Constrained Motion Planning for Multiple Vehicles on $SE(3)$

Alessandro Saccon, A. Pedro Aguiar, Andreas J. Häusler, John Hauser, António M. Pascoal

Abstract—This paper proposes a computational method to solve constrained cooperative motion planning problems for multiple vehicles undergoing translational and rotational motions. The problem is solved by means of the Lie group projection operator approach, a recently developed optimization strategy for solving continuous-time optimal control problems on Lie groups. State constraints (for collision avoidance) are handled by means of a set of barrier functions, turning the optimization approach into an interior point method. A sample computation is shown to demonstrate the effectiveness of the method.

I. INTRODUCTION

This papers addresses the problem of constrained cooperative motion planning for multiple vehicles undergoing translational and rotational motions that are naturally described in $SE(3)$. The practical motivation for this study stems from a number of envisioned mission scenarios for autonomous underwater vehicles (AUVs). A representative example is when a number of AUVs equipped with acoustic and vision sensor units must cooperate to obtain marine habitat maps in complex 3D terrain that includes flat-like surfaces, near vertical cliffs, and overhangs. In this situation, it is crucial that the vehicles carrying distinct but complementary resources move together at close range and change their spatial formation according to the orientation of the terrain being mapped. This entails the development of a motion planner to solve the problem of generating reference trajectories that the vehicles should track in order to move from initial to target poses (positions and attitudes) in a coordinated manner. Such a planner must, at the final stage, include a reasonably accurate model of each vehicle and a possibly rough description of known terrain obstacles.

The constraints that must be taken into consideration are twofold. The first is imposed by the fact that the vehicles must not collide with each other or with the environment. The second arises naturally from the fact that in order to do formation control the vehicles must use acoustic ranging devices and (at very close range) vision sensors to measure relative distances and attitudes. To avoid masking effects, it is crucial that nontrivial geometric constraints be imposed on the allowable motions.

Given the nature of the motion planning problem at hand, namely the presence of nontrivial geometric constraints and the need to take explicitly vehicle dynamics, actuator limitations, and (safety-related) state constraints into account, it is unlikely that standard path planning techniques can be applied to compute energy efficient maneuvers. We therefore exploit the use of a numerical optimal control approach that will allow for the introduction of nontrivial state and input constraints. At this stage, however, we restrict ourselves to kinematic models of the vehicles and take into account only inter-vehicle constraints.

The motion planner that we describe is based on the Lie group projection operator approach introduced in [1], [2]. This approach lifts the artificial constraints imposed by the choice of any local parametrization of the rotation matrices that describe the attitude of the vehicles. Furthermore, the exploitation of the geometry of the state space $SE(3)$ allows for the computation of the first and second derivatives of the vehicle dynamics, cost, and constraints in an intrinsic fashion, thus avoiding the computation of first and second derivatives of the local parametrization, which leads to computational advantages. State and inputs constraints are handled using the barrier function approach [3].

Solving complex motion planning problems for real time applications is a challenging task. We expect that the computational method we are developing will be coupled with other recent planning philosophies such motions primitives [4] or multi-level planners [5] (e.g., to obtain a system trajectory after the graph search phase in a roadmap is completed).

The literature on vehicle motion planning is vast and defies a simple summary. The plethora of methodologies available are quite diverse; we refer to, e.g., [6], [7], [8] and the references therein for a recent account.

The key contributions of this paper are threefold: i) it is shown that the Lie group projection operator approach can be used to solve constrained motion planning problem for multiple vehicles, demonstrating its use on the Lie group $SE(3)$; ii) it is the first time that the barrier function approach (cf. [3]) is applied to solve a constrained optimal control problem on a Lie group. In particular, we detail how to compute first and second order derivatives of $N(N - 1)/2$ constraint functions that act pairwise on a set of $N$ vehicles; finally, iii) the paper provides numerical evidence that the Lie group projection operator might be a viable auxiliary tool for investigating sub-Riemannian geometric problems on Lie groups.

Geometric methods are becoming standard tools in numer-
Theorem 3 (Main Theorem). Let \( (X, d_X) \) be a metric space and \( f : X \to \mathbb{R} \) be a continuous function. Given \( x^* \in X \) such that \( f(x^*) = \min_{x \in X} f(x) \), let \( A \) be the set of minimizers of \( f \) and \( \zeta(x) \) be the directional derivative of \( f \) at \( x \). Then, the following conditions are equivalent:

1. \( \zeta(x) = 0 \) for all \( x \in A \).
2. \( f \) is differentiable at \( x^* \) and \( \nabla f(x^*) = 0 \).

Proof. Condition 1 implies condition 2, because if \( \zeta(x) = 0 \) for all \( x \in A \), then \( f \) is constant on \( A \), hence differentiable with \( \nabla f(x^*) = 0 \). Conversely, if \( f \) is differentiable at \( x^* \) and \( \nabla f(x^*) = 0 \), then \( f \) is locally constant near \( x^* \), and hence \( \zeta(x) = 0 \) for all \( x \in A \).
Roughly speaking, the projection operator approach can be thought as a Newton method in infinite dimension. The approach is based on (and derives its name from) the projection operator \( P \) [25], which is an operator that maps a generic curve \( \xi(t) = (\alpha(t), \mu(t)) \in G \times \mathbb{R}^m \), \( t > 0 \), into a trajectory \( \eta(t) = (g(t), u(t)) \in G \times \mathbb{R}^m \), \( t > 0 \), of the system (4). The operator \( P \) is defined through the feedback system

\[
\dot{\xi}(t) = g(t)\lambda(g(t), u(t), t), \quad g(0) = \alpha(0), \\
\dot{u}(t) = \mu(t) + K(t)\left[ \log(g(t)^{-1} \alpha(t)) \right],
\]

where \( K(t) : g \to \mathbb{R}^m \) is a linear map, which can be thought as a standard linear feedback as soon as a basis is chosen for the Lie algebra \( \mathfrak{g} \). It is straightforward to verify that \( P \) is indeed a projection, i.e., it satisfies \( P^2 := P \circ P = P \).

Given a trajectory \( \xi(t) = (g(t), u(t)) \) of the control system (4), its (left-trivialized) linearization is defined as the time-varying linear system

\[
\dot{z}(t) = A(\xi(t), t)z(t) + B(\xi(t), t)v(t),
\]

with \((z(t), v(t)) \in \mathfrak{g} \times \mathbb{R}^m \), \( t \geq 0 \) and where

\[
A(\xi, t) := D_1\lambda(g, u, t) \circ T_gL_g - \text{ad}_{\lambda(g, u, t)}, \\
B(\xi, t) := D_2\lambda(g, u, t).
\]

The projection operator approach consists in applying the following iterative method

**Algorithm** (Projection operator Newton method)

**given** initial trajectory \( \xi_0 \in \mathcal{T} 

for \( i = 0, 1, 2, \ldots \) 

1. (search direction) \( \zeta_i = \arg \min_{\zeta_i \in T_{\xi_i}} D\hat{h}(\xi_i) \cdot \xi_i \zeta + \frac{1}{2} D^2\hat{h}(\xi_i) \cdot (\xi_i \zeta_i, \xi_i \zeta) \)

2. (step size) \( \gamma_i = \arg \min_{\gamma \in (0, 1]} \hat{h}(\xi_i \exp(\gamma \zeta_i)) \)

3. (update) \( \xi_{i+1} = P(\xi_i \exp(\gamma_i \zeta_i)) \)

end

In (12), \( \hat{h} \) is the cost functional appearing in (5) and \( \hat{h} \) is the functional obtained by composing \( h \) with the projection operator \( P \), i.e., \( \hat{h} := h \circ P \). At each iterate, the search direction minimization (12) is performed on the tangent space \( T_{\xi_i} \mathcal{T} \), that is, we search over the curves \( \zeta(\cdot) = (z(\cdot), v(\cdot)) \) that satisfies (9). Then, the step size subproblem (13) is considered. The classical approximate solution obtained using backtracking line search with Armijo condition [26, Chapter 3] can be used to compute the optimal step size \( \gamma_i \). Finally, the update step (14) projects each iterate on to the trajectory manifold and the process restarts as long as termination conditions have not been met.

The convergence to a local minimum that satisfies second order sufficient conditions for optimality can be readily checked.

**III. THE OPTIMAL MOTION PLANNING PROBLEM**

The motion planning problem that we consider can be briefly described as follows. Given a fixed time interval \([0, T], T > 0\), and a number of vehicles \( N \), compute the inputs that steer them from a given initial pose at time \( t = 0 \) to a desired final state at time \( t = T \) while minimizing an energy-related criterion and avoiding inter-vehicle collisions.

**A. Vehicle dynamics**

In what follows, the state of the \( i \)-th vehicle is represented by an element of the Lie group \( \text{SE}(3) \) as \( g[i] = (R[i], p[i]) \), \( i \in \{1, 2, \ldots, N\} \). At the kinematic level, the motion of each vehicle is described by the equations

\[
\dot{R}[i] = R[i]0; q[i]; p[i])^\wedge, \\
\dot{p}[i] = R[i]u[i]; 0; 0),
\]

where \( ; \) denotes row concatenation. In the above equations, the inputs for the \( i \)-th kinematic model are the pitch rate
control problem

B. The motion planning problem

The motion planning problem is obtained by solving the optimal control problem

\[
\min_{(\mathbf{g}, \mathbf{u})} \int_0^T \frac{1}{2} \sum_{i=1}^N \|\mathbf{u}^{[i]}(\tau)\|^2_R \, d\tau,
\]

where \(\mathbb{R}^{3 \times 3} \supseteq R = R^T > 0\) is a weighting matrix, subject to the dynamic constraint

\[
\dot{\mathbf{g}}(t) = f(g(t), \mathbf{u}(t)),
\]

with initial and final constraints

\[
\begin{align*}
\mathbf{R}^{[i]}(0) &= \mathbf{R}^{[i]}_0, \\
\mathbf{R}^{[i]}(T) &= \mathbf{R}^{[i]}, \\
\mathbf{p}^{[i]}(0) &= \mathbf{p}^{[i]}_0, \\
\mathbf{p}^{[i]}(T) &= \mathbf{p}^{[i]},
\end{align*}
\]

for \(i \in \{1, \ldots, N\}\). The collective dynamics (19) is obtained by stacking all the control systems (17) into a control system with state \(\mathbf{g} = [\mathbf{g}^{[1]}, \mathbf{g}^{[2]}, \ldots, \mathbf{g}^{[N]}] \in \text{SE}(3)^N\) and input \(\mathbf{u} = [\mathbf{u}^{[1]}, \mathbf{u}^{[2]}, \ldots, \mathbf{u}^{[N]}] \in \mathbb{R}^{3N}\). Note that in this paper the product Lie group SE(3)^N will play the role of the Lie group \(G\) discussed in Section II. We furthermore impose \(N_c = N(N-1)/2\) collision constraints

\[
\|\mathbf{p}^{[i]}(t) - \mathbf{p}^{[j]}(t)\|^2_R / D^2 - 1 \geq 0
\]

for \(i, j \in \{1, \ldots, N\}, i < j,\) and \(t \in [0, T]\). The collision constraint (22) requires that the distance between any two vehicles is never less than a fixed distance \(D\). To visualize this set of constraints, one may imagine that each vehicle is contained within a safety spherical hull of diameter \(D\) and that, during a maneuver, the spheres are to remain essentially intersection free, being allowed to just touch one another.

IV. SOLVING THE MOTION PLANNING PROBLEM

We handle the inequality constraints on the state and control through the barrier function approach described in [3]. The terminal constraint is indirectly (and approximately) addressed using a terminal cost penalty, cf. [24]. Therefore, given the optimal motion planning problem introduced in Section III-B, from the cost functional (18) and inequality constraints (22), we obtain the augmented cost functional

\[
\int_0^T l(\mathbf{g}, \mathbf{u}) + \sum_{k=1}^{N_c} \varepsilon \beta_\delta(c_k(\mathbf{g}, \mathbf{u})) \, d\tau + m(\mathbf{g}(T))
\]

where

\[
l(\mathbf{g}, \mathbf{u}) := \frac{1}{2} \sum_{i=1}^N \|\mathbf{u}^{[i]}\|^2_R,
\]

\[
c_k(\mathbf{g}, \mathbf{u}) := \|\mathbf{p}^{[i]} - \mathbf{p}^{[j]}\|^2_R / D^2 - 1
\]

with \(i, j \in \{1, \ldots, N\}, i < j,\) and \(k = k(i, j) \in \{1, \ldots, N_c\}\). The terminal cost is defined as

\[
m_\rho(\mathbf{g}) := \frac{1}{2} \sum_{i=1}^N \rho_R \|\mathbf{R}^{[i]} - (\mathbf{R}^{[i]}_f)^T \mathbf{R}^{[i]}\|^2 + \rho_p \|\mathbf{p}^{[i]} - \mathbf{p}^{[i]}_f\|^2,
\]

with \(\rho = (\rho_R, \rho_p)\). We now explain in details the cost (23).

a) Constraint indexing: Each collision constraint is indexed through the index \(k\). This index is computed from the indexes \(i, j\) of the two vehicles to which the constraint applies. For a generic value of \(N\), one gets \(k(i, j) = j + i(N - 1) - i(i + 1)/2, i < j, i, j \in \{1, \ldots, N\}\). For \(N = 3\), e.g., \(k(1, 2) = 1, k(1, 3) = 2,\) and \(k(2, 3) = 3\). For reasons that will be clarified later, we define \(k(j, i) = k(i, j)\), \(i \leq j,\) and \(k(i, i) = 0, i \in \{1, \ldots, N\}\), making \([k(i, j)]\) into a symmetric matrix.

b) Modified barrier function: In this work, we use an approximate barrier function based on the one proposed in [3] that is well adapted to constraints of the form (22). Recall that [3], for \(0 < \delta \leq 1\),

\[
\beta_\delta(z) = \begin{cases} 
-\log z & z > \delta \\
\frac{k-1}{k} \left( \frac{z-k\delta}{(k-1)\delta} \right)^k - 1 \log \delta & z \leq \delta
\end{cases}
\]

provides an approximate log barrier function that can be evaluated outside of the strictly feasible region \(z > 0\). (Here \(k > 1\) is an even integer, usually taken to be \(k = 2\)). In many applications, including those using an input constraint like \(1 - \|\mathbf{u}\|^2_R \geq 0\), the constraint function is naturally bounded above by 1. On the other hand, an exclusion constraint function such as \(c_k(\mathbf{g}, \mathbf{u})\) in (25) will often have values substantially greater than 1, resulting in an (artificially) lower than expected cost in (23) due to the possibly large magnitude of \(-\log c_k(\mathbf{g}, \mathbf{u})\). To capture the fact that we only want to penalize attempted collisions, we make use of the “hockey stick” function

\[
\sigma(z) = \begin{cases} 
\tanh(z) & z \geq 0 \\
z & \text{otherwise}
\end{cases}
\]

to saturate each exclusion constraint function \(c_k(\mathbf{g}, \mathbf{u})\) expressing, in exponential fashion (look at \(-\log \tanh z, z > 0\)), the manner in which our concern for collision fades as the separation distance increases. The approximate barrier function in (23) will thus be taken to be \(\tilde{\beta}_\delta(z) = \tilde{\beta}_\delta(\sigma(z))\).

c) Terminal cost \(m\): The terminal cost (26) weighs the deviation of the final state \(\mathbf{g}^{[i]}(T) = (\mathbf{R}^{[i]}(T), \mathbf{p}^{[i]}(T))\) from the desired final state \(\mathbf{g}^{[i]}_f = (\mathbf{R}^{[i]}_f, \mathbf{p}^{[i]}_f)\). Note that \(\|I - \mathbf{R}\|^2_F\) refers to the squared Frobenius norm \(\text{tr}(I - \mathbf{R})^T(I - \mathbf{R})\) that, together with its first and second (covariant) derivatives, has been described in [2], [23]. The parameters \(\rho_R\) and \(\rho_p\) are chosen large enough to ensure that the final state of each vehicle approaches the desired terminal condition with a prescribed tolerance (penalty approach).
V. OPTIMAL DESCENT DIRECTION

The descent direction for the projection operator Newton method is computed by solving the subproblem (12). As shown in [2], given a trajectory $\xi(t) = (g(t), u(t))$ of (19), $t \in [0, T]$, the subproblem (12) is equivalent to solving a LQ optimal control problem of the form

$$\min_{(z,v)(\cdot)} \int_0^T a(\tau)^T z(\tau) + b(\tau)^T v(\tau) + \frac{1}{2} [z(\tau)^T W(\tau) v(\tau)]^T z(\tau) d\tau + a_1^T z(T) + \frac{1}{2} z(T)^T P_1 z(T),$$

subject to the dynamic constraint

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

$$z(0) = 0.$$  

The general expressions for the left-trivialized linearization $A$ and $B$ appearing in (29) have been given in (10) and (11), respectively. The general expressions of the vectors $a$, $b$, $a_1$ and matrices $W$, $P_1$ in terms of the (left-trivialized) dynamics $\lambda$ and the integral cost $l$, and terminal cost $m$ are given in [2]. Due to space limitations, we will not provide the expressions of these quantities. An extended, eight pages version of this paper containing the explicit expressions for the special case of the cost (23) and dynamics (19) is readily available upon direct request to the authors.

VI. SAMPLE CALCULATION

We consider in this section a planning example for three vehicles. The example shows the effectiveness of the planner in finding trajectories for the three vehicles that match the prescribed initial and final conditions and avoid inter-vehicle collisions. The scenario is depicted in Figure 1. The planned trajectory has a total duration of 10 seconds and 6 snapshots of the animation are shown.

The desired initial and final vehicle positions lie at the corners of an hexagon and each vehicle must reach the corner opposite to the one it starts from. The hexagon diameter has length 20 m. Starting with a zero initial roll angle, each vehicle must reach the final position with a roll angle of 60 degrees.

Because no direct control of the roll velocity is available, the planner generates the desired roll motion mainly by cycling the yaw and pitch rate commands, exploiting the nonlinear controllability of the model. This phenomenon is evident by looking at the optimal inputs given in Figure 4. Its effect on the roll angle of each vehicle can be appreciated in Figure 1, where one can also see the cycling motion in the yaw angle. Two vehicles (blue and yellow) perform this wiggling at the beginning of the optimal maneuver, while the third (green) waits until it “slips through” the other two vehicles, thus avoiding a possible collision. The speed of each vehicle is kept approximately equal to 2 m/s as each vehicle has to cover a distance of approximately 20 m in 10 s.

The collision between the two slower vehicles does not
occur as one passes on top of the other as shown in the detailed snapshots of Figure 2. We can monitor at each iteration the value of the constraint functions by plotting the value of \( c_k(g(t), u(t)), t \in [0, T], k \in \{1, 2, 3\} \) as shown in the upper part of Figure 3. Due to the symmetry of the problem, two constraints take the same value along the optimal trajectory (the distance of the green-yellow and the green-blue vehicle pairs is the same). The touching of the spherical hulls of the yellow and blue vehicles is reflected in the fact that the corresponding constraint function goes to zero. The graph at the bottom of Figure 3 makes evident the role of the barrier function \( \beta_3 \) in penalizing the possible collision between the hulls. Note how \( \beta_3(c_k(g, u)) \) gets larger as the corresponding constraint function \( c_k(g, u) \) approaches zero.

VII. CONCLUSIONS

We have discussed the application of the Lie group projection operator approach to a constrained optimization problem involving a set of dynamics systems on \( SE(3) \). We have detailed how to construct the quadratic approximation of the original problem and, in particular, how to compute the first and second derivative of the constraint functions that arise in the problem of cooperative planning for a group of \( N \) vehicles. A sample computation was discussed, showing that with this approach it is possible to compute a optimal trajectory that solves the constrained optimization problem, improving the confidence that the strategy can be a viable way to deal with more complex vehicle dynamics and geometric constraints. Convergence rate, existence and uniqueness, executions time, updating rules for the constraint parameters \( \varepsilon \) and \( \delta \), and further implementation issues will be addressed in future work.

For a single vehicle (and no collision constraints), the particular optimal control problem studied in this paper corresponds to the computation of the geodesics for a sub-Riemannian manifold defined on the Lie groups \( SE(3) \) [27], [28], [29], [30]. The Lie group projection operator approach might be therefore used as an auxiliary tool in the study of this interesting branch of differential geometry. Further research is required however to understand if, e.g., we can obtain strictly abnormal minimizers [31].

REFERENCES