Optimal Control on Lie Groups: Implementation Details of the Projection Operator Approach

Alessandro Saccon * **John Hauser** ** **A. Pedro Aguiar** *

* *Institute for Systems and Robotics, Instituto Superior Técnico, Lisboa, Portugal (email: {asaccon, pedro}@isr.ist.utl.pt)*
** *Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, (email: john.hauser@colorado.edu)*

Abstract:
This paper discusses key implementation details required for computing the solution of a continuous-time optimal control problem on a Lie group using the *projection operator approach*. In particular, we provide the explicit formulas to compute the time-varying linear quadratic problem which defines the search direction step of the algorithm. We also show that the projection operator approach on Lie groups generates a sequence of adjoint state trajectories that converges, as a local minimum is approached, to the adjoint state trajectory of the first order necessary conditions of the Pontryagin's Maximum Principle, placing it between direct and indirect optimization methods.
As illustrative example, an optimization problem on SO(3) is introduced and numerical results of the projection operator approach are presented, highlighting second order converge rate of the method.

Keywords: Optimal control, Lie groups, direct method, rotation matrix

1. INTRODUCTION

In (Saccon et al., 2010a), the Lie group projection operator approach is introduced. The method is the extension to Lie groups of the projection operator approach for optimization of trajectory functionals developed in (Hauser, 2002).

Originally developed for unconstrained optimization problems, the strategy has been improved to handle optimal state transfer problems in (Hauser, 2003) and state-control constraints using a penalty function approach (Hauser and Saccon, 2006). The method has been used, e.g., to obtain a dynamic inversion procedure for the dynamics of racing motorcycles (Saccon et al., 2008, 2010b), to compute feasible trajectories for control-constrained systems (Notarstefano et al., 2007), for motion planning and control of automated marionettes, (Murphey and Egerstedt, 2007), and to explore the capabilities of a tilt-rotor VTOL aircraft (Notarstefano and Hauser, 2010).

The algorithm can be thought as a generalization of the Newton's method to the infinite dimensional setting and exhibits second order convergence rate to a local minimum. The method computes a search direction, at each step, solving a linear quadratic problem obtained from first and second derivatives of the incremental cost, terminal cost, and control system vector field. In this work, we provide explicit formulas to compute the linear dynamics and the quadratic incremental and terminal costs for the Lie group projection operator approach. Those formulas derive naturally from the theory descried in (Saccon et al., 2010a).

A quite interesting property of the algorithm in the flat case, is that it also generates a sequence of adjoint state trajectories that converges, as a local minimum is approached, to the adjoint state trajectory of the first order necessary conditions of the Pontryagin's Maximum Principle. In this work, we show that this property is retained in the Lie groups Projection Operator Approach. Notably, this fact places the algorithm between direct and indirect optimization methods.

The paper is organized as follows. Section 2 discusses the mathematical preliminaries and adopted notation. The projection operator approach, both on flat spaces and Lie groups, is reviewed in Section 3. Explicit formulas to compute the search direction in the Lie group projection operator approach are detailed in Section 4. As illustrative example of these formulas, an optimal control problem on the Lie group SO(3) is presented in Section 5, together with numerical results. Conclusions are drawn in Section 6.

2. MATHEMATICAL PRELIMINARIES

In this section, we introduce standard definitions and notation that will be used throughout the paper. We assume that the reader is familiar with the theory of finite dimensional smooth manifolds, matrix Lie groups,

⋆ This work was supported in part by projects CONAV/FCT-PT (PTDC/EEA-CRO/113820/2009), Co3-AUVs (EU FP7 no. 231378), FCT-ISR/IST plurianual funding program, and the CMU-Portugal program. The first author benefited from a postdoctoral scholarship of FCT. The work of the second author was supported in part by AFOSR FA9550-09-1-0470 and by an invited scientist grant from the Foundation for Science and Technology (FCT), Portugal.

covariant differentiation. We refer to the books (Boothby, 1986), (Abraham et al., 1988), (Lee, 1997) for a review on differentiable manifolds and covariant differentiation and to (Varadarajan, 1984), (Rossmann, 2002) for a review of the theory of Lie groups and Lie Algebra. A *smooth manifold* will be indicated with the letter M or N. A point on the manifold will be denoted simply by x. T_xM and T^*_xM denote, respectively, the *tangent* and *cotangent spaces* of M at x. A generic tangent vector is usually written as v_x or w_x, where the subscript indicates the base point at which the tangent vectors are attached. The *tangent* and *cotangent bundles* of M are denoted by TM and T^*M, respectively. The *natural bundle projection* from TM to M is denoted by $\pi : TM \to M$, so that $\pi \mathrm{v}_x = x$. A generic vector field on a manifold M is denoted by $X : M \to TM$. A vector field X is a *section* of the tangent bundle TM, that is, it satisfies $\pi X(x) = x$.

Given a function $f : M \to N$, its *tangent map* is represented by $\mathbf{D}f : TM \to TN$ (or also as $Tf : TM \to TN$). Tangent maps act naturally on tangent vectors. Given a vector $\mathrm{v}_x \in T_xM$, $\mathbf{D}f(x) \cdot \mathrm{v}_x \in T_{f(x)}N$ (or $T_xf(\mathrm{v}_x)$) is the evaluation of the tangent map of f in the direction v_x at x. Tangent maps act naturally on vector fields as well. Given a vector field $X : M \to TM$, the writing $\mathbf{D}f \cdot X : M \to TN$ (or $Tf(X) : M \to TN$) denotes at $x \in M$ the tangent vector $\mathbf{D}f(x) \cdot X(x) \in T_{f(x)}N$ (or $T_xf(X(x))$). Given an affine connection ∇ on a manifold M, we write $\nabla_X Y$ and D_t to indicate respectively, the covariant derivative of the vector field Y in the direction X and the covariant differentiation with respect to the parameter t. The *parallel displacement* along a curve $\gamma(t)$, $t \in I$, from $t = t_0$ to $t = t_1$ of a vector $V_0 \in T_{\gamma(t_0)}M$ is represented by $P_\gamma^{t_1 \leftarrow t_0} V_0$. We also adopt the notation $\mathbb{D}Y \cdot X$ to mean $\nabla_X Y$.

The symbol $\mathbb{D}^2 f(x) \cdot (v_x, w_x)$ is the *second geometric derivative* of a function $f : M \to N$ at $x \in M$ evaluated in the directions v_x, $w_x \in TM$ (Saccon et al., 2010a). Given two connections ${}^1\nabla$ and ${}^2\nabla$ defined on M and N, respectively, the second geometric derivative is defined as

$$\mathbb{D}^2 f(x) \cdot (v_x, w_x) =$$
$$\lim_{h\to 0} 1/h \left({}^2P^{0\leftarrow h}_{f\circ\gamma} \mathbf{D}f(\gamma(h)) \cdot {}^1P^{h\leftarrow 0}_{\gamma} v_x - \mathbf{D}f(\gamma(0)) \cdot v_x \right),$$

where $\gamma : (-\varepsilon, \varepsilon) \to M$, $\varepsilon > 0$, is a differentiable curve satisfying $\gamma(0) = x$ and $\dot\gamma(0) = w_x$, and ${}^1P_\gamma$ and ${}^2P_\gamma$ are the parallel displacements associated to the connections ${}^1\nabla$ and ${}^2\nabla$, respectively. Note that $\mathbb{D}^2 f(x) \cdot (v_x, w_x) \in T_{f(x)}N$.

When f is a real valued function, $f : M \to \mathbb{R}$, then $\mathbb{D}^2 f$ reduces to the *second covariant derivative* (Absil et al., 2008, Sections 5.6 and 5.7).

A generic *Lie group* is denoted by G. The group *identity* is denoted by e. *Left* and *right translations* of $x \in G$ (a group element) by the group element $g \in G$ are denoted by $L_g x$ and $R_g x$, respectively. When convenient, we will adopt the shorthand notation gx, xg, $g\mathrm{v}_x$, $\mathrm{v}_x g$ for, in the same order, $L_g x$, $R_g x$, $T_x L_g(\mathrm{v}_x)$ and $T_x R_g(\mathrm{v}_x)$. A *left-invariant vector field* on G is a vector field such that $X(L_g x) = T_x L_g(X(x))$. Given a tangent vector at the identity $\varrho \in T_e G$, the symbol X_ϱ means the left-invariant vector field defined by $X_\varrho(g) := T_e L_g(\varrho)$. The *Lie algebra* of G is $\mathfrak{g}$. The Lie algebra $\mathfrak{g}$ is identified with the tangent space $T_e G$ endowed with the *Lie bracket* operation $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, defined by $[\varrho, \varsigma] := [X_\varrho, X_\varsigma](e)$, where the latter bracket is the Jacobi-Lie bracket of the left-invariant vector fields X_ϱ and X_ς evaluated at the group identity.

The mapping $I_g(x) = gxg^{-1}$ is called *inner automorphism*. The *adjoint representation* of the Lie group G on the algebra $\mathfrak{g}$ is written as $\mathrm{Ad}_g : \mathfrak{g} \to \mathfrak{g}$ and is the tangent map obtained differentiating $I_g(x)$ with respect to x at $x = e$. We recall that $\mathfrak{g} = T_e G$. Furthermore, the *adjoint representation* of the Lie algebra $\mathfrak{g}$ onto itself is written as $\mathrm{ad}_\varrho : \mathfrak{g} \to \mathfrak{g}$ and it is obtained differentiating $\mathrm{Ad}_g(\varsigma)$ with respect to g at $g = e$, in the direction ϱ. We recall that $\mathrm{ad}_\varrho \varsigma = [\varrho, \varsigma]$. The *exponential map* is denoted by $\exp : \mathfrak{g} \to G$ and its inverse (in a neighborhood of the identity) by $\log : G \to \mathfrak{g}$.

3. THE PROJECTION OPERATOR APPROACH

The projection operator approach on a flat space and on a Lie group is reviewed in this section.

3.1 Flat space projection operator approach

The projection operator approach to the optimization of trajectory functionals, developed in (Hauser, 2002), allows one to perform local Newton optimization of the (integral plus terminal) cost functional

$$h(x, u) := \int_0^T l(\tau, x(\tau), u(\tau))\, d\tau + m(x(T)) \qquad (1)$$

over the set $\mathcal{T}$ of trajectories (i.e., state and control pairs) of a nonlinear system $\dot x = f(x, u)$ subject to a fixed initial condition x_0.

Restricting our attention to the set $\mathcal{T}$ of *exponentially stabilizable* trajectories, one can show that $\mathcal{T}$ has the structure of a (infinite dimensional) Banach manifold (Hauser and Meyer, 1998), allowing us to use vector space operations (Luemberger, 1969) to effectively explore it.

To work on the trajectory manifold $\mathcal{T}$, one *projects* curves ξ in the ambient Banach space onto the trajectory manifold, giving $\eta = \mathcal{P}(\xi) \in \mathcal{T}$, by using a local linear time-varying trajectory tracking controller. Suppose that $\xi(t) = (\alpha(t), \mu(t))$, $t \geq 0$, is a bounded curve (e.g., an approximate trajectory of f) and let $\eta(t) = (x(t), u(t))$, $t \geq 0$, be the trajectory of f determined by the nonlinear feedback system

$$\begin{aligned} \dot x(t) &= f(x(t), u(t)), \qquad x(0) = \alpha(0), \\ u(t) &= \mu(t) + K(t)(\alpha(t) - x(t)) . \end{aligned} \qquad (2)$$

Under certain (mild) conditions on f and K (Hauser and Meyer, 1998), this feedback system defines a continuous, nonlinear *projection operator*

$$\mathcal{P} : \xi = (\alpha, \mu) \mapsto \eta = (x, u) .$$

It is straightforward to see that, independent of K, if ξ is a trajectory of f, then ξ is a fixed point of $\mathcal{P}$, $\xi = \mathcal{P}(\xi)$. Noting that the constrained and unconstrained optimization problems

$$\min_{\xi \in \mathcal{T}} h(\xi) \quad \text{and} \quad \min_{\xi} h(\mathcal{P}(\xi))$$

are *essentially* locally equivalent, one may develop Newton and quasi-Newton descent methods for trajectory optimization in an effectively unconstrained manner by working with the cost functional $\tilde{h}(\xi) := h(\mathcal{P}(\xi))$. By saying that the problem are essentially locally equivalent, we mean the following: If $\xi^* \in \mathcal{T}$ is a constrained local minimum of h, then it is an unconstrained local minimum of $\tilde{h}$. If ξ^+ is an unconstrained local minimum of $\tilde{h}$, then $\xi^* = \mathcal{P}(\xi^+)$ is a constrained local minimum of $\mathcal{T}$. This observation was the basis for the development of the following Newton algorithm for the optimization of h over $\mathcal{T}$ (Hauser, 2002):

Algorithm (Projection operator Newton method)
given initial trajectory $\xi_0 \in \mathcal{T}$
for $i = 0, 1, 2, \ldots$
redesign feedback K if desired/needed
(search direction)

$$\zeta_i = \arg \min_{\zeta \in T_{\xi_i}\mathcal{T}} \mathbf{D}\tilde{h}(\xi_i) \cdot \zeta + \tfrac{1}{2} \mathbf{D}^2 \tilde{h}(\xi_i) \cdot (\zeta, \zeta) \tag{3}$$

$$\gamma_i = \arg \min_{\gamma \in (0,1]} \tilde{h}(\xi_i + \gamma \zeta_i) \qquad \text{(step size)} \tag{4}$$

$$\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i) \qquad \text{(update)} \tag{5}$$

end

The local Newton update, valid in a neighborhood of a second order sufficient local minimum, is given by (5) where the *search direction* ζ_i is the solution of the (finite horizon, time-varying) linear quadratic (LQ) optimal control problem (3).

In the *flat* Banach space case, the usual chain rule applies and one finds that

$$\mathbf{D}^2 \tilde{h}(\xi) \cdot (\zeta, \zeta) = \mathbf{D}^2 h(\xi) \cdot (\zeta, \zeta) + \mathbf{D}h(\xi) \cdot \mathbf{D}^2 \mathcal{P}(\xi) \cdot (\zeta, \zeta)$$

(for $\xi \in \mathcal{T}$, $\zeta \in T_\xi \mathcal{T}$) is a well defined object; see (Hauser and Meyer, 1998) for some projection operator calculus. The solution of the above LQ problem involves first and second order approximations of the nonlinear system about a given trajectory as well as the solution of the associated Riccati equations.

3.2 Lie group projection operation approach

When the system evolves on a Lie group, a number of interesting questions arise. What is the linearization of the system? How do we define and compute a second order approximation of the system? In (Saccon et al., 2010a), the authors have shown how these questions can be addressed, extending the Projection Operator approach for a dynamical system defined on a Lie group G, that is

$$\dot{g} = f(g, u, t) = g(t)\lambda(g(t), u(t), t)\,, \tag{6}$$

where $f : G \times \mathbb{R}^m \times \mathbb{R} \to TG$ and $\lambda(g, u, t) := g^{-1} f(g, u, t)$ is the *left-trivialization* of f. In this case, the projection operator $\mathcal{P}$ is redefined as

$$\begin{aligned} \dot{g}(t) &= g(t)\lambda(g(t), u(t), t)\,, \quad g(0) = \alpha(0)\,, \\ u(t) &= \mu(t) + K(t)\big[\log(g(t)^{-1}\alpha(t))\big]\,, \end{aligned} \tag{7}$$

where $K(t) : \mathfrak{g} \to \mathbb{R}^m$ is a linear operator, which can be thought as a standard linear feedback as soon as a basis is chosen for the Lie algebra $\mathfrak{g}$. The authors have shown that given a *trajectory* $\xi(t) = (g(t), u(t))$ of the closed-loop system (7) one can defined its (left-trivialized) linearization as the curve $(z(t), v(t)) \in \mathfrak{g} \times \mathbb{R}^m$, $t \geq 0$, satisfying

$$\begin{aligned} \dot{z}(t) &= A(\xi(t), t)z(t) + B(\xi(t), t)v(t)\,, \\ v(t) &= -K(t)z(t)\,, \end{aligned} \tag{8}$$

with A and B equal to

$$A(\xi, t) := \mathbf{D}_1 \lambda(g, u, t) \circ TL_g - \mathrm{ad}_{\lambda(g,u,t)}\,, \tag{9}$$

$$B(\xi, t) := \mathbf{D}_2 \lambda(g, u, t)\,. \tag{10}$$

Finally, the following Newton method is proposed:

Algorithm (Projection operator Newton method)
given initial trajectory $\xi_0 \in \mathcal{T}$
for $i = 0, 1, 2, \ldots$
redesign feedback K if desired/needed
(search direction)

$$\zeta_i = \arg \min_{\xi_i\zeta \in T_{\xi_i}\mathcal{T}} \mathbf{D}h(\xi_i) \cdot \xi_i\zeta + \tfrac{1}{2} \mathbb{D}^2 \tilde{h}(\xi_i) \cdot (\xi_i\zeta, \xi_i\zeta) \tag{11}$$

$$\gamma_i = \arg \min_{\gamma \in (0,1]} \tilde{h}(\xi_i \exp(\gamma \zeta_i)) \qquad \text{(step size)} \tag{12}$$

$$\xi_{i+1} = \mathcal{P}(\xi_i \exp(\gamma_i \zeta_i)) \qquad \text{(update)} \tag{13}$$

end

In (Saccon et al., 2010a), the derivation of the above Newton algorithm requires to define the concept of *second geometric derivative*, which is related to the problem of constructing a second order approximation of a function between two smooth manifolds M_1 and M_2. The second geometric derivative of a function f reduces to the *second covariant derivative* when $M_2 = \mathbb{R}$ (that is, when f is a real function). See Section 2 for the definition of the second geometric and covariant derivatives.

As detailed in (Saccon et al., 2010a), these derivatives are computed using the (0) connection, whose parallel displacement along a differentiable curve $\gamma : \mathbb{R} \to G$ for a generic Lie group G is defined as

$$P_\gamma^{t_1 \leftarrow t_0} \mathrm{v}_0 = 1/2\,(\,x_1 x_0^{-1} \mathrm{v}_0 + \mathrm{v}_0 x_0^{-1} x_1)\,, \quad v_0 \in T_{x_0}G\,, \tag{14}$$

with $t_0,\ t_1 \in \mathbb{R}$, $\gamma(t_0) = x_0$, $\gamma(t_1) = x_1$.

At each iterate, the search direction minimization (11) is performed on the tangent space to the trajectory manifold (that is, we search over the curves $\zeta(\cdot) = (z(\cdot), v(\cdot))$ that satisfies (8)). Then, the step size subproblem (12) is considered. As in the flat space case, the classical *approximate* solution obtained using backtracking line search with Armijo condition (Nocedal and Wright, 1999., Chapter 3) can be used to compute γ_i. Once γ_i has been computed, the update step (13) *projects* each iterate on to the trajectory manifold and the process restarts as long as termination conditions have not been met. Note that, when $G = \mathbb{R}^n$, the algorithm is equivalent to the algorithm introduced in (Hauser, 2002) and reviewed in Section 3.1.

In the next section, we discuss the details of the search direction subproblem.

4. EXPLICIT FORMULAS TO COMPUTE THE SEARCH DIRECTION

The search direction subproblem (11) requires the minimization of the functional $\mathbf{D}h(\xi) \cdot \xi\zeta + \frac{1}{2} \mathbb{D}^2 \tilde{h}(\xi) \cdot (\xi\zeta, \xi\zeta)$ over the Banach space $T_\xi \mathcal{T}$. In (Saccon et al., 2010a), the authors have shown that when $\xi \in \mathcal{T}$ and $\xi\zeta \in T_\xi\mathcal{T}$, the term $\mathbb{D}^2 \tilde{h}(\xi) \cdot (\xi\zeta, \xi\zeta)$ is equal to

$$\mathbb{D}^2 h(\xi) \cdot (\xi\zeta, \xi\zeta) + \mathbf{D}h(\xi) \cdot \mathbb{D}^2 \mathcal{P}(\xi) \cdot (\xi\zeta, \xi\zeta)\,. \tag{15}$$

Let $\mathbf{e}_i$, $i = 1, \ldots, n+m$, be a basis for $\mathfrak{g} \times \mathbb{R}^m$. Each $(z, v) \in \mathfrak{g} \times \mathbb{R}^m$ can be uniquely written as $(z, v) = z^1 \mathbf{e}_1 +$

$\cdots + z^n \mathbf{e}_n + v^1 \mathbf{e}_{n+1} + \cdots + v^m \mathbf{e}_{n+m}$. Given $\xi \in G \times \mathbb{R}^m$ define $l_{ij}(\xi, t) \in \mathbb{R}$ as

$$l_{ij}(\xi, t) := \mathbb{D}_1^2 l(\xi, t) \cdot (\xi \mathbf{e}_i, \xi \mathbf{e}_j) \tag{16}$$

and $\lambda_{ij}(\xi) \in \mathfrak{g}$ as

$$\lambda_{ij}(\xi) := \mathbb{D}^2 \lambda(\xi) \cdot (\xi \mathbf{e}_i, \xi \mathbf{e}_j) \tag{17}$$

$$+ 1/2 \left(\mathrm{ad}_{\varpi_1(\mathbf{e}_i)} \mathbf{D}\lambda(\xi) \cdot \xi \mathbf{e}_j + \mathrm{ad}_{\varpi_1(\mathbf{e}_j)} \mathbf{D}\lambda(\xi) \cdot \xi \mathbf{e}_i \right), \tag{18}$$

where $\varpi_1 : \mathfrak{g} \times \mathbb{R}^m \to \mathfrak{g}$, $\varpi_1(z, v) = z$. The following key result can be proven using the same technique presented in the proof of Proposition 3.2 in (Hauser, 2002), replacing the flat space expressions with those presented in (Saccon et al., 2010a).

Proposition 4.1. Given $\xi \in \mathcal{T}$ and $\xi\zeta \in T_\xi \mathcal{T}$, with $\xi(t) = (g(t), u(t))$, $t \geq 0$, and $\zeta(t) = (z(t), v(t))$, $t \geq 0$, the quadratic form (15) can be computed as

$$\int_0^T \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix}^T W(\tau) \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix} d\tau + z(T)^T P_1 z(T), \tag{19}$$

where P_1 is the symmetric $n \times n$ matrix representing $\mathbb{D}^2 m(g(T)) \cdot (g(T)(\cdot), g(T)(\cdot))$ and $W(t) = [w_{ij}(t)]$ is the bounded symmetric $(n+m) \times (n+m)$ matrix with elements given by

$$w_{ij}(t) = l_{ij}(\xi(t), t) + \sum_{k=1}^{n} p_k(t) \lambda_{ij}^k(\xi(t)), \tag{20}$$

where $l_{ij}(\xi(t), t)$ is given by (16), $\lambda_{ij}^k(\xi(t))$ is the k-th component of λ_{ij} in (17), and $p^k(t)$ is the k-th component of the *adjoint* state trajectory $p(t) \in \mathfrak{g}^*$, $t \geq 0$, defined below, relative to the dual basis of $\mathbf{e}_i$, $i = 1, \ldots, n$. The adjoint state $p(t) \in \mathfrak{g}^*$ satisfies the differential equation

$$-\dot{p}(t) = A_{cl}(t)^T p(t) + a(t) - K(t)^T b(t), \tag{21}$$

$$p(T) = a_1, \tag{22}$$

where $A_{cl}(t) := A(\xi(t)) - B(\xi(t)) K(t)$, and $a(t)$, $b(t)$ and a_1 are the vector representation of the pairings $\langle a(t), z \rangle = \mathbf{D}_1 l(g(t), u(t), t) \cdot g(t) z$, $\langle b(t), v \rangle = \mathbf{D}_2 l(g(t), u(t), t) \cdot v$, and $\langle a_1, z \rangle = \mathbf{D} m(g(T)) \cdot g(T) z$.

Remark. Equation (21) is a "stabilized" version of the adjoint equation associated to the left-trivialized pre-Hamiltonian $\hat{H}^-(g, p, u) := l(g, u) + \langle p, \lambda(g, u) \rangle$, which is naturally associated to the optimal control problem of our interest. Indeed, the necessary conditions for optimality of the (left-trivialized) Pontryagin Maximum Principle are

$$g^{-1} \dot{g} = \frac{\partial \hat{H}}{\partial p}^{-} (g, p, u^*(g, p)) \tag{23}$$

$$\dot{p} = \mathrm{ad}^*_{g^{-1}\dot{g}} p - (T L_g)^* \frac{\partial \hat{H}}{\partial g}^{-} (g, p, u^*(g, p)) \tag{24}$$

$$u^*(g, p) = \arg\min_u \hat{H}^-(g, p, u), \tag{25}$$

with boundary conditions $g(0) = g_0$ and $p(T) = a_1$ (see, e.g., (Jurdjevic, 1997, Chapter 12, Corollary 1)). Recalling the definition of $A(\xi(t))$ and $a(t)$, it is straightforward to verify that (24) equals $-\dot{p} = A^T(\xi(t)) p + a(t)$. Note that (21), instead, is equal to $-\dot{p} = A^T(\xi(t)) p + a(t) - K^T(t)(b(t) + B(\xi(t))^T p)$. The necessary condition (25) implies $\partial \hat{H}^-(g, p, u^*(g, p))/\partial u = 0$, i.e., $b^T(t) + p^T(t) B(\xi(t)) = 0$. Therefore, approaching a (local) optimal solution, $p(t)$ in (21) converges to the solution of (24), since $b^T(t) + p^T(t) B(\xi(t))$ tends to zero. ■

We can now write the subproblem (11) in matrix form. Since $\mathbf{D}h(\xi) \cdot \xi\zeta$ is equal to

$$\int_0^T a(\tau)^T z(\tau) + b(\tau)^T v(\tau) \, dt + a_1^T z(T), \tag{26}$$

the subproblem (11) is equivalent to solving the optimal control problem

$$\min_{(z,v)(\cdot)} \int_0^T a(\tau)^T z(\tau) + b(\tau)^T v(\tau) + \frac{1}{2} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix}^T W(\tau) \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix} dt$$

$$+ a_1^T z(T) + \frac{1}{2} z(T)^T P_1 z(T), \tag{27}$$

subject to the dynamic constraint

$$\dot{z}(t) = A(\xi(t)) z(t) + B(\xi(t)) v(t), \tag{28}$$

$$z(0) = 0. \tag{29}$$

The above linear quadratic optimal control problem is solvable by standard techniques (see, e.g., (Anderson and Moore, 1989)) and it amounts to solving a Riccati differential equation backward in time, from which a time-varying *affine* state feedback can be derived.

5. OPTIMAL CONTROL ON SO(3)

This section presents an optimal control problem for a system evolving on the *nonabelian* Lie group SO(3), together with the numerical results obtained by using the algorithm detailed in Section 3.2 to solve it. In particular, we show how to form the linear quadratic optimal control problem (27), giving explicit formulae to compute the matrices A, B, W, and P_1 and the vectors a, b, and a_1. Also, we show that the algorithm exhibits second order convergence rate.

5.1 Problem formulation

Let $\|M\|_P$ denotes, with M and $P \in \mathbb{R}^{3\times 3}$ and $P = P^T > 0$, the weighted *Frobenius* matrix norm defined as $\sqrt{\mathrm{tr}(M^T P M)}$. Let $\big(g_d(t), u_d(t)\big) \in \mathrm{SO}(3) \times \mathbb{R}^3$, $t \in [0, T]$, be a desired (approximate) state-control curve. Let $\bar{Q}$, $\bar{R}$, and $\bar{P}_f \in \mathbb{R}^{3\times 3}$ be symmetric positive define matrices and g_0 and g_f two elements of SO(3). We define the *hat* operator $\wedge : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 3}$ as the Lie algebra isomorphism

$$\mathbb{R}^3 \ni \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \mapsto \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \in \mathfrak{so}(3). \tag{30}$$

The goal is to *minimize* the integral plus terminal cost

$$\int_0^T l(g, u, \tau) \, d\tau + m(g(T)), \tag{31}$$

with *incremental cost*

$$l(g, u, \tau) := \frac{1}{2} \|e - g_d^{-1}(\tau) \, g\|_{\bar{Q}}^2 + \frac{1}{2} \|u - u_d(\tau)\|_{\bar{R}}^2 \tag{32}$$

and *terminal cost*

$$m(g) := \frac{1}{2} \|e - g_f^{-1} g\|_{\bar{P}_f}^2 \tag{33}$$

over the set of curves $(g(t), u(t)) \in \mathrm{SO}(3) \times \mathbb{R}^3$, $t \in [0, T]$, subject to the *dynamic constraint*

$$\dot{g}(t) = g(t) \hat{u}(t), \tag{34}$$

$$g(0) = g_0. \tag{35}$$

Since (34) is already in the left-trivialized form (6) with $\lambda(g,u) = u$, given a trajectory $\xi(t) = (g(t),u(t))$, $t \in [0,T]$, its left-trivialized linearization is

$$\begin{aligned} A(\xi(t)) &:= \mathbf{D}_1\lambda(g(t),u(t)) \circ TL_{g(t)} - \mathrm{ad}_{\lambda(g(t),u(t))} \\ &= -\mathrm{ad}_{u(t)} = -\hat{u}(t)\,, \end{aligned} \tag{36}$$

$$B(\xi(t)) := \mathbf{D}_2\lambda(g(t),u(t)) = I\,. \tag{37}$$

The expression for the vectors $a(t)$, $b(t)$ and a_1 are

$$a(t)^T z = \mathbf{D}_1 l(g(t),u(t),t)\cdot g(t)\hat{z} = -\mathrm{tr}(\bar{Q} g_d^T(t) g(t)\hat{z})\,,$$
$$b(t)^T v = \mathbf{D}_2 l(g(t),u(t),t)\cdot v = (u(t)-u_d(t))^T \bar{R} v\,,$$
$$a_1^T z = \mathbf{D}_1 m(g(T))\cdot g(T)\hat{z} = -\mathrm{tr}(\bar{P}_f {g_f}^T g(T)\hat{z})\,.$$

The matrices $W(t)$ and P_1 can be computed once the second covariant derivative of the function

$$F(g) := \frac{1}{2}\|e - g_1^{-1} g\|^2_{\bar{P}}\,, \tag{38}$$

with $g_1 \in \mathrm{SO}(3)$ and $\bar{P} = \bar{P}^T > 0$, is known. Note how the function $F(g)$ appears in the expressions of the incremental and terminal costs. The first and second covariant derivatives of $F(g)$ are given by

$$\mathbf{D}F(g)\cdot gz = -\mathrm{tr}(\bar{P} g_1^T g\hat{z})\,, \tag{39}$$

$$\mathbb{D}^2 F(g)\cdot(gz_1, gz_2) = -\mathrm{tr}\left(\bar{P} g_1^T g \frac{\hat{z}_2\hat{z}_1 + \hat{z}_1\hat{z}_2}{2}\right)\,. \tag{40}$$

In principle, one could obtain the vector and matrix representations of the above derivatives by using the identities $\mathrm{tr}(\hat{x}^T A) = x^T(A - A^T)^\vee$ and $\mathrm{tr}(\hat{x}^T A\hat{y}) = y^T((\mathrm{tr}A)I - A)x$, valid, as direct calculation shows, for each $x, y \in \mathbb{R}^3$ and $A \in \mathbb{R}^{3\times 3}$ (the *vee* operator $^\vee$ is just the inverse of the *hat* operator $^\wedge$ defined in (30)). However, we found that a simpler and more elegant expressions for those derivatives can be obtained by using unit quaternions. Define the matrix P according to the transformation

$$P = (\mathrm{tr}\bar{P})I - \bar{P}\,, \tag{41}$$

with inverse

$$\bar{P} = (1/2\,\mathrm{tr}P)I - P\,, \tag{42}$$

and let $q \in \mathbb{R}^4$ be one of the two unit quaternions corresponding to the rotation matrix $g_1^{-1}g$. Let $q_s \in \mathbb{R}$ and $q_v \in \mathbb{R}^3$ denote, respectively, the scalar and vector parts of the unit quaternion $q = (q_s, q_v^T)^T$. Remarkably, the following identity holds

$$F(g) = \frac{1}{2}(2q_v)^T P(2q_v)\,. \tag{43}$$

Note that the formula is, as it has to in order to be a function defined on SO(3), invariant under the antipodal symmetry $(q_s, q_v) \mapsto (-q_s, -q_v)$. From (43), $\mathbf{D}F(g)\cdot g\hat{z}$ and $\mathbb{D}^2F(g)\cdot(g\hat{z}_1, g\hat{z}_2)$ equal respectively

$$2\,q_v^T P(q_s I + \hat{q}_v)z\,, \tag{44}$$

and

$$z_2^T((q_s I + \hat{q}_v)^T P(q_s I + \hat{q}_v) - (q_v^T P q_v)I)z_1\,. \tag{45}$$

Due to space limitations, we will not provide a formal proof of these formulas. They can be easily checked numerically against the equivalent expressions (39) and (40).

Equations (44) and (45) provide immediately the vector and matrix representations that we need to compute the matrices $W(t)$, $t \in [0,T]$, and P_1. Define $Q = Q^T > 0$ from $\bar{Q}$ according to (41). Using (20), we see that $W(t)$ equals

$$\begin{bmatrix} (q_s I + \hat{q}_v)^T Q(q_s I + \hat{q}_v) - (q_v^T Q q_v)I & -1/2\hat{p}(t) \\ 1/2\hat{p}(t) & \bar{R} \end{bmatrix}\,, \tag{46}$$

where $q = (q_s, q_v^T)^T$ is the unit quaternion representation of $g_d(t)^{-1}g(t)$. Equation (46) has been obtained as follows. Let $\zeta_{1,k}$ and $\zeta_{2,k}$, k=1, …, $n+m$ be the components of $\zeta_1 = (z_1, v_1)$ and $\zeta_2 = (z_2, v_2) \in \mathfrak{g}\times\mathbb{R}^m$, with respect the basis $\mathbf{e}_k$, $k = 1,\ldots,n+m$. The diagonal entries of $W(t)$ in (46) derive from the matrix representation of $\mathbb{D}^2 l(\xi(t),t)\cdot(\xi(t)\zeta_1, \xi(t)\zeta_2) = l_{ij}(\xi(t),t)\zeta_{1,i}\zeta_{2,j}$, which is obtained, concerning the state part, from (45). The off diagonal terms are obtained computing $p_k(t)\lambda^k_{ij}(\xi(t),t)\zeta_{1,i}\zeta_{2,j}$ which is equal to $\langle p(t), 1/2\left(\mathrm{ad}_{z_1}v_2 + \mathrm{ad}_{z_2}v_1\right)\rangle$, because $\mathbb{D}^2\lambda(\xi) \equiv 0$. Finally, $P_1 = (q_s I + \hat{q}_v)^T P_f(q_s I + \hat{q}_v) - (q_v^T P_f q_v)I$, where (q_s, q_v) is the unit quaternion representation of $g_f^{-1}g(T)$.

5.2 Numerical results

In Figure 1, we show the optimal solution obtained by applying the descent algorithm detailed in Section 3.2 to the problem (31)–(35). The following set of parameters is chosen. The time horizon is $T = 20$s and the initial condition g_0 is the rotational matrix corresponding to the unit quaternion $[0.7986\,, 0.2457\,, -0.2457\,, 0.4914]^T$. The desired trajectory $\xi_d(t) = (g_d(t), u_d(t))$, $t \in [0,T]$, appearing in the incremental cost (32), is the trivial trajectory $(e,0)$, $t \in [0,T]$. The weighting matrix $\bar{Q}$ is equal to $\bar{Q} = (1/2\,\mathrm{tr}\,Q)I - Q$, the inverse of the transformation (41), with $Q = \mathrm{diag}(2\,,5\,,3)$. The weighting matrix $\bar{R}$ is equal to $\mathrm{diag}(1\,,6\,,3)$. The rotational matrix g_f in the terminal cost (33) corresponds to the unit quaternion $q_f = [0.2673\,, 0.5345\,, 0\,, 0.8018]^T$, while the weighting matrix $\bar{P}_f$ is obtained from $P_f = \mathrm{diag}(20, 20, 20)$, in the same way $\bar{Q}$ is obtained from Q.

The initial trajectory $\xi(t) = (g(t), u(t))$, $t \in [0,T]$, is the constant trajectory $(g_0, 0)$, $t \in [0,T]$. At each iteration, the projection operator feedback $K(t)$, $t \in [0,T]$, is designed solving a standard LQR problem (with diagonal weighting matrices, both equal to the identity).

To integrate the differential equations required at each iteration of the algorithm, we used the function `ode45` of Mathworks Matlab, storing all the trajectories with a sampling period of 0.01s. The absolute and relative tolerances of the ODE solver is set to 10^{-8} and 10^{-6}, respectively. The termination condition is $\mathbf{D}h(\xi_k)\cdot\xi_k\zeta_k \approx h(\xi_{k+1}) - h(\xi_k) \leq 10^{-8}$. The backtracking line search algorithm parameters (see Section 3.2), are set to $\bar{\gamma} = 1$, $c = 0.4$, and $\rho = 0.7$. The algorithm takes about 2 seconds to run on a laptop equipped with a Intel Core 2 Duo CPU P8600 2.40 GHz. The algorithm is coded as a main m-function which calls a series of S-functions written in `C`.

Figure 2 shows that the algorithm takes 5 steps to converge. In the first iteration, we have observed that the backtracking line search takes 5 iterations as the quadratic approximation of the functional does not approximate accurately the cost functional in the optimal search direction ζ_k. However, starting from the second iteration, full lengths steps are taken ($\gamma_k = 1$) and the algorithm shows a quadratic convergence rate. Quadratic convergence rate in the neighborhood of a local minimum satisfying second order sufficient conditions has been formally proven for the flat space case in Proposition 5.1 in (Hauser, 2002). For Lie groups, the proof is under development, but we suspect it to be a straightforward extension.

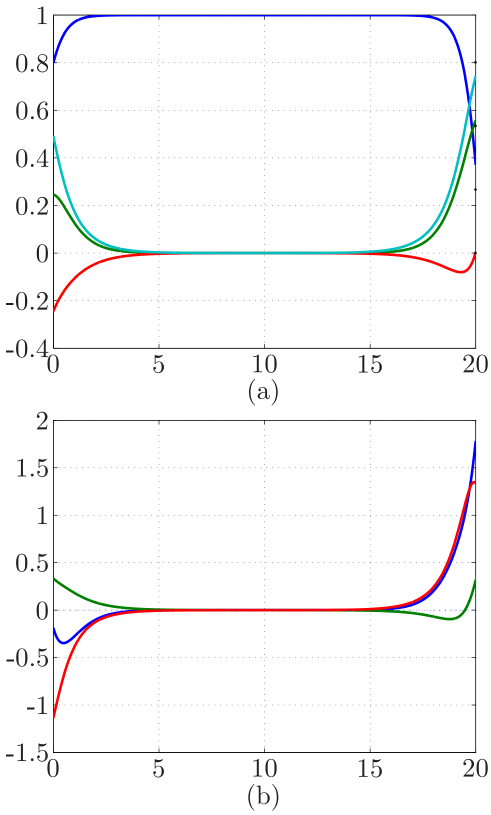

Fig. 1. Optimal state-control trajectory. Part (a) shows the optimal state trajectory versus time. The state is represented using unit quaternions (the scalar part is in blue). Part(b) shows the optimal control.

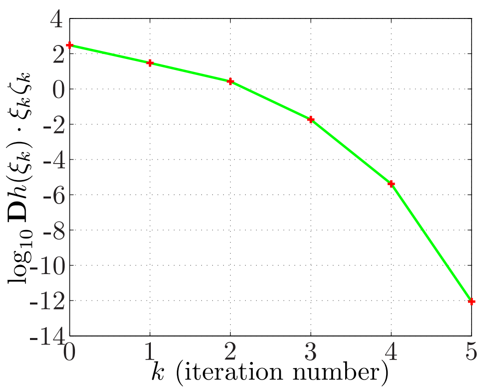

Fig. 2. Quadratic convergence rate. The plot shows $\log_{10} \mathbf{D}h(\xi_k) \cdot \xi_k\zeta_k \approx \log_{10}(h(\xi_{k+1}) - h(\xi_k))$ as a function of the number of iterations.

6. CONCLUSION

In this paper, we have shown how to compute the linear dynamics and the linear-plus-quadratic incremental and terminal costs that defines the search direction in the Lie group projection operator approach. A numerical example on the Lie group SO(3) has been presented, highlighting implementation details and demonstrating the expected second order convergence rate of the method.

REFERENCES

Abraham, R., Marsden, J.E., and Ratiu, T. (1988). *Manifolds, tensor analysis, and applications*. Springer-Verlag, New York.

Absil, P.A., Mahony, R., and Sepulchre, R. (2008). *Optimization algorithms on Matrix Manifolds*. Princeton University Press.

Anderson, B.D. and Moore, J.B. (1989). *Optimal Control: Linear Quadratic Methods*. Prentice Hall.

Boothby, W. (1986). *An introduction to differentiable manifolds and Riemannian geometry*. Pure and applied mathematics. Academic Press, Boston, 2nd edition.

Hauser, J. (2002). A projection operator approach to the optimization of trajectory functionals. In *15th IFAC World Congress*. Barcelona, Spain.

Hauser, J. and Meyer, D. (1998). The trajectory manifold of a nonlinear control system. In *37th IEEE Conference of Decision and Control (CDC)*, volume 1, 1034–1039.

Hauser, J. (2003). On the computation of optimal state transfers with application to the control of quantum spin systems. In *American Control Conference (ACC)*.

Hauser, J. and Saccon, A. (2006). A barrier function method for the optimization of trajectory functionals with constraints. In *45th IEEE Conference on Decision & Control*.

Jurdjevic, V. (1997). *Geometric control theory*. Cambridge University Press.

Lee, J.M. (1997). *Riemannian manifolds: an introduction to curvature*. Springer, New York.

Luemberger, D.G. (1969). *Optimization by Vector Space Methods*. John Wiley & Sons, New York.

Murphey, T. and Egerstedt, M. (2007). Choreography for marionettes: Imitation, planning, and control. In *IEEE Internation Conference on Intelligent and Robotic Systems*. San Diego, CA, USA.

Nocedal, J. and Wright, S.J. (1999.). *Numerical optimization*. Springer Verlag, New York.

Notarstefano, G. and Hauser, J. (2010). Modeling and dynamic exploration of a tilt-rotor VTOL aircraft. In *8th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*. Bologna, Italy.

Notarstefano, G., Hauser, J., and Frezza, R. (2007). Computing feasible trajectories for control-constrained systems: the pvtol aircraft. In *7th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*. Pretoria, South Africa.

Rossmann, W. (2002). *Lie groups. an introduction through linear groups*. Oxford University Press.

Saccon, A., Hauser, J., and Aguiar, A. (2010a). Optimal control on non-compact Lie groups: A projection operator approach. In *IEEE Conference on Decision and Control*. Atlanta, Georgia, USA. Online version available at `http://www.isr.ist.utl.pt/~asaccon`.

Saccon, A., Hauser, J., and Beghi, A. (2008). A virtual rider for motorcycles: An approach based on optimal control and maneuver regulation. In *International Symposium on Communications, Control and Signal Processing (ISCCSP)*. St. Julians, Malta.

Saccon, A., Hauser, J., and Beghi, A. (2010b). Trajectory exploration of a rigid motorcycle model. *Accepted for publication in IEEE Transactions on Control Systems Technology*.

Varadarajan, V. (1984). *Lie groups, Lie algebras, and their representations*. Springer-Verlag, New York.