Systems & Control Letters 108 (2017) 56–63

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification☆

Changliu Liu *, Masayoshi Tomizuka

Department of Mechanical Engineering, University of California at Berkeley, Berkeley CA 94720, United States

ARTICLE INFO

Article history:
Received 7 December 2016
Received in revised form 21 June 2017
Accepted 18 August 2017

Keywords:
Optimal control
Nonlinear optimization
Non convex optimization
Motion planning

ABSTRACT

Real time trajectory optimization is critical for robotic systems. Due to nonlinear system dynamics and obstacles in the environment, the trajectory optimization problems are highly nonlinear and non convex, hence hard to be computed online. Liu, Lin and Tomizuka proposed the convex feasible set algorithm (CFS) to handle the non convex optimization in real time by convexification. However, one limitation of CFS is that it will not converge to local optima when there are nonlinear equality constraints. In this paper, the slack convex feasible set algorithm (SCFS) is proposed to handle the nonlinear equality constraints, e.g. nonlinear system dynamics, by introducing slack variables to relax the constraints. The geometric interpretation of the method is discussed. The feasibility and convergence of the SCFS algorithm is proved. It is demonstrated that SCFS performs better than existing non convex optimization methods such as interior-point, active set and sequential quadratic programming, as it requires less computation time and converges faster.

1. Introduction

Although great progresses have been made in motion planning for robotic systems [1], challenges remain in real time planning in dynamic uncertain environment. The applications include but are not limited to real time navigation [2], autonomous driving [3], robot arm manipulation and human robot cooperation [4]. To achieve real time safety and efficiency, the robot motion should be re-planned from time to time when new information is obtained during operation, which requires the motion planning algorithms to run fast enough online.

This paper focuses on optimization-based motion planning, where an ideal low level tracking controller is assumed. The method fits into the framework of model-predictive control (MPC) [5], where an optimal trajectory is obtained by solving a constrained optimization or optimal control problem in receding horizons. The optimization problem may be highly nonlinear due to the dynamic constraints, and highly non convex due to the constraints for obstacle avoidance, which makes it hard to be solved online.

Convexification [6] is a popular way to deal with non convexity by transforming the non convex problem into a convex one. One popular convexification method is the sequential quadratic programming (SQP), which approximates the non convex problem as a sequence of quadratic programming (QP) problems and solves them iteratively. References for SQP can be founded in [7] and [8]. The method has been successfully applied to offline robot motion planning as discussed in [9] and [10]. However, as SQP is designed for general purpose, the unique geometric structure of the motion planning problems is neglected. As a consequence, it is hard to use for real time applications.

In practice, the cost function for motion planning is usually designed to be convex [11,12], while the non convexity mainly comes from the physical constraints, e.g. obstacles. Regarding this observation, the convex feasible set algorithm (CFS) [13] was proposed to handle motion planning problems with convex objective functions and non convex inequality constraints. The idea of the CFS algorithm is to transform the origin problem into a sequence of convex subproblems by obtaining convex feasible sets within the non convex inequality constraints, and then iteratively solve the convex subproblems until solutions converge. The difference between CFS and SQP lies in the methods in obtaining the convex subproblems, where the geometric structure of the motion planning problem is fully considered in CFS.

However, one limitation of CFS is that it may not converge to local optima under nonlinear equality constraints (such as nonlinear dynamic constraints) as the convex feasible set for a nonlinear equality constraint may reduce to a singleton point.

In this paper, the slack convex feasible set algorithm (SCFS) is introduced to handle optimization problems with convex cost

☆ The work is supported in part by Berkeley fellowship and in part by FANUC Corporation, Japan.

* Corresponding author.
E-mail addresses: changliuliu@berkeley.edu (C. Liu), tomizuka@berkeley.edu (M. Tomizuka).

http://dx.doi.org/10.1016/j.sysconle.2017.08.004
0167-6911/

functions and non convex equality and inequality constraints. The idea is to relax the nonlinear equality constraints to several nonlinear inequality constraints using slack variables and then solve the relaxed problem using CFS. The feasibility, convergence and optimality of the algorithm will be proved in the paper. The performance of SCFS will be compared to that of SQP as well as other existing non convex optimization algorithms.

The remainder of the paper will be organized as follows: in Section 2, a benchmark motion planning problem is proposed; Section 3 reviews the convex feasible set algorithm; Section 4 introduces the slack convex feasible set algorithm; Section 5 illustrates the performance of SCFS; Section 6 concludes the paper.

2. Problem formulation

2.1. The notations

Denote the state of the robot as $x \in X \subset \mathbb{R}^n$ where X represents n dimensional state space. Denote the control input of the robot as $u \in U \subset \mathbb{R}^m$ where U represents m dimensional control space.[1] Suppose the robot needs to travel from x^{start} to x^{goal}. The robot trajectory is denoted as $\mathbf{x} = [x_0^T, x_1^T, \ldots, x_h^T]^T \in X^{h+1}$ where x_q is the robot state at time step q and h is the planning horizon. Without loss of generality, the sampling time t_s is assumed to be 1. Similarly, the input trajectory is denoted as $\mathbf{u} = [u_0^T, u_1^T, \ldots, u_{h-1}^T]^T \in U^h$ where u_q is the robot input at time step q. Let u_q^j denote the jth entry in u_q for $j = 1, \ldots, m$.

2.2. The benchmark problem and the assumptions

Problem 1 (*The Benchmark Problem*)**.** Consider the following optimization problem

$$\min_{\mathbf{x},\mathbf{u}} \quad J(\mathbf{x}, \mathbf{u}) \tag{1}$$

$$s.t. \quad \mathbf{x} \in \Gamma, \mathbf{u} \in \Omega, \mathcal{G}(\mathbf{x}, \mathbf{u}) = 0, \tag{2}$$

where $J : X^{h+1} \times U^h \to \mathbb{R}$ is the cost function; Γ is the constraint on the augmented state space X^{h+1}; Ω is the constraint on the augmented control space U^h; and $\mathcal{G} : X^{h+1} \times U^h \to \mathbb{R}^{mh}$ represents the dynamic relationship between states and inputs. Assumptions 2 to 6 are required.

Assumption 2 (*Cost Function*)**.** The cost function $J(\mathbf{x}, \mathbf{u}) = J_1(\mathbf{x}) + J_2(\mathbf{u})$ is smooth and bounded below by 0. J_1 is strictly convex. J_2 is strictly convex and symmetric, and it achieves minimum at $\mathbf{u} = 0$.

Assumption 3 (*State Constraint*)**.** The constraint Γ is a collection of linear equality constraints, linear inequality constraints and N nonlinear inequality constraints, i.e. $\Gamma = \cap_i \Gamma_i$ where

$$\Gamma_i = \begin{cases} \{\mathbf{x} : \phi_i(\mathbf{x}) \geq 0\} & i = 1, \ldots, N \\ \{\mathbf{x} : A_{eq}\mathbf{x} = b_{eq}\} & i = N+1 \\ \{\mathbf{x} : A\mathbf{x} \leq b\} & i = N+2, \end{cases} \tag{3}$$

$A_{eq} \in \mathbb{R}^{k_{eq} \times n(h+1)}$, $b_{eq} \in \mathbb{R}^{k_{eq}}$, $A \in \mathbb{R}^{k \times n(h+1)}$, and $b \in \mathbb{R}^k$. $k_{eq} < n(h+1)$ and k are the dimensions of the constraints. $rank(A_{eq}) = k_{eq}$. Function $\phi_i : \mathbb{R}^{n(h+1)} \to \mathbb{R}$ is continuous, piecewise smooth and semi-convex, e.g. there exists a positive semi-definite matrix $H_i^* \in \mathbb{R}^{n(h+1) \times n(h+1)}$ such that for any $\mathbf{x}$, $v \in \mathbb{R}^{n(h+1)}$, $\phi_i(\mathbf{x}+v) - 2\phi_i(\mathbf{x}) + \phi_i(\mathbf{x}-v) \geq -v^T H_i^* v$. Moreover, the interior of the inequality constraints is nontrivial, i.e. $\cap_i \{\mathbf{x} : \phi_i(\mathbf{x}) > 0\} \neq \emptyset$.[2]

The linear equality constraints are for boundary conditions at the start point and the goal point. The linear inequality constraints are for state limits. The nonlinear equality constraints are for collision avoidance where ϕ_i can be identified as a signed distance function to an obstacle. The semi-convexity assumption on ϕ_i is satisfied if there is no concave corner in the obstacle.

Assumption 4 (*Control Constraint*)**.** The constraint Ω is a box constraint such that $-\bar{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}$ for some constant vector $\bar{\mathbf{u}} := [\bar{u}_0^1, \ldots, \bar{u}_{h-1}^m]^T > 0$ where $\bar{u}_q^j \in \mathbb{R}^+$ is the bound for u_q^j for all q and j.

Assumption 5 (*Dynamic Constraint*)**.** The dynamic equation $\mathcal{G}(\mathbf{x}, \mathbf{u}) = 0$ is affine in $\mathbf{u}$, i.e. there exist smooth functions $F : X^{h+1} \to \mathbb{R}^{mh}$ and $H : X^{h+1} \to \mathbb{R}^{mh \times mh}$ such that

$$\mathcal{G}(\mathbf{x}, \mathbf{u}) = F(\mathbf{x}) + H(\mathbf{x})\mathbf{u} = 0. \tag{4}$$

H is assumed to be diagonal, non-singular and positive definite. Eq. (4) is equivalent to

$$f_q^j(\mathbf{x}) + h_q^j(\mathbf{x})u_q^j = 0, \forall q = 0, \ldots, h-1, j = 1, \ldots, m, \tag{5}$$

where $f_q^j : X^{h+1} \to \mathbb{R}$ and $h_q^j : X^{h+1} \to \mathbb{R}^+$ are entries in F and H, which are smooth with bounded derivatives and Hessians.

Eqs. (4) and (5) cover a wide range of typical nonlinear dynamic systems. For robot arms, let x be the robot joint position and u be the torque input, then the relationship between x and u is in the form of (5), i.e.

$$M(x_q)(x_{q+1} - 2x_q + x_{q-1}) + N(x_q, x_q - x_{q-1}) = u_q, \tag{6}$$

where $M(\cdot)$ is the generalized inertia matrix and $N(\cdot, \cdot)$ is the Coriolis and centrifugal forces. Finite differences are used to compute joint velocity and joint acceleration.

Since the input u may not be physical, the dynamic equation can also be understood in a broader sense, which is just an equation that captures the relationship between the state x and the parameter u that needs to be optimized. For example, in trajectory planning for automated vehicles, the yaw rate of the trajectory needs to be minimized. Let x be the position of the rear axle of the vehicle and u be the yaw rate, then the relationship between the yaw rate and the vehicle state assuming no tire slip ((7) below) satisfies (5),

$$(x_q - x_{q-1}) \times (x_{q+1} - x_q) = \|x_q - x_{q-1}\|^2 u_q, \tag{7}$$

where $\times$ denotes the cross product.

Moreover, the state x can also be non physical. For example, in speed profile planning for a given path [14], the state x is chosen as the time stamps along the path, while the path is sampled evenly with distance d. The input u is chosen to be the speed. The relationship between x and u ((8) below) also satisfies (5),

$$\frac{d}{x_q - x_{q-1}} = u_q. \tag{8}$$

By Assumptions 3 to 5, the constraints in (2) form a K dimensional manifold $\mathcal{M}$ where $K = n(h+1) - k_{eq}$.[3] In order for the optimization to be nontrivial, the manifold $\mathcal{M}$ should have non empty interior, which leads us to the following assumption.

Assumption 6 (*Connected Nontrivial Domain*)**.** The domain that satisfies (2) is connected. There exist $\mathbf{x}^*$ and $\mathbf{u}^*$ that satisfy all the constraints in (2) such that $A\mathbf{x}^* < b$, $\phi_i(\mathbf{x}^*) > 0$ for all i and $-\bar{\mathbf{u}} < \mathbf{u}^* < \bar{\mathbf{u}}$.

[1] Control input u is not necessarily a physical input (such as the throttle angle for a vehicle). It can be any parameter that needs to be considered in the trajectory optimization (such as the yaw rate of a vehicle).

[2] This is to exclude the case that some combination of nonlinear inequality constraints indeed forms a nonlinear equality constraint, such as $\Gamma = \{\mathbf{x} : \phi_i(\mathbf{x}) \geq 0, -\phi_i(\mathbf{x}) \geq 0\}$.

[3] The dimension of the decision variables $\mathbf{x}$ and $\mathbf{u}$ is $n(h+1) + mh$. As there are $k_{eq} + mh$ independent equality constraints, the dimension of the manifold is reduced to $n(h+1) - k_{eq}$.

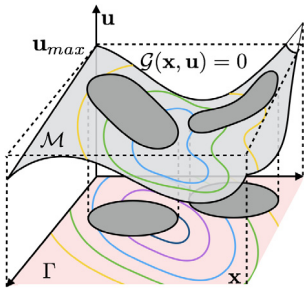


Fig. 1. Illustration of the geometry of Problem 1.

The geometry of Problem 1 is illustrated in Fig. 1 where the horizontal plane represents the augmented state space X^{h+1} and the vertical axis represents the augmented control space U^h. The dark holes represent the infeasible set $X^{h+1} \setminus \Gamma$. The manifold $\mathcal{M}$ is represented by the shaded curvy surface. The cost J on the manifold $\mathcal{M}$ is shown as the contours. Although the cost J is convex in $\mathbf{x}$ and $\mathbf{u}$, the cost $J|_{\mathcal{M}}$ is non convex as $\mathcal{M}$ is nonlinear and non convex.

3. The convex feasible set algorithm

The convex feasible set algorithm was proposed in [13] to solve the following problem.

Problem 7 (*The Benchmark Problem for the CFS Algorithm*)**.**

$$\min_{\mathbf{x}\in\Gamma} J(\mathbf{x}), \tag{9}$$

where J satisfies Assumption 2 and Γ satisfies Assumption 3. Moreover, the interior Γ^o of Γ is non empty.

Problem 7 is a special case of Problem 1. The geometry of Problem 7 is illustrated in Fig. 2, which is identical to the geometry on the plane $\mathbf{u} = 0$ in Fig. 1. In this section, the CFS algorithm will be briefly reviewed, followed by the discussion of its limitations in handling Problem 1. The application of the CFS algorithm is discussed in [15].

3.1. The CFS algorithm

A convex feasible set $\mathcal{F}$ for the set Γ is a convex set such that $\mathcal{F} \subset \Gamma$. It is easier to find a minimum of J in the convex feasible set $\mathcal{F}$ than in the non convex Γ. Moreover, as Γ can be covered by several (may be infinitely many) convex feasible sets, we can efficiently search the non convex space Γ for solutions by solving a sequence of convex optimizations constrained in the convex feasible sets.

The idea is implemented iteratively. At iteration k, given a reference point $\mathbf{x}^{(k)}$, a convex feasible set $\mathcal{F}(\mathbf{x}^{(k)}) \subset \Gamma$ is computed around $\mathbf{x}^{(k)}$. Then a new reference point $\mathbf{x}^{(k+1)}$ will be obtained by solving the following convex optimization problem

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}\in\mathcal{F}(\mathbf{x}^{(k)})} J(\mathbf{x}). \tag{10}$$

The iteration will terminate if either the solution converges or the descent in cost is small.

For a reference point $\mathbf{x}^r$, the desired convex feasible set $\mathcal{F}(\mathbf{x}^r)$ should be computed as $\mathcal{F}(\mathbf{x}^r) := \cap_i \mathcal{F}_i(\mathbf{x}^r)$ where $\mathcal{F}_i(\mathbf{x}^r) \subset \Gamma_i$ is the desired convex feasible set for Γ_i. $\mathcal{F}_i(\mathbf{x}^r)$ in different cases are illustrated in Fig. 3 where the dark shaded areas represent the infeasible sets $X^{h+1} \setminus \Gamma_i$. Mathematical definition of $\mathcal{F}_i(\mathbf{x}^r)$ is stated below.

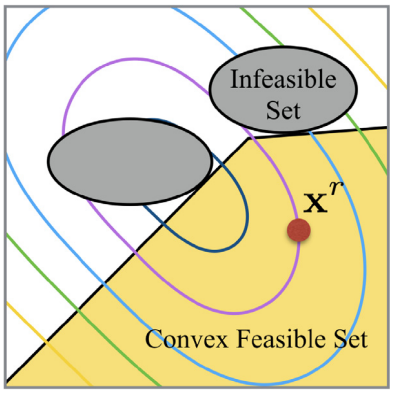
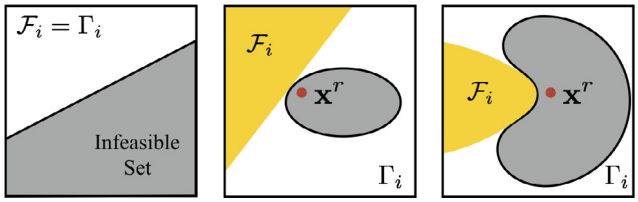


Fig. 2. Illustration of the geometry of Problem 7.



(a) Case 1. (b) Case 2. (c) Case 3.

Fig. 3. Construction of the convex feasible set $\mathcal{F}_i$ from Γ_i.

Case 1: Γ_i is convex

Define $\mathcal{F}_i := \Gamma_i$. Hence all linear constraints remain the same in the convex feasible set.

Case 2: $X^{h+1} \setminus \Gamma_i$ is convex

In this case, ϕ_i is convex since the signed distance function to a convex obstacle is convex. Then $\phi_i(\mathbf{x}) \geq \phi_i(\mathbf{x}^r) + \nabla\phi_i(\mathbf{x}^r)(\mathbf{x} - \mathbf{x}^r)$, where ∇ is the gradient operator.[4] The desired convex feasible set $\mathcal{F}_i$ for Γ_i at $\mathbf{x}^r$ is defined as

$$\mathcal{F}_i(\mathbf{x}^r) := \{\mathbf{x} : \phi_i(\mathbf{x}^r) + \nabla\phi_i(\mathbf{x}^r)(\mathbf{x} - \mathbf{x}^r) \geq 0\}. \tag{11}$$

Case 3: neither Γ_i nor $X^{h+1} \setminus \Gamma_i$ is convex

In this case, ϕ_i is neither convex nor concave. By Assumption 3, we define a new convex function $\tilde{\phi}_i$ as $\tilde{\phi}_i(\mathbf{x}) := \phi_i(\mathbf{x}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^r)^T H_i^*(\mathbf{x} - \mathbf{x}^r)$. Then $\tilde{\phi}_i(\mathbf{x}) \geq \tilde{\phi}_i(\mathbf{x}^r) + \nabla\tilde{\phi}_i(\mathbf{x}^r)(\mathbf{x} - \mathbf{x}^r) = \phi_i(\mathbf{x}^r) + \nabla\phi_i(\mathbf{x}^r)(\mathbf{x} - \mathbf{x}^r)$ where $\nabla\phi_i$ is identified with the sub-gradient of $\tilde{\phi}_i$ at points that are not differentiable. The desired convex feasible set $\mathcal{F}_i$ for Γ_i at $\mathbf{x}^r$ is then defined to be

$$\mathcal{F}_i(\mathbf{x}^r) := \{\mathbf{x} : \phi_i(\mathbf{x}^r) + \nabla\phi_i(\mathbf{x}^r)(\mathbf{x} - \mathbf{x}^r) \geq \frac{1}{2}(\mathbf{x} - \mathbf{x}^r)^T H_i^*(\mathbf{x} - \mathbf{x}^r)\}. \tag{12}$$

The convex feasible set for Problem 7 chosen according to the above rules is illustrated in Fig. 2 when ϕ_i's are signed distance functions to the obstacles. Note that $\mathbf{x}^r$ may be infeasible only at the first iteration.

3.2. Theoretical results

The feasibility and convergence of the CFS algorithm is summarized in the following theorem, which is proved in [13].

[4] At the point where ϕ_i is not differentiable, $\nabla\phi_i$ is a sub-gradient which is chosen such that the steepest descent of J in the set Γ is always included in the convex feasible set $\mathcal{F}$.

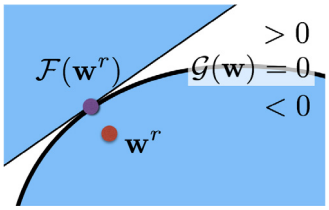


Fig. 4. The convex feasible set for a nonlinear equality constraint.

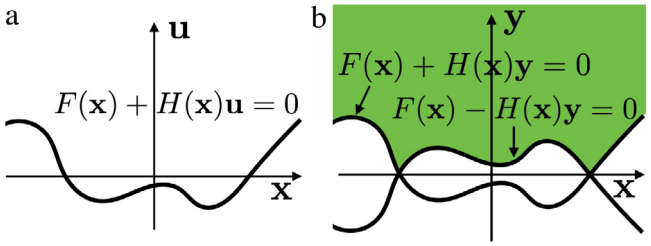


Fig. 5. Illustration of (a) the nonlinear equality constraint and (b) the relaxed nonlinear inequality constraint for the slack variable.

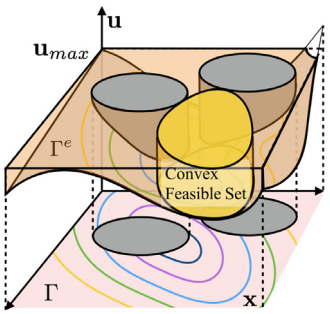


Fig. 6. The convex feasible set for Problem 11.

Theorem 8 (*Convergence of CFS*). *For Problem 7, the sequence* $\{\mathbf{x}^{(k)}\}$ *generated by iteration (10) with the convex feasible sets* $\mathcal{F}(\mathbf{x}^{(k)})$ *chosen according to the three cases will converge to a local optimum* $\mathbf{x}^o \in \Gamma$ *of Problem 7 for any nearly feasible initial value* $\mathbf{x}^{(0)}$.

Definition 9. We call $\mathbf{x}^o$ a local optimum if the first order optimality condition, e.g. the Karush–Kuhn–Tucker (KKT) condition, is satisfied (see [13], Fig. 5).

Another definition for local optima concerns with optimality within a neighborhood. We call those points *absolute local optima*. It is worth noting that KKT points may not be absolute local optima. Nonetheless, for motion planning problem, linear independent constraint qualification (LICQ) is usually satisfied at absolute local optima, which implies that absolute local optima should be KKT points.

Definition 10. We call $\mathbf{x}^r$ nearly feasible if $\mathcal{F}(\mathbf{x}^r) \neq \emptyset$.

It is also shown in [13] that one sufficient condition for $\mathcal{F}(\mathbf{x}^r)$ to be nonempty is: for each entry of $\mathbf{x}^r$, at most one nonlinear constraint can be violated.

3.3. Limitations

CFS is good at handling non convex inequality constraints. However, it cannot handle the nonlinear equality constraint (4) in Problem 1. Indeed, the convex feasible set for $\mathcal{M}$ may degenerate to a singleton point. Let $\mathbf{w} = [\mathbf{x}^T, \mathbf{u}^T]^T$. Then Problem 1 is transformed to

$$\min_{\mathbf{w}\in\mathcal{M}} J(\mathbf{w}), \tag{13}$$

where $\mathcal{M} = (\Gamma \oplus \Omega) \cap \{\mathcal{G}(\mathbf{w}) \geq 0\} \cap \{\mathcal{G}(\mathbf{w}) \leq 0\}$. We break the nonlinear constraint $\mathcal{G}(\mathbf{w})$ into two inequality constraints in order to put (13) in the form of Problem 7 and apply the CFS algorithm. Suppose $\mathcal{G}$ is convex in $\mathbf{w}$. Then the convex feasible set for the nonlinear equality constraints $\{\mathcal{G}(\mathbf{w}) \geq 0, -\mathcal{G}(\mathbf{w}) \geq 0\}$ is $\{\mathcal{G}(\mathbf{w}^r) + \nabla\mathcal{G}(\mathbf{w}^r)(\mathbf{w} - \mathbf{w}^r) \geq 0, \mathcal{G}(\mathbf{w}) \leq 0\}$, which degenerates to one point as shown in Fig. 4. The CFS algorithm will get stuck in that point, which is feasible but not necessarily optimal. Theorem 8 does not apply, since $\mathcal{M}$ violates Assumption 3 if we regard $\mathbf{w}$ as the state variables. Hence local optima may not be achieved by applying the CFS algorithm on Problem 1.

4. The slack convex feasible set algorithm

The slack convex feasible set algorithm (SCFS) will be introduced in this section to solve Problem 1. The key idea is to introduce slack variables and transform the nonlinear equality constraint (4) to a set of non degenerating nonlinear inequality constraints such that the CFS algorithm can be applied.

4.1. The relaxed problem

Let $\mathbf{y}$ be the slack variable for $\mathbf{u}$ such that $-\mathbf{y} \leq \mathbf{u} \leq \mathbf{y}$. Symmetry of Problem 1 in $\mathbf{u}$ is exploited in the relaxation. By Assumption 2, $J_2(\mathbf{u}) \leq J_2(\mathbf{y})$. By Assumption 4, $\mathbf{y} \leq \bar{\mathbf{u}}$ implies $-\bar{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}$. Using the slack variable $\mathbf{y}$, constraint (4) can be relaxed to two inequality constraints with respect to $\mathbf{y}$, e.g. $F(\mathbf{x}) - H(\mathbf{x})\mathbf{y} \leq F(\mathbf{x}) + H(\mathbf{x})\mathbf{u} = 0 \leq F(\mathbf{x}) + H(\mathbf{x})\mathbf{y}$. Then we have the following relaxed problem.

Problem 11 (*The Relaxed Problem*).

$$\min_{\mathbf{x},\mathbf{y}} \quad J(\mathbf{x}, \mathbf{y}) \tag{14}$$

$$s.t. \quad \mathbf{x} \in \Gamma, \mathbf{y} \leq \bar{\mathbf{u}} \tag{15}$$

$$F(\mathbf{x}) + H(\mathbf{x})\mathbf{y} \geq 0, F(\mathbf{x}) - H(\mathbf{x})\mathbf{y} \leq 0. \tag{16}$$

Consider the extended state $\mathbf{z} = [\mathbf{x}^T, \mathbf{y}^T]^T$. The relaxed problem can be transformed to

$$\min_{\mathbf{z}\in\Gamma^e} J(\mathbf{z}), \tag{17}$$

where $\Gamma^e := (\Gamma \oplus \{\mathbf{y} \leq \bar{\mathbf{u}}\}) \cap \{F(\mathbf{x}) + H(\mathbf{x})\mathbf{y} \geq 0\} \cap \{-F(\mathbf{x}) + H(\mathbf{x})\mathbf{y} \geq 0\}$. The difference between Problem 11 and Problem 1 is illustrated in Fig. 5, where the curve in Fig. 5(a) represents the nonlinear equality constraint (4) in Problem 1 and the shaded area in Fig. 5(b) represents the nonlinear inequality constraints (16) in Problem 11. Due to the introduction of the slack variable $\mathbf{y}$, the nonlinear equality constraint is successfully removed. Γ^e is illustrated in Fig. 6, whose dimension is $n(h+1) + mh - k_{eq}$. $\mathcal{M}$ belongs to the boundary of Γ^e.

The intuition behind the relaxation is that: as J_2 achieves minimum at $\mathbf{u} = 0$, the optimization algorithm will "pull" the optimal solution down to $\mathcal{M}$ which is on the "bottom" of Γ^e, so that we may still get the same optima as in Problem 1. This property will be proved in Theorem 13, Lemma 14 and Proposition 15. We first verify that the relaxed problem satisfies the assumptions in Problem 7 so that CFS can be applied.

Lemma 12. *Γ^e satisfies Assumption 3 if we take $\mathbf{z}$ as the state variables.*

Proof. It is easy to verify that Γ^e is a collection of linear equality constraints, linear inequality constraints and nonlinear inequality constraints, e.g. $\Gamma^e = \cap_i \Gamma_i^e$ where

$$\Gamma_i^e = \begin{cases} \{\mathbf{z} : \phi_i(\mathbf{x}) \geq 0\} & i = 1, \ldots, N \\ \{\mathbf{z} : \varphi_q^j(\mathbf{z}) \geq 0\} & i = N + qm + j \\ \{\mathbf{z} : \psi_q^j(\mathbf{z}) \geq 0\} & i = N + (q+h)m + j \\ \{\mathbf{z} : A_{eq}\mathbf{x} = b_{eq}\} & i = N + 2hm + 1 \\ \{\mathbf{z} : A\mathbf{x} \leq b\} & i = N + 2hm + 2 \\ \{\mathbf{z} : \mathbf{y} \leq \mathbf{u}_{max}\} & i = N + 2hm + 3, \end{cases} \tag{18}$$

$\varphi_q^j(\mathbf{z}) := f_q^j(\mathbf{x}) + h_q^j(\mathbf{x})y_q^j$, and $\psi_q^j(\mathbf{z}) := -f_q^j(\mathbf{x}) + h_q^j(\mathbf{x})y_q^j$. φ_q^j and ψ_q^j are smooth as f_q^j and h_q^j are smooth. We just need to check if they are semi-convex. Since f_q^j and h_q^j have bounded derivatives and Hessians and y_q^j is bounded, the Hessian of φ_q^j is bounded below, e.g.

$$\begin{aligned} \nabla^2_{[\mathbf{x},y_q^j]}\varphi_q^j &= \begin{bmatrix} \nabla^2_{\mathbf{x}} f_q^j(\mathbf{x}) + \nabla^2_{\mathbf{x}} h_q^j(\mathbf{x}) y_q^j & \nabla_{\mathbf{x}} h_q^j(\mathbf{x}) \\ \nabla_{\mathbf{x}} h_q^j(\mathbf{x}) & 0 \end{bmatrix} \\ &\succeq \begin{bmatrix} \nabla^2_{\mathbf{x}} f_q^j + \nabla^2_{\mathbf{x}} h_q^j y_q^j - \nabla_{\mathbf{x}} h_q^j \nabla_{\mathbf{x}}^T h_q^j & 0 \\ 0 & -1 \end{bmatrix} \\ &\succeq -\begin{bmatrix} H_q^j & 0 \\ 0 & 1 \end{bmatrix}, \end{aligned} \tag{19}$$

where H_q^j depends on the bounds on $\nabla^2_{\mathbf{x}} f_q^j$, $\nabla^2_{\mathbf{x}} h_q^j$, $\nabla_{\mathbf{x}} h_q^j$ and y_q^j. Similar condition holds for $\nabla^2_{[\mathbf{x},y_q^j]}\psi_q^j$.

Then we need to verify that the interior of the nonlinear inequality constraints is non empty. Consider $\mathbf{x}^*$ and $\mathbf{u}^*$ in Assumption 6. Let $\mathbf{z}^* = [(\mathbf{x}^*)^T, \bar{\mathbf{u}}^T]^T$. Then $F(\mathbf{x}^*) - H(\mathbf{x}^*)\bar{\mathbf{u}} < 0 = F(\mathbf{x}^*) + H(\mathbf{x}^*)\mathbf{u}^* < F(\mathbf{x}^*) + H(\mathbf{x}^*)\bar{\mathbf{u}}$. Hence $\mathbf{z}^* \in \cap_i\{\mathbf{z} : \phi_i(\mathbf{x}) > 0\} \cap \{\mathbf{z} : F(\mathbf{x}) + H(\mathbf{x})\mathbf{y} > 0\} \cap \{\mathbf{z} : -F(\mathbf{x}) + H(\mathbf{x})\mathbf{y} > 0\}$. Thus Γ^e satisfies Assumption 3. □

4.2. The SCFS algorithm

The steps in the SCFS the algorithm are summarized below.

Algorithm 1 (*SCFS*)**.** Solve Problem 1 by:

1. Transform Problem 1 to Problem 11;
2. Initialize $\mathbf{z}^{(0)}$;
3. Apply CFS on Problem 11 by iteratively solving the following problem

$$\mathbf{z}^{(k+1)} = \arg \min_{\mathbf{z} \in \mathcal{F}^e(\mathbf{z}^{(k)})} J(\mathbf{z}), \tag{20}$$

where $\mathcal{F}^e(\mathbf{z}^{(k)})$ (illustrated in Fig. 6) is the convex feasible set of Γ^e constructed according to the three cases in Section 3.1. Denote the solution as $\hat{\mathbf{z}} = [\hat{\mathbf{x}}^T, \hat{\mathbf{y}}^T]^T$;
4. Construct the solution $\hat{\mathbf{w}} = [\hat{\mathbf{x}}^T, \hat{\mathbf{u}}^T]^T$ for Problem 1 from $\hat{\mathbf{z}}$ by setting $\hat{\mathbf{u}}$ to be $-H^{-1}(\hat{\mathbf{x}})F(\hat{\mathbf{x}})$.

4.3. Theoretical results

Theorem 13 (*Convergence of SCFS*)**.** *Algorithm 1 will converge to a local optimum of Problem 1 for any nearly feasible initial value $\mathbf{z}^{(0)}$.*

This is the main result in this paper. Before proving the theorem, we first show that the local optima of Problem 11 is equivalent to the local optima of Problem 1.

Lemma 14. *Any local optimum $\hat{\mathbf{z}} = [\hat{\mathbf{x}}^T, \hat{\mathbf{y}}^T]^T$ of Problem 11 satisfies $|H^{-1}(\hat{\mathbf{x}})F(\hat{\mathbf{x}})| = \hat{\mathbf{y}}$.*

Proof. Let $\hat{\mathbf{u}} = -H^{-1}(\hat{\mathbf{x}})F(\hat{\mathbf{x}})$. For each entry $\hat{u}_q^j$ in $\hat{\mathbf{u}}$, constraint (16) implies that $\hat{u}_q^j = -(h_q^j)^{-1}(\hat{\mathbf{x}})f_q^j(\hat{\mathbf{x}}) \in [-\hat{y}_q^j, \hat{y}_q^j]$. Suppose there exist some q and j such that $|\hat{u}_q^j| < \hat{y}_q^j$. Define $\tilde{\mathbf{z}} := [\hat{\mathbf{x}}^T, \hat{y}_0^1, \ldots, \hat{y}_q^j - \epsilon, \ldots, \hat{y}_{h-1}^m]$ for small $\epsilon \in (0, \hat{y}_q^j - |\hat{u}_q^j|)$. Since $|\hat{u}_q^j| < \hat{y}_q^j$, then $\varphi_q^j(\hat{\mathbf{z}}) > 0$ and $\psi_q^j(\hat{\mathbf{z}}) > 0$. Due to the continuity of φ_q^j and ψ_q^j, $\varphi_q^j(\tilde{\mathbf{z}}) > 0$ and $\psi_q^j(\tilde{\mathbf{z}}) > 0$ for ϵ sufficiently small. Hence $\tilde{\mathbf{z}} \in \Gamma^e$. However $J(\tilde{\mathbf{z}}) < J(\hat{\mathbf{z}})$ by Assumption 2, which contradicts the fact that $\hat{\mathbf{z}}$ is a local optimum. Hence $|\hat{u}_q^j| = \hat{y}_q^j$ for all q and j. □

Proposition 15 (*Equivalence*)**.** *If $\hat{\mathbf{z}} = [\hat{\mathbf{x}}^T, \hat{\mathbf{y}}^T]^T$ with $\hat{\mathbf{y}} = |H^{-1}(\hat{\mathbf{x}})F(\hat{\mathbf{x}})|$ is a local optimum of Problem 11, then $\hat{\mathbf{w}} = [\hat{\mathbf{x}}^T, \hat{\mathbf{u}}^T]^T$ with $\hat{\mathbf{u}} = -H^{-1}(\hat{\mathbf{x}})F(\hat{\mathbf{x}})$ is a local optimum of Problem 1, and vice versa.*

Proof. Since $\hat{\mathbf{z}}$ is a local optimum in Problem 11, there exist Lagrangian multipliers η_q^j, ξ_q^j, $\lambda_i \leq 0$, $\alpha_q^j \geq 0$, $\Lambda_1 \geq 0$ and $\Lambda_2 \in \mathbb{R}^{k_{eq}}$ for all i, j and q such that $\nabla L^r|_{\hat{\mathbf{z}}} = 0$, where the Lagrangian L^r is defined as

$$\begin{aligned} L^r = J &+ \sum_{q,j} (\eta_q^j \varphi_q^j + \xi_q^j \psi_q^j + \alpha_q^j (y_q^j - \bar{u}_q^j)) + \sum_i \lambda_i \phi_i \\ &+ \Lambda_1^T (A\mathbf{x} - b) + \Lambda_2^T (A_{eq}\mathbf{x} - b_{eq}). \end{aligned} \tag{21}$$

Taking the partial derivatives at $\hat{\mathbf{z}}$, we have

$$\begin{aligned} 0 = \nabla_{\mathbf{x}} J_1 &+ \sum \eta_q^j \nabla_{\mathbf{x}} \varphi_q^j + \sum \xi_q^j \nabla_{\mathbf{x}} \psi_q^j + \sum \lambda_i \nabla_{\mathbf{x}} \phi_i \\ &+ A^T \Lambda_1 + A_{eq}^T \Lambda_2, \end{aligned} \tag{22}$$

$$0 = \nabla_{y_q^j} J_2 + \eta_q^j \nabla_{y_q^j} \varphi_q^j + \xi_q^j \nabla_{y_q^j} \psi_q^j + \alpha_q^j. \tag{23}$$

The complementary slackness condition at $\hat{\mathbf{z}}$ is

$$\eta_q^j \psi_q^j = \xi_q^j \varphi_q^j = \lambda_i \phi_i = \alpha_q^j (\hat{y}_q^j - \bar{u}_q^j) = 0, \tag{24}$$

$$\Lambda_1^T (A\hat{\mathbf{x}} - b) = 0. \tag{25}$$

We need to show that (22)–(25) imply the first order optimality condition for Problem 1 at $\hat{\mathbf{w}}$. Define $\beta_q^j, \gamma_q^j \geq 0$ such that

$$\beta_q^j = \begin{cases} \alpha_q^j & u_q^j > 0 \\ 0 & u_q^j \leq 0 \end{cases}, \gamma_q^j = \begin{cases} \alpha_q^j & u_q^j < 0 \\ 0 & u_q^j \leq 0 \end{cases}.$$

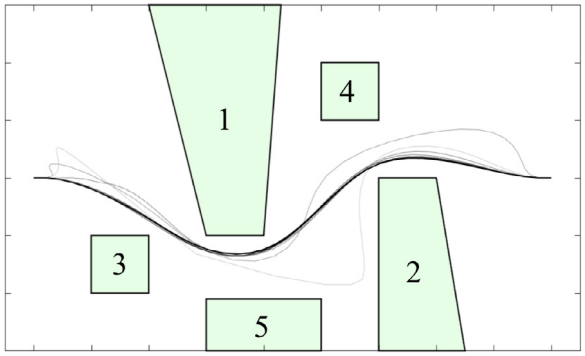
By Lemma 14, $|\hat{\mathbf{u}}| = \hat{\mathbf{y}}$. When $u_q^j = y_q^j > 0$, $\varphi_q^j = 0$ and $\psi_q^j \neq 0$. When $u_q^j = -y_q^j < 0$, $\psi_q^j = 0$ and $\varphi_q^j \neq 0$. By complementary slackness (24) and symmetry of J_2, (23) implies

$$0 = \begin{cases} \nabla_{u_q^j} J_2 + \eta_q^j h_q^j + \beta_q^j & u_q^j > 0 \\ -\nabla_{u_q^j} J_2 + \xi_q^j h_q^j - \gamma_q^j & u_q^j < 0 \\ \eta_q^j h_q^j + \xi_q^j h_q^j & u_q^j = 0 \end{cases}. \tag{26}$$

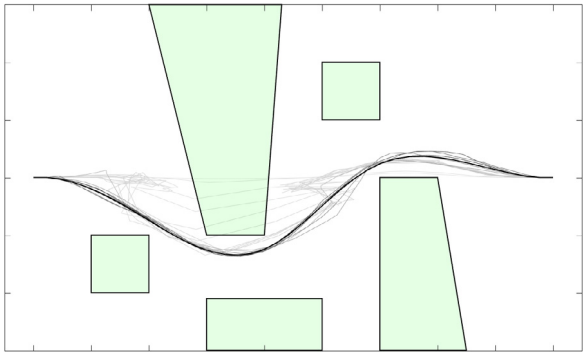
For the third equation in (26), since $h_q^j > 0$ and $\eta_q^i, \xi_q^j \leq 0$, then $\eta_q^i = \xi_q^j = 0$ when $u_q^j = 0$. Define

$$\delta_q^j = \begin{cases} \eta_q^j & u_q^j > 0 \\ -\xi_q^j & u_q^j < 0 \\ 0 & u_q^j = 0 \end{cases}.$$

Then the complementary slackness (24) implies that $\eta_q^j \nabla_{\mathbf{x}} \varphi_q^j + \xi_q^j \nabla_{\mathbf{x}} \psi_q^j = \delta_q^j (\nabla_{\mathbf{x}} f_q^j + \nabla_{\mathbf{x}} h_q^j u_q^j)$. Hence (22) and (26) imply that at $\hat{\mathbf{w}}$,



(a) The trajectories in SCFS.



(b) The trajectories in ITP-J.

Fig. 7. The motion planning problem in 2D.

$$0 = \nabla_{\mathbf{x}} J_1 + \sum \delta_q^j \nabla_{\mathbf{x}} (f_q^j + h_q^j u_q^j) + \sum \lambda_i \nabla_{\mathbf{x}} \phi_i + A^T \Lambda_1 + A_{eq}^T \Lambda_2, \tag{27}$$

$$0 = \nabla_{u_q^j} J_2 + \delta_q^j \nabla_{u_q^j} (f_q^j + h_q^j u_q^j) + \beta_q^j + \gamma_q^j. \tag{28}$$

The above equation is equivalent to $\nabla L|_{\hat{\mathbf{w}}} = 0$ where the Lagrangian L is defined as

$$L = J + \sum_i \lambda_i \phi_i + \Lambda_1^T (A\mathbf{x} - b) + \Lambda_2^T (A_{eq}\mathbf{x} - b_{eq}) + \sum_{q,j} \left(\delta_q^j (f_q^j + h_q^j u_q^j) + \beta_q^j (u_q^j - \bar{u}_q^j) + \gamma_q^j (-u_q^j - \bar{u}_q^j) \right). \tag{29}$$

It is easy to check that primal feasibility (30), dual feasibility (31) and complementary slackness (32) are all satisfied.

$$\hat{\mathbf{x}} \in \Gamma, -\bar{\mathbf{u}} \le \hat{\mathbf{u}} \le \bar{\mathbf{u}}, F(\hat{\mathbf{x}}) + H(\hat{\mathbf{x}})\hat{\mathbf{u}} = 0, \tag{30}$$

$$\beta_i \ge 0, \gamma_i \ge 0, \tag{31}$$

$$\delta_q^j (f_q^j + h_q^j \hat{u}_q^j) = \beta_q^j (\hat{u}_q^j - \bar{u}_q^j) = \gamma_q^j (-\hat{u}_q^j - \bar{u}_q^j) = 0. \tag{32}$$

Hence $\hat{\mathbf{w}}$ is a local optimum in Problem 1.

To prove the other direction, we just need to reverse the above process, which will not be elaborated due to space limitations. □

Proof of Theorem 13. For any nearly feasible initial value $\mathbf{z}^{(0)}$, the sequence $\{\mathbf{z}^{(k)}\}$ in Algorithm 1 converges to a local optimum of Problem 11 by Theorem 8 and Lemma 12. Hence $\hat{\mathbf{z}} = \lim_{k\to\infty} \mathbf{z}^{(k)}$ is a local optimum of Problem 11. Then $\hat{\mathbf{w}}$ is a local optimum of Problem 1 by Proposition 15. □

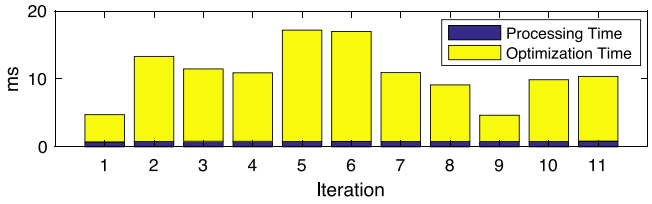


Fig. 8. Decomposition of the computation time in SCFS.

5. Numerical examples

The performance of SCFS is illustrated in a motion planning problem for a mobile robot in a crowded environment as shown in Fig. 7. State $x \in \mathbb{R}^2$ denotes the planar position of the vehicle. Control $u \in \mathbb{R}$ is the yaw rate. Then $\mathbf{x} = [x_0^T, x_1^T, \ldots, x_h^T]^T \in \mathbb{R}^{2(h+1)}$ is the planar trajectory. $\mathbf{u} = [u_0^T, u_1^T, \ldots, u_{h-1}^T]^T \in \mathbb{R}^h$ is the trajectory of yaw rate. $\mathbf{x}$ and $\mathbf{u}$ satisfy the dynamic equation in (7) for any q. The area occupied by the jth obstacle at time step q is denoted as $\mathcal{O}_{j,q} \in \mathbb{R}^2$. Define $\phi_{j,q}(\mathbf{x}) = d(x_q, \mathcal{O}_{j,q}) - d_{min}$ where $d(\cdot,\cdot)$ computes the signed distance between the vehicle and the jth obstacle and $d_{min} > 0$ is the minimum distance requirement. The vehicle needs to travel from x^{start} to x^{goal} with small yaw rate without colliding with any obstacle. The optimization problem is formulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) = w_1 \|\mathbf{x} - \mathbf{x}^r\|_Q^2 + w_2 \|\mathbf{x}\|_S^2 + w_3 \|\mathbf{u}\|_R^2 \\ s.t. \quad & x_0 = x^{start}, x_h = x^{goal}, -\bar{\mathbf{u}} \le \mathbf{u} \le \bar{\mathbf{u}} \\ & (x_q - x_{q-1}) \times (x_{q+1} - x_q) = \|x_q - x_{q-1}\|^2 u_q \\ & \phi_{j,q}(\mathbf{x}) \ge 0, \forall j, \forall q = 1, \ldots, h-1. \end{aligned}$$

The cost function is designed to be quadratic where $w_1, w_2, w_3 \in \mathbb{R}^+$. $\|\mathbf{x} - \mathbf{x}^r\|_Q^2 := (\mathbf{x} - \mathbf{x}^r)^T Q (\mathbf{x} - \mathbf{x}^r)$ penalizes the distance from the target trajectory to a reference trajectory $\mathbf{x}^r$. $\|\mathbf{x}\|_S^2 := \mathbf{x}^T S \mathbf{x}$ concerns with the smoothness of the trajectory. $\|\mathbf{u}\|_R^2 := \mathbf{u}^T R \mathbf{u}$ penalizes the magnitude of yaw rate. In a bounded environment X, the second order derivatives of the nonlinear equality constraint are also bounded. Hence the lower bound of Hessian in (12) can be chosen manually offline. Admittedly, the requirement of the lower bound of Hessian introduces inconvenience in implementing Algorithm 1. Relaxation of the lower bound will be considered in the future work.

The simulation was run in C++ on a MacBook of 2.3 GHz using Intel Core i7. Planning horizon is $h = 40$. The reference trajectory $\mathbf{x}^r$ is a straight line connecting the start point and the goal point (which is not shown in the figure). The green sections correspond to the static obstacles $\mathcal{O}_{j,q}$. The initial value is chosen as $\mathbf{x}^{(0)} = \mathbf{x}^r$ and $\mathbf{u}^{(0)} = 0$. The termination condition is that the step size $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$ is smaller than 0.001.

Algorithm 1 (SCFS) converges after 11 iterations with total computation time 119.9 ms and final cost 1408.25. Eq. (20) is solved using KNITRO [16] interior-point method. The trajectories under different iterations are shown in Fig. 7. The dark curve is the optimal path. The gray curves are the solutions before convergence, the lighter the earlier in iterations. The decomposed computation time in SCFS is shown in Fig. 8. The computation time during each iteration is composed of two parts, the processing time, e.g. the time spent to transform the problem to a convex problem, and the optimization time, e.g. the time spent to solve the convex optimization. The processing time for each iteration is around 0.80 ms (7.3% of the total time), which is very consistent through iterations. The optimization time varies from iteration to iteration.

Table 1
The comparison among various algorithms.

Method	Computation time (ms)	Iterations	Cost
SCFS	119.9	11	1408.25
ITP-J	419.4	94	1489.72
ITP	14031.4	90	1489.58
ACT-J	5970.2	718	1489.71
SQP-J	20632.7	848	3487.26

For comparison, the planning problem is also solved using built-in non convex optimization solvers in KNITRO directly, where the same termination condition is applied. In Table 1, the computation time in interior point (ITP), sequential quadratic programming (SQP), and active set (ACT) methods is compared. ITP-J, ACT-J and SQP-J refer to the case that we provide Jacobian to the solver. Otherwise, the gradient is computed numerically by central finite differences. The Hessian is computed numerically using quasi-Newton symmetric rank 1 method for all cases, which has desired accuracy and relatively small computation load. For ITP, the computation time reduced 96.7% when Jacobian is provided.[5] As shown in the table, the other two methods (ACT and SQP) are much slower than ITP and require more iterations to converge. As ITP-J is most efficient, we only provide detailed comparison between SCFS and ITP-J. Fig. 9 shows the run time statistics of the two methods, e.g cost $J(\mathbf{x})$ and feasibility (absolute value of the maximum violation of the constraints) during each iteration. The cost profiles have similar characteristics under the two methods, i.e. the cost first jumps up to make the problem feasible, then drops down to the optimal value. The difference is that the cost drops down much faster in SCFS than in ITP-J, as shown in the zoomed-in figure. Moreover, ITP-J finds a local optimum with higher cost than SCFS. The trajectories in ITP-J are shown in Fig. 7(b), where the color of the trajectories corresponds to the iteration number, the lighter the earlier. Different from the trajectories in SCFS, the trajectories in the early iterations of ITP-J are not feasible and the step size in ITP-J is smaller.

The performance of SCFS under different conditions is shown in Fig. 10, where the number of sampling points h changes from 20 to 40, and the number of obstacles changes from 1 to 5. The start point and the goal point are fixed. We use "l obstacles" to denote the case that we only consider polygons 1 to l in Fig. 7(a). The average processing time in SCFS scales linearly with respect to h and l. In general, the total computation time in SCFS also grows with respect to h and l. However, the trend in the total computation time is not as clear as in the average processing time, since the optimization time for each iteration and the number of iterations up to convergence are highly problem-specific and sensitive to initial conditions. Nonetheless, SCFS always outperforms ITP-J in terms of computation time as shown in Fig. 7(a). In those cases, SCFS either finds a better optima than ITP-J or converges to the same local optima as ITP-J.

6. Conclusion

This paper discussed the geometric structure of the trajectory optimization problem for nonlinear robotic systems. Based on the geometric understanding, the slack convex feasible set algorithm was proposed to solve the constrained trajectory optimization problem by relaxation and convexification. The feasibility, convergence and optimality of the algorithm were proved. Simulation results also validated the efficiency of the algorithm and illustrated its advantage over existing algorithms.

[5] Note that the number of iterations and the final cost are not identical in ITP-J and ITP. That is due to the fact that the constraint is non differentiable at some points, e.g. at a corner point of an obstacle. At those points, the sub-gradients computed by the Jacobian and by the central finite differences are different.

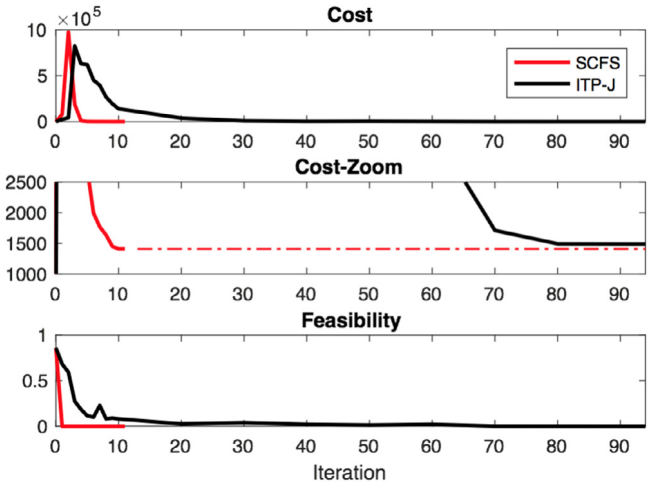


Fig. 9. Profiles of cost and feasibility in SCFS and ITP-J.

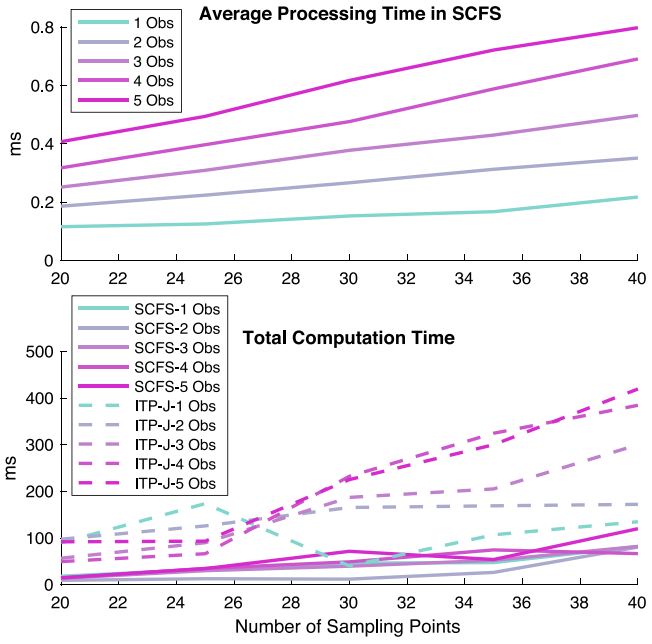


Fig. 10. Performance of SCFS under different conditions.

References

[1] J.-C. Latombe, Robot Motion Planning, Vol. 124, Springer Science & Business Media, 2012.
[2] E. Frazzoli, M.A. Dahleh, E. Feron, Real-time motion planning for agile autonomous vehicles, J. Guid. Control Dyn. 25 (1) (2002) 116–129.
[3] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, J.P. How, Real-time motion planning with applications to autonomous urban driving, IEEE Trans. Control Syst. Technol. 17 (5) (2009) 1105–1118.
[4] C. Liu, M. Tomizuka, Algorithmic safety measures for intelligent industrial co-robots, in: Proceedings of 2016 IEEE International Conference on Robotics and Automation, IEEE, 2016, pp. 3095–3102.
[5] T.M. Howard, C.J. Green, A. Kelly, Receding horizon model-predictive control for mobile robot navigation of intricate paths, in: Field and Service Robotics, Springer, 2010, pp. 69–78.
[6] M. Tawarmalani, N.V. Sahinidis, Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Vol. 65, Springer Science & Business Media, 2002.

[7] P. Spellucci, A new technique for inconsistent qp problems in the sqp method, Math. Methods Oper. Res. 47 (3) (1998) 355–400.
[8] K. Tone, Revisions of constraint approximations in the successive qp method for nonlinear programming problems, Math. Program. 26 (2) (1983) 144–152.
[9] T.A. Johansen, T.I. Fossen, S.P. Berge, Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming, IEEE Trans. Control Syst. Technol. 12 (1) (2004) 211–216.
[10] J. Schulman, J. Ho, A.X. Lee, I. Awwal, H. Bradlow, P. Abbeel, Finding locally optimal. collision-free trajectories with sequential convex optimization, in: Robotics: Science and Systems, Vol. 9, 2013, pp. pp. 1–10.
[11] J. Van Den Berg, P. Abbeel, K. Goldberg, LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information, Int. J. Robot. Res. 30 (7) (2011) 895–913.
[12] N. Ratliff, M. Zucker, J.A. Bagnell, S. Srinivasa, Chomp: Gradient optimization techniques for efficient motion planning, in: Proceedings of 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 489–494.
[13] C. Liu, C.-Y. Lin, M. Tomizuka, The convex feasible set algorithm for real time optimization in motion planning, SIAM J. Control Optim. (2016) in review, arXiv:1709.00627.
[14] C. Liu, W. Zhan, M. and Tomizuka, Speed profile planning in dynamic environments via temporal optimization, in: Proceedings of 2017 IEEE Intelligent Vehicle Symposium, 2017.
[15] C. Liu, C.-Y. Lin, Y. Wang, M. Tomizuka, Convex feasible set algorithm for constrained trajectory smoothing, in: Proceedings of 2017 American Control Conference, 2017.
[16] R.H. Byrd, J. Nocedal, R.A. Waltz, Knitro: An Integrated Package for Nonlinear Optimization, Springer US, Boston, MA, 2006, pp. 35–59.