a given end-effector configuration, there still exists infinitely many joint angle *trajectories* that can bring the manipulator from the initial configuration $\boldsymbol{\theta}_0$ towards $\boldsymbol{\theta}_d$ with a specified velocity, acceleration... while respecting the system kinematic and dynamic constraints, see Fig. 7B.

Unlike configuration/velocity redundancies, trajectory redundancy is generally of infinite dimension. Finding a convenient way of *parameterizing* a subset of admissible trajectories is then of particular interest. We have seen, from the development of section II that, given a time instant τ, the set of admissible trajectories that can be obtained by affinely deforming an original trajectory $\boldsymbol{\theta}(t)_{t\in[0,T]}$ is – roughly – the orbit of that trajectory under the action of a subgroup of dimension $n^2-(k+p)n$ of GA_n. To obtain a more convenient description, we now propose a construction that removes the dependance upon τ, allowing thereby the composition of deformations at different time instants.

For a given τ, let $G(\tau)$ denote the group of admissible affine transformations at time τ. As we have seen, $G(\tau)$ is of dimension $n^2-(k+p)n$ if $\boldsymbol{\theta}$ is not singular at τ.

Now consider two finite sequences $\hat{\tau} = \{\tau_1, \ldots, \tau_l\}$ (with $0 \leq \tau_1 < \ldots < \tau_l < T$) and $\hat{g} = \{g_1, \ldots, g_l\} \in G(\tau_1) \times \ldots \times G(\tau_l) = \widehat{G}(\hat{\tau})$. We define the deformation $f_{\hat{\tau},\hat{g}}$ of $\boldsymbol{\theta}$ by

$$\forall t \in [0, \tau_1],\ f_{\hat{\tau},\hat{g}}(\boldsymbol{\theta})(t) = \boldsymbol{\theta}(t)$$

$$\forall i \in [1, l],\ \forall t \in (\tau_i, \tau_{i+1}],\ f_{\hat{\tau},\hat{g}}(\boldsymbol{\theta})(t) = (g_1 \circ \ldots \circ g_i)(\boldsymbol{\theta}(t)),$$

with the convention $\tau_{l+1} = T$.

Given two sequences $\hat{\tau}$ and $\hat{g} \in \widehat{G}(\hat{\tau})$, we define the inverse of $f_{\hat{\tau},\hat{g}}$, denoted $f_{\hat{\tau},\hat{g}}^{-1}$, by the deformation associated with the sequences $\hat{\tau}$ and $\hat{g}^{-1} = \{g_1^{-1}, \ldots, g_l^{-1}\}$.

Given $\hat{\tau}$, $\hat{\tau}'$, $\hat{g} \in \widehat{G}(\hat{\tau})$ and $\hat{g}' \in \widehat{G}(\hat{\tau}')$, we define the product $f_{\hat{\tau},\hat{g}} \star f_{\hat{\tau}',\hat{g}'}$ by the deformation associated with the sequences $\hat{\tau} \cup \hat{\tau}'$ and $\hat{g} \sqcup \hat{g}'$ where the latter sequence is constructed as follow

- For all $\tau_i \in \hat{\tau} \setminus \hat{\tau}'$, put g_i into $\hat{g} \sqcup \hat{g}'$ at the position corresponding to τ_i ;
- For all $\tau'_i \in \hat{\tau}' \setminus \hat{\tau}$, put g'_i into $\hat{g} \sqcup \hat{g}'$ at the position corresponding to τ'_i ;
- For all $\sigma \in \hat{\tau}' \cap \hat{\tau}$, say $\sigma = \tau_i = \tau'_j$, put $g_i \circ g'_j$ into $\hat{g} \sqcup \hat{g}'$ at the position corresponding to σ.

We can now state the following proposition

Proposition 1 The set of all $f_{\hat{\tau},\hat{g}}$ endowed with the inverse and product operations as defined above is a Lie group of dimension $n^2 - (k+p)n + 1$. We call this group the *affine deformation group* of $\boldsymbol{\theta}(t)_{t\in[0,T]}$ and denote it by $A(\boldsymbol{\theta})$. $\triangle$

Recall that we have previously identified the redundancy of velocities with a certain group of translations T_S of dimension $n - m$. Similarly, we identify here part of the redundancy of trajectories with the group $A(\boldsymbol{\theta})$, in the sense that the orbit of $\boldsymbol{\theta}$ under the action of $A(\boldsymbol{\theta})$ is the set of all the admissible trajectories that can be obtained from $\boldsymbol{\theta}$ by finite sequences of affine deformations. Note that, in the limit of large n, which is the case of highly redundant manipulators or of humanoids, the dimension of $A(\boldsymbol{\theta})$ grows linearly with n^2 allowing thereby more "freedom" than configuration redundancy alone (the dimension of T_S grows linearly with n only).

The group property and the matrix representation of the admissible deformations allow searching efficiently (using random sampling, optimization, etc.) within the space of trajectory redundancy, as partially illustrated in sections II and III. We also believe that a better understanding of $A(\boldsymbol{\theta})$ – in particular that of its associated Lie algebra – can provide powerful methods to make trajectory deformations.

V. Conclusion

We have presented a new method of trajectory deformation for redundant manipulators based on affine transformations. This method is exact, fast, and guarantees the smoothness of the resulting trajectories. Furthermore, the resulting trajectories can be chosen so as to remain close the original trajectory (with explicit bounds), to optimize a given cost function, or to satisfy inequality constraints at specific time instants. Combined with further kinematics or dynamics filtering, the method can also yield trajectories that satisfy constraints that apply on continuous time intervals, in a coarse-to-fine manner.

This method is advantageous with respect to spline- or dynamic-system-based approaches in that it does not require choosing an exogenous functions basis (such as hierarchical spline basis [6], wavelet spline basis [12], Gaussian kernels [3, 13]...) : indeed, the only "basis functions" we use are the original joint angle trajectories. In particular, there is no need to fine-tune the basis functions or to put extra constraints on the coefficients multiplying the basis functions in order to avoid undesirable behaviors (such as spline trajectories that undulate too much [6] or wavelets that have too much energy [12]). Finally, the purely *endogenous* nature of the proposed method allows the deformed motions to preserve qualitative properties, such as e.g. the difficult-to-quantify "naturalness" [17], of the original motions with no extra effort.

Our current research focuses on developing this method for full-scale applications in character animation [9, 6, 16, 17] and humanoid robots control [15, 13].

Acknowledgments

We would like to thank Mr. Hamano for the human experiment data, Dr. Srinivasa and Mr. Santacruz for help with the humanoid robot experiment, Prof. Bennequin and Dr. Diankov for helpful discussions. This work was supported by "Grants-in-Aid for Scientific Research" for JSPS fellows and by a JSPS postdoctoral fellowship.

References

[1] D. Bennequin, R. Fuchs, A. Berthoz, and T. Flash. Movement timing and invariance arise from several geometries. *PLoS Comput Biol*, 5(7): e1000426, Jul 2009. doi: 10.1371/journal.pcbi.1000426.

[2] M. Gleicher. Retargetting motion to new characters. In *ACM SIGGRAPH*, pages 33–42. ACM, 1998.

[3] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1398–1403, 2002.

[4] O. Kanoun, F. Lamiraux, and P.-B. Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27(4):785–792, 2011. doi: 10.1109/TRO.2011.2142450.

[5] F. Lamiraux, D. Bonnafous, and O. Lefebvre. Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 20 (6):967–977, 2004. doi: .829459.

[6] J. Lee and S.Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *ACM SIGGRAPH*, pages 39–48. ACM, 1999.

[7] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, 1990.

[8] Q.-C. Pham. Fast trajectory correction for nonholonomic mobile robots using affine transformations. In *Robotics: Science and Systems*, 2011.

[9] C. Rose, B. Guenter, B. Bodenheimer, and M.F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *ACM SIGGRAPH*, pages 147–154. ACM, 1996.

[10] K. Seiler, S. Singh, and H. Durrant-Whyte. Using Lie group symmetries for fast corrective motion planning. In *Algorithmic Foundations of Robotics IX*, 2010.

[11] B. Siciliano and J.-J. E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *International Conference on Advanced Robotics*, pages 1211–1216, 1991. doi: 10.1109/ICAR.1991.240390.

[12] A. Ude, C.G. Atkeson, and M. Riley. Planning of joint trajectories for humanoid robots using B-spline wavelets. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2223–2228. IEEE, 2000.

[13] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815, 2010.

[14] Y Uno, M Kawato, and R Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. minimum torque-change model. *Biol Cybern*, 61(2):89–101, 1989. ISSN 0340-1200.

[15] K. Yamane and Y. Nakamura. Dynamics filter – concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, 2003. doi: 10.1109/TRA.2003.810579.

[16] K. Yamane and Y. Nakamura. Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on visualization and computer graphics*, pages 352–360, 2003.

[17] K. Yamane, J.J. Kuffner, and J.K. Hodgins. Synthesizing animations of human manipulation tasks. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 532–539. ACM, 2004.