Simple Inverse Kinematics Using Ipopt

Stefano Dafarra

7 February 2017

1 Problem Statement

The problem consists in finding the joint angles, the so-called *robot shape* S, given the model and the desired transformation between two frames. It is possible to state this problem in a optimization framework, then solve it through the Ipopt solver.

1.1 Objective

The objective consists in defining the set of joint variables S in order to obtain the desired transformation between a *parent* frame and an *end effector* (or *target*) frame. It is supposed to have at least one joint between these two frames, with at least one degree of freedom (DoF) (so that S is a vector of at least one element). Notice that only 1 DoF joints are considered. The set of optimization variables coincide with S.

2 Notation

- $\dot{x}$ is the time derivative of x
- $\|\cdot\|_K^2$ is the K-weighted 2-norm.
- 1_n represents a $n \times n$ identity matrix. $0_{n\times m} \in \mathbb{R}^{n\times m}$ is a zero matrix while $0_n = 0_{n\times 1}$ is a zero column vector of size n.
- $\top$ indicates *transpose.*
- $*$ refers to a desired item (considered as a datum).

3 Cost Function

Given the desired transformation ${}^pH_t \in SE(3)$, it is possible to define a desired target position $p^* \in \mathbb{R}^3$ and a desired orientation ${}^pR_t \in SO(3)$, both expressed with respect to the *parent* frame. For what concern the rotation, it can be expressed in quaternion form $\mathbf{q}^*$, with the additional constraints of having the real part greater or equal than 0. Finally we can define $p_q^* \in \mathbb{R}^7, p_q^* = \left[\begin{smallmatrix} p^{*\top} & \mathbf{q}^{*\top} \end{smallmatrix}\right]^\top$.

Define $n \in \mathbb{N}$ as the number of joints (1 DoF each) considered when computing inverse kinematics (so that $S \in \mathbb{R}^n$), $K_f \in \mathbb{R}^{7\times 7}$ and $K_s \in \mathbb{R}^{n\times n}$ two symmetric positive semi-definite matrix of weights.

Finally the cost function is the following:

$$\min_S \frac{1}{2}\|p_q - p_q^*\|^2_{K_f} + \frac{1}{2}\|S - S^*\|^2_{K_s}. \tag{1}$$

Notice that the second term is a *regularization* term useful when $n > 6$: it drives the solution toward S^* in case of redundant manipulators. For what concerns the first term, considering the position part of p_q, the application of the 2-norm is straightforward. Considering the quaternion part, the 2-norm can be used as a metric for the rotation error (see Eq.(18) of [1], where the minimum operator can be avoided thanks to the assumption on the modulus of the quaternion).

We can rewrite Eq.(1) as

$$\min_S \frac{1}{2}(p_q - p_q^*)^\top K_f (p_q - p_q^*) + \frac{1}{2}(S - S^*)^\top K_s (S - S^*) = \tag{2a}$$

$$= \min_S \frac{1}{2}p_q^\top K_f p_q - \frac{1}{2}p_q^\top K_f p_q^* - \frac{1}{2}p_q^{*\top} K_f p_q + \frac{1}{2}p_q^{*\top} K_f p_q^* + \tag{2b}$$

$$\frac{1}{2}S^\top K_s S - \frac{1}{2}S^\top K_s S^* - \frac{1}{2}S^{*\top} K_s S + \frac{1}{2}S^{*\top} K_s S^*. \tag{2c}$$

Exploiting the assumption on the symmetry of the weights matrix and considering that nor $\frac{1}{2}p_q^{*\top} K_f p_q^*$ neither $\frac{1}{2}S^{*\top} K_f S^*$ depend on S, we can finally rewrite Eq.(2):

$$\min_S \frac{1}{2}p_q^\top K_f p_q - p_q^{*\top} K_f p_q + \frac{1}{2}S^\top K_s S - S^{*\top} K_s S. \tag{3}$$

Here p_q is obtained through forward kinematics from the shape S, thus:

$$\min_S \frac{1}{2}p_q(S)^\top K_f p_q(S) - p_q^{*\top} K_f p_q(S) + \frac{1}{2}S^\top K_s S - S^{*\top} K_s S. \tag{4}$$

3.1 Gradient

Defining

$$F = \frac{1}{2}p_q(S)^\top K_f p_q(S) - p_q^{*\top} K_f p_q(S) + \frac{1}{2}S^\top K_s S - S^{*\top} K_f S \tag{5}$$

we can obtain the gradient of the cost function as:

$$\frac{\partial F}{\partial S} = \left[\frac{1}{2}\left(MJ\right)^\top K_f p_q\right]^\top + \frac{1}{2}p_q^\top K_f MJ - p_q^{*\top} K_f MJ + \tag{6a}$$

$$+ \frac{1}{2}K_s^\top S + \frac{1}{2}K_s S - S^{*\top} K_f \tag{7}$$

where the first term is obtained by applying Eq.(50) of [2]. Exploiting again the symmetry of the weight matrices we obtain:

$$\frac{\partial F}{\partial S} = (p_q - p_q^*)^\top K_f M J + (S - S^*)^\top K_s \tag{8}$$

which can be transposed in

$$\left[\frac{\partial F}{\partial S}\right]^\top = \mathbb{G} = J^\top M^\top K_f (p_q - p_q^*) + K_s (S - S^*). \tag{9}$$

In the above equations, G defines the gradient of the cost function, J is the relative jacobian between the parent and the target frame, while M is the map between a twist and a 7-D vector composed by the linear velocity and the derivative of the quaternion.

4 Relative Jacobian

The relative Jacobian included in the gradient of Eq.(9) is the partial derivative of the forward kinematics function with respect ot S. Thus, it is necessary to pay attention on the formulation with which the Jacobian is expressed. In particular the *mixed* representation has to be adopted. Indeed, we don't need the time derivative, thus this representation fits our goal since it keeps the angular and linear velocity "decoupled". On the other hand, for computations, we will adopt the *body fixed* representation (*left trivialization*), having care of converting the Jacobian to the mixed one at end.

Define with ${}^dV_{c,d}$ the left-trivialized velocity of a frame d with respect to c. We can write it as a function of S:

$${}^dV_{c,d} = {}^dJ_{c,d}\dot{S} \tag{10}$$

where ${}^dJ_{c,d}$ is the *left-trivialized* relative Jacobian. Using the composition of velocities, it is possible to define ${}^dV_{c,d}$ having the relative velocities of c and d with respect to a third frame w. In particular:

$${}^dV_{c,d} = {}^dX_w {}^wV_{c,w} + {}^dV_{w,d} \tag{11}$$

where X represent an *adjoint* transformation. Here we can exploit the fact that

$${}^wV_{c,w} = -{}^wV_{w,c}. \tag{12}$$

Indeed we have the same frame with respect to the relative velocity is expressed. What changes is that the velocity of c is expressed with respect to w and not vice versa. Now the goal is to express all these relative velocities in *left trivialized* formulation, with respect to the world frame.

$${}^dV_{c,d} = -{}^dX_w {}^wV_{w,c} + {}^dV_{w,d} = -{}^dX_w \left({}^wX_c {}^cV_{w,c}\right) + {}^dV_{w,d}. \tag{13}$$

At this stage we can introduce the *left-trivialized* Jacobian expressed in world frame:

$$ {}^{d}J_{c,d}\dot{S} = -{}^{d}X_{c}\,{}^{c}J_{w,c}\dot{S} + {}^{d}J_{w,d}\dot{S}. \tag{14} $$

This equation should hold for any $\dot{S}$, so:

$$ {}^{d}J_{c,d} = -{}^{d}X_{c}\,{}^{c}J_{w,c} + {}^{d}J_{w,d}. \tag{15} $$

Now, recalling our initial goal of having the derivative of the forward kinematics with respect to S, the *mixed representation* which adopts the origin of d and the orientation of c, is effective, since it decouples the linear and angular velocity. For this conversion, it is necessary to multiply Eq.(15) by the appropriate adjoint transformation:

$$ {}^{d[c]}J_{c,d} = {}^{d[c]}X_{d}\,{}^{d}J_{c,d} = -{}^{d[c]}X_{c}\,{}^{c}J_{w,c} + {}^{d[c]}X_{d}\,{}^{d}J_{w,d}. \tag{16} $$

5 The Map M

The last term of Eq.(9) to be defined is the map M. It maps a twist into a 7-D vector composed by linear position and quaternion derivative. Taking in mind that the relative Jacobian of Eq. (16), provide a *right trivialized* angular velocity, M is the following

$$ M = \begin{bmatrix} \mathbb{1}_3 & 0_{3\times 3} \\ 0_{4\times 3} & \frac{1}{2}E^{\top} \end{bmatrix} \tag{17} $$

where E is defined in [3]. In particular, by defining a quaternion $\mathbf{q}$ as a 4-D vector $\mathbf{q} = [q_0, q_1, q_2, q_3]$ with q_0 the real part, E is the following:

$$ E = \begin{bmatrix} -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{18} $$

which can be written in a short form as:

$$ E = \begin{bmatrix} -Im(\mathbf{q}) & skew\left(Im(\mathbf{q})\right) + q_0 \cdot \mathbb{1}_3 \end{bmatrix} \tag{19} $$

where

$$ Im(\mathbf{q}) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad skew(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{20} $$

Remark Since the *mixed* representation of the Jacobian carries the angular velocity part in *right-trivialized* formulation, it is necessary to use E inside M. Otherwise, the matrix called G in [3] should have been employed.

6 Constraints

The only constraints to be involved in the formulation are the lower and upper bounds on the optimization variables S, defined through the model.

7 Alternative formulation

A possible alternative formulation consists in augmenting the optimization variables in order to contain even p_q. In other words, the set of optimization variables χ is the following:

$$\chi = \begin{bmatrix} p \\ \mathbf{q} \\ S \end{bmatrix}. \tag{21}$$

Using this augmented formulation, it is necessary to add a set of 7 constraints which relate p_q to S through forward kinematics. Thus, the increment on the number optimization variables is counterbalanced by the same number of constraints. With respect to the previous formulation, the cost function is easier to be formulated and to compute its first and second derivative. On the other hand, the "complexity" is just moved on the computation of the gradient of the constraints, while its Hessian is far from being trivial.

References

[1] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[2] R. J. Barnes, "Matrix differentiation," *Department of Civil Engineering, University of Minnesota*, 2014.

[3] B. Graf, "Quaternions and dynamics," *arXiv preprint arXiv:0811.2889*, 2008.