53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Optimization on the space of rigid and flexible motions: an alternative manifold optimization approach

Pirooz Vakili[¶], Hanieh Mirzaei[†], Shahrooz Zarbafian[¶], Ioannis Ch. Paschalidis[§], Dima Kozakov[‡], Sandor Vajda[‡]

***Abstract*— In this paper we consider the problem of minimization of a cost function that depends on the location and poses of one or more rigid bodies, or bodies that consist of rigid parts hinged together. We present a unified setting for formulating this problem as an optimization on an appropriately defined manifold for which efficient manifold optimizations can be developed. This setting is based on a Lie group representation of the rigid movements of a body that is different from what is commonly used for this purpose. We illustrate this approach by using the steepest descent algorithm on the manifold of the search space and specify conditions for its convergence.**

I. INTRODUCTION

In this paper we consider the problem of minimization of a cost function that depends on the location and poses of one or more rigid bodies, or bodies that consist of rigid parts hinged together. This is a formulation used in a number of engineering applications such as workpiece or camera localization or calibration (see, e.g., [1], [2], [3], [4]) or computer vision (see, e.g., [5], [6]). Our interest in this problem comes from the local optimization problems encountered in the area of computational docking of biological macromolecules (see, e.g., [7], [8], [9], [10]).

The main contribution of this paper is to describe a unified setting for formulating this problem as that of an optimization on an appropriately defined manifold: we show that this problem can be formulated as an optimization on a Lie group (i.e., a group that simultaneously has a differentiable manifold structure consistent with its group structure) that is a *direct product* of its components Lie groups; furthermore, the components are endowed with appropriate structures that allow for efficient computation of gradients, exponential parametrization; therefore, gradient based optimization algorithms on the product manifold can be efficiently performed. We illustrate this approach by describing and analyzing the steepest descent algorithm on the product manifold/Lie group. As will be explained below, a critical element of this construction is an alternative Lie group representation of the rigid movements of a body that is different from the commonly used representation.

† Division of Systems Eng., Boston University, hanieh@bu.edu
‡ D. Kozakov, and S. Vajda are with the Dept. of Biomedical Eng., Boston University, {midas, vajda}@bu.edu
§ Dept. of Electrical & Computer Eng., and Division of Systems Eng., Boston University, yannisp@bu.edu
¶ Dept. of Mechanical Eng., Boston University, szarbaf@bu.edu
¶ Corresponding author. Dept. of Mechanical Eng. and Division of Systems Eng., Boston University, vakili@bu.edu

The above optimization problem can be formulated as a constrained Euclidean optimization problem. The advantage of such a formulation is that the search space is a Euclidean space with a well-known geometry for which various efficient and well-understood optimization algorithms are available. On the other hand, the dimension of the resulting search space can be very large, leading to slow convergence of optimization algorithms. By formulating the problem as a manifold optimization we arrive at a search space with the smallest possible dimension. The question then becomes whether we can efficiently optimize on such a manifold. Many standard optimization algorithms on Euclidean spaces generalize to (Riemannian) manifolds (see, e.g., [11], [12]). However, the geometry of the resulting manifold may present challenges for optimization (see, e.g., [2], [12]) and the efficiency of such generalizations depends on the ease with which certain quantities such as gradients of functions or geodesics of the manifold can be computed. The authors of [13], driven by goal of obtaining efficient manifold optimization algorithms, generalize the class of valid (convergent) optimization algorithms significantly and reduce the computational burden of such algorithms through certain approximations. On the other hand, for some manifolds such as the manifold of orientation-preserving rotations in $\mathbb{R}^3$, i.e., the Special Orthogonal group $SO(3)$, gradients and geodesics can be easily computed and direct manifold optimization can be used efficiently (see, e.g., [14] for a detailed treatment of this case).

The space of rigid moves of a body, i.e., combination of rotations and translations is commonly represented by the Special Euclidean group (or simply the Euclidean group) $SE(3)$ (see, e.g. [15]). With this representation, the rigid body optimization is formulated as an optimization on $SE(3)$. While optimization on $SO(3)$ can be viewed as "easy," optimization on $SE(3)$ presents special challenges mainly due to the fact that $SE(3)$ is a *semi-product* of the Lie groups of rotation, $SO(3)$, and translations, $\mathbb{R}^3$. The semi-product structure introduces a "mismatch" between the natural Riemannian metric on $SO(3)\times\mathbb{R}^3$, as a direct product of $SO(3)$ and $\mathbb{R}^3$ manifolds, and the group structure of $SE(3)$. As a result, direct generalization of standard Euclidean optimization algorithms to $SE(3)$ becomes non-trivial (for an informative discussion and some proposed remedies, see [2]).

By contrast, we use an alternative group of rigid body transformations that corresponds to the Lie group $SO(3)\times$

$\mathbb{R}^3$, i.e., the direct product of the Lie groups $SO(3)$ and $\mathbb{R}^3$. Given this formulation, the rigid body optimization problem can be naturally defined as an optimization on $SO(3) \times \mathbb{R}^3$. As a result, the difficulties associated with optimization on $SE(3)$ are fully avoided. Furthermore, this representation allows us to append possible flexibilities of the body as well as possibly introduce multiple bodies to the search space and treat the resulting optimization problem as an optimization on a direct-product Lie group where computation on its components can be performed efficiently. We illustrate this approach by using the steepest descent algorithm on the manifold of the search space.

The rest of the paper is organized as follows. In Section II we provide some preliminary definitions and results and introduce our general setting. We introduce our alternative group of rigid body transformations in Section III. In Section IV we give a description of the steepest descent algorithm on a Lie group that is the direct product of its components and analyze its convergence. We conclude in Section V.

II. PRELIMINARIES

In this section we briefly review some preliminary definitions and results that will be needed in what follows. We presume some familiarity with differential geometric terminology and results on the part of the reader and state the results in a fairly general context. The main message of this section is that direct product Lie groups inherit many of the relevant structures needed for gradient based optimization from their component Lie groups and that optimization algorithms on the component Lie groups imply an optimization algorithm on the direct product Lie group in a straightforward manner.

A. Direct and semidirect product of Lie groups

Let G_1 and G_2 be two Lie groups with group operations denoted by $*$ and $*'$ respectively. Then, the direct product of G_1 and G_2, denoted by $G_1 \times G_2$, is group operation defined by:

$$(g_1, g_2) \diamond (g_1', g_2') = (g_1 * g_1',\ g_2 *' g_2'),$$

on the product space $\{(g_1, g_2); g_1 \in G_1, g_2 \in G_2\}$. It can be easily verified that with the operation $\diamond$, $G_1 \times G_2$ is a group. It is also a product manifold, therefore, it is a Lie group. This Lie group is called the *direct product* of the component Lie groups.

To define a semi-product of G_1 and G_2, assume that a smooth action

$$h : G_1 \times G_2 \to G_2,$$

is given. Define the operation $\diamond'$ on $G_1 \times G_2$ by

$$(g_1, g_2) \diamond' (g_1', g_2') = (g_1 * g_1',\ g_1' *' h(g_1, g_2')).$$

Again, it can be verified that with the operation $\diamond'$, the product manifold $G_1 \times G_2$ is a group and therefore a Lie group. This Lie group is called the *semi-direct product* of the component Lie groups G_1 and G_2 and denoted by

$$G_1 \rtimes_h G_2.$$

Note that by contrast to the direct product of G_1 and G_2 where the group action is performed componentwise, in the case of the semi-direct product, a coupling between the components of G_1 and G_2 is created through the function h. This coupling can be a source of complications as is the case with $SE(3) = SO(3) \rtimes \mathbb{R}^3$.

B. Exponential map

Let G be a Lie group. The tangent space at the identity of the group, identified with the space of left-invariant vector fields on G, is an algebra under the Lie bracket operation; it is called the Lie algebra of G and is denoted by $\mathfrak{g}$ (see, e.g., [15]). The *exponential map* of a Lie group at the group identity, e, maps points in $\mathfrak{g}$ to points on the manifold G and in some sense it provides the most appropriate local parametrization of the manifold at e for the purpose of gradient based optimization:

$$exp : \mathfrak{g} = T_eG \to G,$$

where T_eG is the tangent space of G at e. For any $v \in T_eG$, let V denote the left-invariant vector field associated with v and $\Phi_v(t)$ the integral curve of V, i.e., a curve satisfying $d\Phi_v(t)/dt = V(t)$ along with $\Phi_v(0) = e$. Then, the exponential at v is defined as:

$$exp(v) = \Phi_v(1).$$

It can be shown that:

$$exp((t+s)v) = exp(tv) * exp(sv).$$

In other words, $\Phi_v : R \to G$ generates a one-parameter subgroup of G. If G has a bi-invariant Riemannian metric, then these one-parameter subgroups of G are geodesics of G going through e.

This whole machinery can be transported to any other point of the group, say g, by simply defining

$$\Phi_v(t, g) = g * exp(tv).$$

In this case lines going through the origin of the tangent space at any point g, T_gG, are mapped via the exponential map to geodesics of G going through g.

C. Steepest descent algorithm on a Lie group

Assume the above assumptions about the Lie group G are in force and we are interested in minimizing a smooth cost function $f : G \to \mathbb{R}$.

Consider a point g^0 on G. Assume a local parametrization of G at g^0 via the exponential map. The gradient of f with respect to this parametrization specifies a vector v in the tangent space to G at g^0, i.e., $T_{g^0}G$. The equivalent of a line search on the manifold is a search along the geodesic of G in the direction $-v$ starting from g^0, i.e., the one parameter subgroup given by $g^0 * exp(-tv)$. Once a new point $g^1 \in G$ along the geodesic is selected, the process is repeated until some stopping condition is satisfied. Therefore, the steepest descent algorithm is given by:

Steepest descent algorithm
At the mth iteration of the algorithm

- Calculate v^m, the gradient of f at the current point g^m.
- Perform a line search on T_{g^m} along the $-v^m$ direction and find an appropriate step size α^m.
- Set
$$g^{m+1} = g^m * exp(-\alpha^m v^m).$$
Repeat the above steps until the algorithm satisfies a specified convergence criteria.

The computational cost or feasibility of the above algorithm depends on the ease with which the gradient can be computed and the exponential map evaluated. For a more general class of "line search" algorithms on manifolds, see [13], Chapter 4.

D. Direct product of Lie groups

Let $G_1, \cdots, G_k$ be k Lie groups and let $G = G_1 \times \cdots \times G_k$ be the direct product of the component groups. In this case the product group "inherits" many of the relevant structures from its component manifolds (for the simplest case, consider the n dimensional Euclidean space $\mathbb{R}^n$ and its one dimensional components $\mathbb{R}$):

- G is a Lie group;
- Let $g_i \in G_i$ and let $T_{g_i}G_i$ be the tangent space to G_i at g_i, $i = 1, \cdots, k$ and $g = (g_1, \cdots, g_k) \in G$. Then,
$$T_g G = T_{g_1}G_1 \times \cdots \times T_{g_k}G_k,$$
i.e., the tangent space to G at g is simply the direct product of the tangent spaces to the component groups;
- Consider the component exponential maps
$$\Phi_{v_i}(t, g_i) = g_i * exp_i(tv_i), \;\; g_i \in G_i, v_i \in T_{g_i}G_i, \; i = 1, \cdots, k,$$
then, the exponential map of the product manifold, G, is simply evaluated componentwise, i.e.,
$$\Phi_v(t, g) = (g_1 * exp_1(tv_1), \cdots, g_k * exp_k(tv_k)),$$
where $g = (g_1, \cdots, g_k) \in G$ and $v = (v_1, \cdots, v_k) \in T_g G$.
- The gradient of a function f on G can be computed "componentwise," and the steepest descent algorithm can also be implemented using information on components.

For additional properties, see , e.g., [6], Appendix A.

III. AN ALTERNATIVE GROUP OF RIGID BODY TRANSFORMATIONS

In this section we begin by clarifying the distinction between the common representation of the rigid movement of a body via the Special Euclidean group $SE(3) = SO(3) \rtimes \mathbb{R}^3$ (see, e.g., [16], [17], [15]) and our proposed representation via $SO(3) \times \mathbb{R}^3$. We then give an example of how flexible movements of parts of the body can be appended to the rigid movement of the whole body by using a direct product construction. We begin with a review of the $SE(3)$ group.

A. Euclidean group $SE(3)$

Let $SO(3) = \{R \in \mathbb{R}^{3\times 3}; R^T R = I; det(R) = 1\}$ denote the group of orientation-preserving rotations of $\mathbb{R}^3$ and consider the set $SO(3) \times \mathbb{R}^3 = \{(R,t); R \in SO(3), t \in \mathbb{R}^3\}$. Let $g = (R,t)$ and $g' = (R',t')$ be two elements of $SO(3) \times \mathbb{R}^3$. Then, the group operation of $SE(3)$ is defined by:
$$g'g = (R'R, \; R't + t').$$
Let $h : SO(3) \times \mathbb{R}^3 \to \mathbb{R}^3$ be defined by $h(R,t) = Rt$. Then, we have: $SE(3) = SO(3) \rtimes_h \mathbb{R}^3$. In other words, $SE(3)$ is the semi-product of $SO(3)$ and $\mathbb{R}^3$ defined by using the function h.

Each member of $SE(3)$ defines an action on $\mathbb{R}^3$ as follows (see, e.g., [15], Section 2.4). For $g \in SE(3)$, let $g : \mathbb{R}^3 \to \mathbb{R}^3$ be defined by:
$$g(q) = Rq + t.$$
This mapping defines an action of the $SE(3)$ group on $\mathbb{R}^3$ since we have:
$$g' \circ g(q) = g'(g(q)) = R'(Rq+t)+t' = R'Rq + R't + t' = g'g(q),$$
where $\circ$ denotes composition of functions. Therefore,
$$g' \circ g = g'g.$$
It can be easily verified that this action corresponds to a rigid body transformation of $\mathbb{R}^3$ (in the stricter sense defined in [16], Chapter 2).

It is well-known that there is a "mismatch" between the natural Riemannian metric on $SO(3) \times \mathbb{R}^3$, as a direct product of $SO(3)$ and $\mathbb{R}^3$ manifolds, and the group structure of $SE(3)$: $SE(3)$ does not have a natural bi-invariant Riemannian metric. One implication of this mismatch is that the geodesics of $SE(3)$ are no longer one-parameter subgroups of $SE(3)$ and optimization algorithms on the component manifolds $SO(3)$ and $\mathbb{R}^3$ do not lead to optimization algorithms on $SE(3)$ (for an informative discussion, see [2]).

To address these difficulties, we describe an alternative representation of rigid motions that we first introduced in [8] and [10].

B. Direct product group $SO(3) \times \mathbb{R}^3$

The direct-product operation on the product group $SO(3) \times \mathbb{R}^3$ is naturally defined by:
$$g' * g = (R'R, t' + t).$$
We use $*$ to denote the direct-product group operation and to distinguish it from the group operation of $SE(3)$.

The novel element of the new representation is the action we associate with this group. We define this action on $\mathbb{R}^3 \times \mathbb{R}^3$ as follows. For $g \in SO(3) \times \mathbb{R}^3$, let $g : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3 \times \mathbb{R}^3$ be defined by:
$$g(q, p) = (R(q-p) + p + t, p + t),$$
$(q, p \in \mathbb{R}^3)$.

In words, the action of g on the first component $q \in \mathbb{R}^3$ is to rotate q according to the rotation matrix R but with the "center of rotation" (i.e., the origin of the coordinate

system) moved to p, and translate it by t. The action of g on the second component simply translates the point p by t. Equivalently, we can think that the action on the second component is of the same type as the action on the first component since $R(p-p)+p+t=p+t$. The following is an immediate result (see [8] for a proof).

Proposition 1: The above transformation defines an action of the group $SO(3)\times\mathbb{R}^3$ on $\mathbb{R}^3\times\mathbb{R}^3$.

We next show that, for any $p\in\mathbb{R}^3$, the action of $SO(3)\times\mathbb{R}^3$ on $\mathbb{R}^3\times\mathbb{R}^3$ is a rigid body transformation of the first component $\mathbb{R}^3$.

Let $\pi_i:\mathbb{R}^3\times\mathbb{R}^3\to\mathbb{R}^3$ $(i=1,2)$ be projections on the first and second coordinate $(\pi_1(q,p)=q,\ \pi_2(q,p)=p)$. For any fixed $p\in\mathbb{R}^3$, let

$$g_p:\mathbb{R}^3\to\mathbb{R}^3\times\mathbb{R}^3,$$

be defined by $g_p(q)=g(q,p)$. Then, we have the following proposition (proof provided in [8]):

Proposition 2: For any p,

$$\pi_1\circ g_p:\mathbb{R}^3\to\mathbb{R}^3,$$

is a rigid body transformation of $\mathbb{R}^3$.

A critical feature of the new representation of rigid body transformations is that, by contrast to the $SE(3)$ formulation, translational moves and rotational moves are decoupled. For example, let $g=(I,t)$, i.e., translation only by t and $g'=(R,0)$, i.e., rotation only by R. Then it can be easily seen that in $SE(3)$,

$$gg'\neq g'g,$$

where as in $SO(3)\times\mathbb{R}^3$

$$g*g'=g'*g.$$

The above can be verified by considering the homogeneous representation of g an g' in $SE(3)$ and $SO(3)\times\mathbb{R}^3$, respectively.

Since the group $SO(3)\times\mathbb{R}^3$ is a direct product of $SO(3)$ and $\mathbb{R}^3$ both as groups and as Riemannian manifolds, there is no mismatch between the group and the natural Riemannian structures and we do not face the complications that are associated with $SE(3)$ rigid body transformations.

Using the $SO(3)\times\mathbb{R}^3$ group to represent rigid moves of a body in $\mathbb{R}$, we are within the framework of the direct product of Lie groups presented in Section II. Therefore, steepest descent optimization algorithms defined on the component groups $SO(3)$ and $\mathbb{R}^3$ directly lead to optimization algorithms on $SO(3)\times\mathbb{R}^3$. Gradient based optimization algorithms on $\mathbb{R}^3$ are well studied. For informative discussions of gradient-based optimization algorithms on $SO(3)$, see, e.g., [14] and [18].

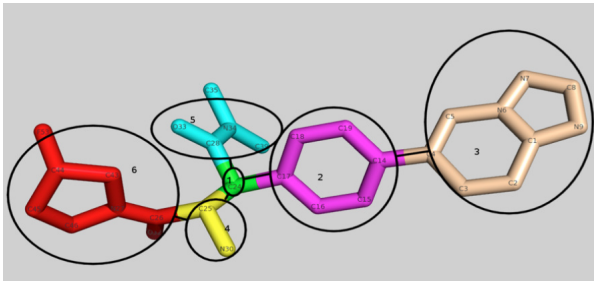

Fig. 1. A flexible body consisting of 6 rigid parts connected together by 5 rotatable bonds.

C. Including flexibility: Lie Group $SO(3)\times\mathbb{R}^3\times T^d$

To show how movements of a body that has some flexibility can be considered in the setting of direct-product Lie groups presented in Section II, we give an example motivated by problems in docking of biological macromolecules. Consider a body consisting of d rigid parts connected together by d rotatable bonds/hinges (see, e.g., Figure 1.)

Let $\theta_i\in\mathbb{R}$ be the rotational parameter associated with the ith bond. Then $\theta=(\theta_1,\cdots,\theta_d)\in\mathbb{R}^d$ represents a combination of bond rotations while the body as whole is stationary. $\mathbb{R}^d$ with the addition operation is a trivial Lie group. Given this group structure, $C=\{2\pi a;\ a\in\mathbb{Z}^d\}$ is a normal subgroup of $\mathbb{R}^d$. Define an equivalence relation on $\mathbb{R}^d$ by letting

$$\theta\sim\theta'\Leftrightarrow\theta-\theta'\in C.$$

The quotient space of $\mathbb{R}^d$ with respect to the equivalence relation $\sim$ (or subgroup C) is a homogeneous space and a Lie group isomorphic to a d-dimensional *torus*:

$$T^d=\underbrace{S^1\times S^1..\times S^1}_{\text{d times}}.$$

See, e.g., [15], section 3.2.

The direct-product group operation on $SO(3)\times\mathbb{R}^3\times T^d$ is naturally defined by:

$$(R',t',[\theta'])*(R,t,[\theta])=(R'R,t'+t,[\theta']+[\theta]),$$

where $[\theta]$ represents the equivalence class of θ.

Given the above associations, it is not difficult to see that the space of the combined movements of the flexible body, including rigid movement and bond rotations, is given by the direct product of $SO(3)\times\mathbb{R}^3$ and T^d both as Riemannian manifolds and as groups. In other words, this space is a Lie group.

Note that the above representation of the space of rigid and flexible motions of the body can be extended to a number of similar bodies simply by taking the direct product of the representations.

IV. STEEPEST DESCENT ON DIRECT-PRODUCT LIE GROUPS

Following our discussion in the previous sections, we consider the following setting:

- $G=G_1\times\cdots\times G_k$ is a Lie group that is the direct product of its component Lie groups;

- For any function f, the component "partial derivatives" of f with respect to the exponential parametrization of G_k at g_k can be easily computed;
- At any $g = (g_1, \cdots, g_k) \in G$ the exponential map from the tangent space at g, T_g, to G can be computed based on the component exponential maps;

In this case, the steepest descent algorithm is given as follows:

A. The steepest descent algorithm

At the mth iteration of the algorithm

- Calculate $v^m = (v_1^m, \cdots, v_k^m)$, the gradient of f at the current point g^m;
- Perform a line search on T_{g^m} along the $-v^m$ direction and find an appropriate step size α^m;
- Set
$$g^{m+1} = g^m * exp(-\alpha^m v^m);$$
Repeat the above steps until the algorithm satisfies a specified convergence criteria.

B. Convergence of the steepest descent algorithm

The convergence of the steepest descent algorithm in the setting we described above is a consequence of a more general result given in [13], chapter 4. The authors consider optimization on a Riemannian manifold G and relax the requirement of a local exponential map parametrization of G. Instead, they define the notion of a *retraction* mapping from the tangent space of the manifold to the manifold that satisfies a "local rigidity condition;" we can loosely interpret local rigidity as follows: once a direction v on the tangent space of the manifold is selected, moving along that direction on the tangent space is mapped by the retraction onto a curve on the manifold that locally (i.e., at the point) moves in the same direction as v. The authors further relax their gradient following/line search algorithm by requiring simply that the direction followed by the algorithm be "gradient-related," namely, if g^m is the "location" of the algorithm at the mth step, v^m, the gradient of f at g^m, and u^m, the line search direction at the mth step, then

$$\limsup_{m\to\infty} \; v^m \cdot u^m < 0,$$

where $v^m \cdot u^m$ is the inner product of v^m and u^m.

In our setting we have assumed that the exponential map parametrization and the gradient of the function we are minimizing are not overly costly to compute. In fact the group structure of G is a source of simplification in this case. The requirement of local rigidity is clearly satisfied as a line in the direction of a vector v on the tangent space is mapped onto the integral curve of the left invariant vector field associated with v. The second condition of gradient-relatedness also holds trivially as we follow the opposite direction of the gradient. Therefore, Theorem 4.3.1 and Corollary 4.3.2 of [13] imply the following proposition.

Proposition 3: Let G be a Lie group satisfying the assumptions specified above and assume f is a smooth function on G such that the level set $L = \{f(g) \leq f(g^0)\}$ of f is compact. Let v^m be the gradient of f at mth step of the steepest descent algorithm described above, then

$$lim_{m\to\infty} \|v^m\| = 0.$$

In other words, the steepest descent algorithm converges to a critical point of the function f on G.

C. An illustrating example

In order to illustrate the behavior of the steepest descent algorithm in our setting we consider a problem of *weighted least square fitting* defined as follows: We are given two sets of points $q_i \in \mathbb{R}^3$ and $q_i^* \in \mathbb{R}^3$, a set of weights $a_i \in R$ (see, e.g., [19], [5]). Our objective is to minimize

$$f = \sum_{i=1}^{n} a_i \|q_i - q_i^*\|^2,$$

by moving the points while keeping the Euclidean distances between q_i's and those between q_i^*'s unchanged. In other words, we consider the set $\{q_i; i = 1, \cdots, n\}$ and $\{q_i^*; i = 1, \cdots, n\}$ as rigid objects.

We can define this problem as an optimization on $SO(3) \times R^3$. Holding q_i^*'s fixed, we can move q_i's as a rigid body using rotation R and translation t. Selecting a "center of rotation" p, we can write the cost function as

$$f(R,t) = \sum_{i=1}^{n} a_i \|R(q_i - p) + p + t - q_i^*\|^2.$$

Figure 2 shows snapshots of the evolution of a steepest descent algorithm where the set $\{q_i; i = 1, \cdots, 4\}$ corresponds to the corner points of the quad on the upper right corner of the first panel and $\{q_i^*; i = 1, \cdots, 4\}$ are the corner points of the quad on the lower left corner of the first panel. The problem can be viewed as placing or "landing" the "quad object" on a prescribed location. The weights are all equal to 1. The blue figure traces the steps of the steepest descent algorithm starting from the top panel and ending at the bottom panel.

V. CONCLUSION

We have presented a unified setting for the problem of minimizing a cost function with respect to rigid and flexible movements of a body in $\mathbb{R}^3$. In this paper, we limited ourselves to the steepest descent algorithm. Generalizations to other optimization algorithms on the Lie groups we have considered in this paper are possible and somewhat straightforward given that such algorithms for general Riemannian manifolds have been developed and analyzed. We have begun exploring the application of the steepest descent algorithm presented in this paper to protein docking problems. Taking some computational cost into account, we had previously opted to use a Euclidean optimization algorithm on the tangent space of our representation of rigid and flexible transformations. Preliminary results of the application of the new algorithm suggests that we can obtain solutions of higher quality than the original approach.

VI. ACKNOWLEDGMENTS

Research supported in part by NIH/NIGMS under grants R01-GM093147, by the NSF under grants CNS-1239021 and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-10390, and by the ONR under grant N00014-10-1-0952.

REFERENCES

[1] Z. Li, J. Guo, and Y. Chu, "Geometric algorithms for workpiece localization," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 864–878, 1998.

[2] S. Gwak, J. Kim, and F. C. Park, "Numerical optimization on the Euclidean group with applications to camera calibration," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 1, pp. 65–74, Feb. 2003.

[3] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *Proceedings of the 48th IEEE Conference on Decision and Control*, Dec. 2009, pp. 901–908.

[4] R. Tron, A. Terzis, and R. Vidal, "Distributed consenus algorithms for image-based localization in camera sensor networks," in *Distributed Video Sensor Networks*, B. B. et al., Ed. Springer-Verlag, 2011, ch. 20, pp. 289–302.

[5] K. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, Sept 1987.

[6] Y. Ma, J. Koseck, and S. Sastry, "Optimization criteria and geometric algorithms for motion and structure estimation," *International Journal of Computer Vision*, vol. 44, pp. 219–249, 2001.

[7] I. C. Paschalidis, Y. Shen, S. Vajda, and P. Vakili, "Sdu: A semi-definite programming-based underestimation method for stochastic global optimization in protein docking," *IEEE Transaction on Automatic Control*, vol. 52, no. 4, pp. 664–676, 2007.

[8] H. Mirzaei, D. Kozakov, D. Beglov, I. C. Paschalidis, S. Vajda, and P. Vakili, "A new approach to rigid body minimization with application to molecular docking," in *51th IEEE Conference on Decision and Control*, 2012, pp. 2983–2988.

[9] H. Mirzaei, E. Villar, S. Mottarella, D. Beglov, I. C. Paschalidis, S. Vajda, D. Kozakov, and P. Vakili, "Flexible refinement of protein-ligand docking on manifolds," in *52th IEEE Conference on Decision and Control*, Dec. 2013, pp. 1392–1397.

[10] H. Mirzaei, D. Beglov, I. C. Paschalidis, S. Vajda, P. Vakili, and D. Kozakov, "Rigid body energy minimization on manifolds for molecular docking," *Journal of Chemical Theory and Computation*, vol. 8, no. 11, pp. 4374–4380, 2012.

[11] R. W. Brockett, "Differential geometry and the design of gradient algorithms," in *Proceedings of the Symposium on Pure Mathematics*, vol. 54, 1994, pp. 69–92.

[12] S. T. Smith, "Optimization techniques on Riemannian manifolds," in *Proc. Fields Inst. Workshop on Hamiltonian and Gradient Flows, Algorithms, and Control*, 1994.

[13] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.

[14] F. C. Park, J. Kim, and C. Kee, "Geometric descent algorithms for attitude determination using the global positioning system," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 1, pp. 26–33, 2000.

[15] J. M. Selig, *Geometric Fundamentals of Robotics*. Springer, 2005.

[16] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[17] G. S. Chirikjian, "Group theory and biomolecular conformation: I. Mathematical and computational models," *Journal of Physics: Condensed Matter*, vol. 22, no. 32, 2010.

[18] A. Edelman, T. A. Arias, and S. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal of Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[19] R. W. Brockett, "Least square matching problems," *Linear Algebra and Its Applications*, vol. 122, no. 2, pp. 761–777, 1989.

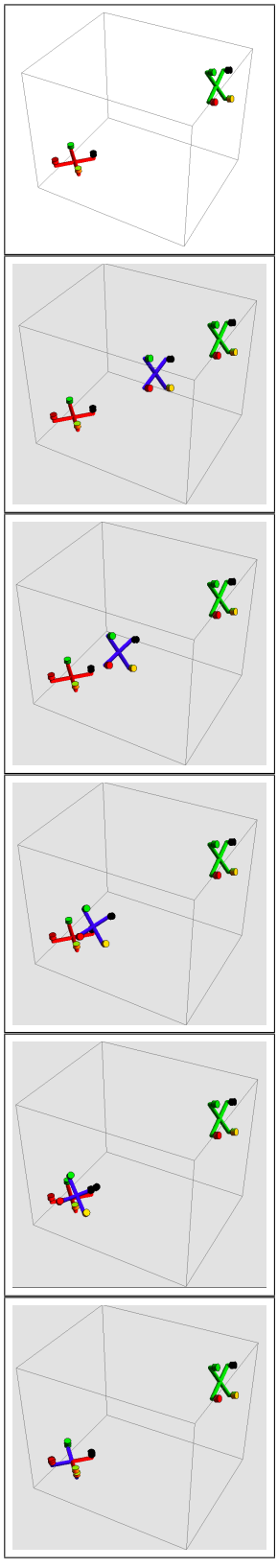

Fig. 2. Snapshots of the evolution of a steepest descent algorithm on $SO(3) \times \mathbb{R}^3$ for placing or "landing" the quad in the top panel, upper right, on a prescribed location given in the top panel, lower left. The blue figure traces the steps of the steepest descent algorithm starting from the second panel and ending at the bottom panel.