



A bilevel optimal motion planning (BOMP) model with application to autonomous parking

Shenglei Shi¹ · Youlun Xiong¹ · Jiankui Chen¹ · Caihua Xiong¹

Received: 22 July 2019 / Accepted: 10 October 2019 / Published online: 12 November 2019
© The Author(s) 2019

Abstract

In this paper, we present a bilevel optimal motion planning (BOMP) model for autonomous parking. The BOMP model treats motion planning as an optimal control problem, in which the upper level is designed for vehicle nonlinear dynamics, and the lower level is for geometry collision-free constraints. The significant feature of the BOMP model is that the lower level is a linear programming problem that serves as a constraint for the upper-level problem. That is, an optimal control problem contains an embedded optimization problem as constraints. Traditional optimal control methods cannot solve the BOMP problem directly. Therefore, the modified approximate Karush–Kuhn–Tucker theory is applied to generate a general nonlinear optimal control problem. Then the pseudospectral optimal control method solves the converted problem. Particularly, the lower level is the J_2 -function that acts as a distance function between convex polyhedron objects. Polyhedrons can approximate objects in higher precision than spheres or ellipsoids. As a result, a fast high-precision BOMP algorithm for autonomous parking concerning dynamical feasibility and collision-free property is proposed. Simulation results and experiment on Turtlebot3 validate the BOMP model, and demonstrate that the computation speed increases almost two orders of magnitude compared with the area criterion based collision avoidance method.

Keywords Autonomous parking · Optimal control · Bilevel optimal motion planning (BOMP) · J_2 -function

1 Introduction

Real-time collision-free motion planning and control for autonomous vehicles have received a considerable amount of attentions, and share many research methods with robotics literature (LaValle 2006; González et al. 2016; Nilsson et al. 2016; Rasekhipour et al. 2016; Ye et al. 2018; Vorobieva et al. 2015; Du and Tan 2015; Upadhyay and Ratnoo 2018; Muller et al. 2007; Xu et al. 2018; Dolgov et al. 2010; Likhachev and Ferguson 2009; Tazaki et al. 2017; Liu et al. 2017; Robinson et al. 2018; Li and Shao 2015; Li et al. 2016;

Zips et al. 2016; Khatib 1986; Oetiker et al. 2009). Typically, autonomous parking is a critical maneuver especially in big narrow cities. Separated methods (Vorobieva et al. 2015; Du and Tan 2015; Upadhyay and Ratnoo 2018; Muller et al. 2007; Xu et al. 2018; Dolgov et al. 2010; Likhachev and Ferguson 2009; Tazaki et al. 2017) are common approaches for parking problems, that vehicle path planning and path tracking are handled separately. Direct learning (Liu et al. 2017) is a novel and simple approach that learns the mapping relation between control inputs and parking trajectories. However, for complex high-precision parking problems, combined approaches (Li and Shao 2015; Li et al. 2016) are more effective, that vehicle motion planning and control are treated as a unified optimal control problem. In this paper, we also treat the parking problem as a combined optimal control problem.

Optimal control (Betts 2010; Bryson 2018; Ross and Karpenko 2012) is a remarkable method to generate high-quality trajectories for robots and has achieved great success in practical applications. It considers robot dynamics and other trajectory constraints in a compact and unified form, and it can deal with any predefined optimization

✉ Youlun Xiong
famt@hust.edu.cn

Shenglei Shi
shshlei@hust.edu.cn

Jiankui Chen
Chenjk@hust.edu.cn

Caihua Xiong
chxiong@hust.edu.cn

¹ School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

objectives. The application of the optimal control method in robot motion planning involves two essential constraints, robot dynamics constraint, and geometry collision-free constraint. And the application challenges lie in how to represent the geometry collision-free constraints effectively. The point-point distance of circles in Robinson et al. (2018) and area criterion of rectangles in Li and Shao (2015) are both straightforward collision avoidance methods. However, circle bounding volume approximation is too conservative to realize motion in complex and high-precision scenarios. And area criterion of rectangles is nonlinear and redundant, such that the optimal control problem is difficult to solve. In addition, the collision-free constraints for high-precision motion planning problems in 3D environment are still difficult to be built. We will show in this paper the J_2 -function (Xiong 1987; Xiong and Ding 1989) settles these challenges, and achieves fast high-precision motion planning for autonomous parking. The J_2 -function is linear programming for collision checking between convex polyhedrons in any dimensional space, and it behaves as a distance function that $J_2 > 0$ indicates collision free. Treating $J_2 \geq \delta$ (δ is a positive safety distance, and J_2 is a function of robot trajectory) as a constraint for the overall optimal control problem, a special J_2 -function based bilevel optimal motion planning (BOMP) model for autonomous parking is obtained.

1.1 Previous work

Before reviewing the various parking motion planning approaches in the literature, we first show a typical three-point maneuver in parallel parking, as shown in Fig. 1. At first, the vehicle moves forward from an initial point to an intermediate point (maneuver A). Subsequently, it follows a collision-free path or trajectory backward to the destination point (maneuver B). If the process only contains one maneuver that connects the initial and terminal configurations without changing moving direction, then it is called a single maneuver process; otherwise, it is a multi-maneuver parking process.

Using the theorems from Dubins (1957), Reeds and Shepp (1990), Wang et al. (2009), many geometric path planning methods (Vorobieva et al. 2015; Du and Tan 2015; Upadhyay and Ratnoo 2018) were proposed in which the path is composed of concatenating arcs and line segments.

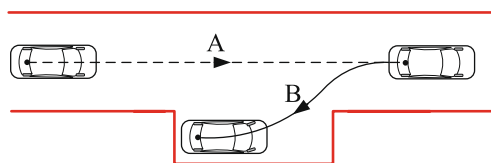


Fig. 1 Typical three-point maneuver in parallel parking

However, the curvature at the transition point between segments is discontinuous, and this implies stopping the vehicle and steering the wheel that can wear the tires and waste time. For single maneuver parking, a function $y = f(x)$ exists in the plane coordinate system, and many approaches use a high-order differentiable function to smooth the planned path. Whereas, the feasibility conditions for single maneuver parking are critical, such as conditions on minimum parking slot length, initial vehicle configuration relative to the parking slot, and the surrounding obstacles. In Vorobieva et al. (2015), the detailed comparisons between single maneuver and multi-maneuver parallel parking using geometric methods were carried out. In Du and Tan (2015), the autonomous vertical parking maneuvers were classified by the segments number and steering direction, and this planning approach achieves high efficiency and high parking success rate. Focusing on the issue that the vehicle may not stop exactly at the prescribed intermediate point in the three-point maneuver process, Upadhyay and Ratnoo (2018) proposed a four parameter logistic curve to connect the terminal point to a neighborhood of the intermediate point.

Graph searching is another type of approach for autonomous parking. At first, the discretization of the environment is applied to construct a graph. Then informative heuristics are used to guide the search to generate a feasible path to the parking spot. Dolgov et al. (2010) applied a variant of A^* search in vehicle state space to generate a feasible path first, then took advantages of the numerical optimization method to shorten and smooth the planned path. Likhachev and Ferguson (2009) utilized multi-resolution discretization method and anytime dynamic A^* search to obtain long dynamically feasible maneuvers in real time. In Tazaki et al. (2017), a new multi-resolution discretization on manually predefined guidelines was proposed, then a graph-like roadmap was created to connect each guideline partition. However, this method heavily relies on guidelines that may cause unnecessary maneuvers. A common precomputation technique is used in these searching methods to reduce online path planning computation complexity. Such as the shortest path from a point to its neighborhood points (Dolgov et al. 2010), the occupied space of the vehicle after taking an action (Likhachev and Ferguson 2009), and the path from guideline partition to partition (Tazaki et al. 2017) are all precomputed.

In Zips et al. (2016), a fast optimization method was used for narrow environment parking, but the heuristic information leads to a non-optimal solution. Artificial potential field (APF) method (Khatib 1986) has made great achievements for robot motion planning in static and dynamic environments. Moreover, a recent application in autonomous parking (Oetiker et al. 2009) demonstrates its capability for nonholonomic systems. However, the APF method is likely to get stuck in a sub-optimal path, and it cannot

handle the multi-maneuver parking because of local planning characteristic.

Previous approaches only solve vehicle path planning problem, but at some times, vehicle trajectory planning concerning vehicle kinematics and dynamics constraints is more useful and important. Kant and Zucker (1986) decomposed the trajectory planning as a combination of path planning and velocity planning, and the decomposition demonstrates apparently that trajectory planning problem is more difficult than path planning problem. In Muller et al. (2007), a two-step method was applied for trajectory planning. In the first step, a collision-free path is generated using sampling based method without considering the kinematics constraint. Then in the second step, a local trajectory planner solved the nonholonomic constrained problem. In Liu et al. (2017), a neural network based parking trajectory planning was proposed with constant speed input and discretized steering angle outputs. In this method, the steering angle is discontinuous. Moreover, obstacle avoidance and multi-maneuver parking cannot be treated easily. In Haddad et al. (2010), an optimization approach along with trapezoidal velocity constraints simplification solved the trajectory planning problem, and the simulation results showed considerable computation time reduction without losing much quality of the solution. In Li and Shao (2015) and Li et al. (2016), the optimal control method is applied to solve the optimal trajectories for autonomous parking. In Li and Shao (2015), rectangular obstacles together with area criterion were used to check collisions, and eight nonlinear constraints are generated between a pair of rectangles. In Li et al. (2016), collision avoidance in regular parallel parking was realized by restricting the vehicle between two function boundaries (red lines in Fig. 1).

1.2 Contributions and organization

In this paper, we present a general BOMP model in which the upper level is an optimal control problem designed for vehicle nonlinear dynamics, while the lower level is the J_2 -function linear programming for geometry collision-free constraint (in Sect. 2). The BOMP model contains an embedded linear programming problem as constraint for the overall optimal control problem. This hierarchy property makes the problem complicated to solve, and traditional optimal control methods cannot solve the BOMP problem. Hence, the modified approximate Karush–Kuhn–Tucker theory (Dutta et al. 2013) is used to simplify the model to a traditional optimal control problem. Then the pseudospectral optimal control method (Ross and Karpenko 2012) is used to solve the converted problem, and an iterative BOMP algorithm is obtained (in Sect. 3). Twelve representative complex parking problems are designed to evaluate the efficiency of the BOMP algorithm, and they can be regarded as

benchmarks for future parking approaches. The highlights of this paper are:

1. The BOMP model combines both advantages of the flexibility of optimal control and the simplicity of linear programming, which makes optimal control a fast and high-precision method for complex motion planning problems.
2. The BOMP algorithm benefits by the iterative convergence strategy. At each iterative, the approximate problem is easy to solve an approximate optimal trajectory to initialize the next iterative. Simulations in autonomous parking demonstrate that the computation speed increases almost two orders of magnitude compared with the area criterion based method (in Sect. 4).

Finally, concluding remarks are given in Sect. 5 and the issues to be researched in this field are also pointed out.

2 Bilevel optimal motion planning

Optimization problem finds an optimal point, while optimal control problem finds an optimal trajectory. So it is a natural question: how to use the optimal control method to solve vehicle trajectory generation and optimization problem in complex and high-precision scenarios? As stated in Sect. 1, the geometry collision-free constraint is the crucial matter. In this section, we will present the BOMP model with an embedded J_2 -function linear programming constraint.

2.1 Vehicle kinematics and basic parking constraints

Vehicle is a typical nonholonomic constrained system, and its motion planning is very complicated. Without loss of generality, the front steering wheels vehicle is considered and the following kinematics model is used:

$$\begin{cases} \dot{x}(t) = v(t) \cos \theta(t), & \dot{y}(t) = v(t) \sin \theta(t) \\ \dot{\theta}(t) = v(t) \tan \alpha(t)/l, & \dot{\alpha}(t) = \omega(t) \end{cases} \quad (1)$$

where $\mathbf{q} = (x, y, \theta) \in \mathcal{R}^2 \times \mathcal{S}$ is the configuration of the vehicle coordinate $\{C\}$ with respect to the world coordinate $\{W\}$ which origins at one corner point of the parking spot, see Fig. 2. The vehicle coordinate $\{C\}$ origins at the mid-point of the rear wheel axis, (x, y) describes the vehicle position, and θ denotes the orientation. v denotes the linear velocity of point (x, y) , α denotes the steering angle of the front wheel and ω denotes the steering velocity.

As the parking process is in low speed and to bound the trajectory curvature and its derivative (Li and Shao 2015),

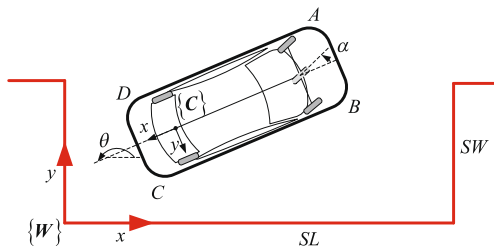


Fig. 2 The coordinate systems attached to the vehicle and the parking spot

the following mechanical and physical constraints are considered.

$$|v(t)| \leq v_{\max}, \quad |\alpha(t)| \leq \alpha_{\max}, \quad |\omega(t)| \leq \omega_{\max} \quad (2)$$

And the initial and the final state constraints are treated as:

$$\begin{cases} x(0) = x_0, y(0) = y_0, \theta(0) = \theta_0 \\ v(0) = 0, v(t_f) = 0, |\theta(t_f)| \leq \epsilon_\theta \\ |(A_y(t_f) + B_y(t_f) - SW)/2| \leq \epsilon_p \\ |(C_y(t_f) + D_y(t_f) - SW)/2| \leq \epsilon_p \\ |(A_x(t_f) + D_x(t_f) - SL)/2| \leq \epsilon_p \\ |(B_x(t_f) + C_x(t_f) - SL)/2| \leq \epsilon_p \end{cases} \quad (3)$$

where (x_0, y_0, θ_0) is the vehicle initial configuration, (A_x, A_y) is the coordinate of vehicle corner point A described in $\{W\}$, (B_x, B_y) , (C_x, C_y) and (D_x, D_y) have the similar meanings, $\epsilon_p \geq 0$ and $\epsilon_\theta \geq 0$ denote the position deviation and the orientation deviation of the vehicle central axis with respect to the parking spot central axis, respectively.

2.2 Simple collision avoidance constraints

The performances of robot collision-free motion planning algorithms depend highly on the geometry collision avoidance methods used between robot and the environment. In real-world, vehicles are always abstracted as rectangles (cubes) or overlapping circles (spheres) to simplify the collision avoidance problem. In this section, the general point-point distance constraints between circles (or spheres) and area criterion constraints between convex polygons are summarized.

Suppose obstacle A is approximated by a circle or sphere of radius r_a that covers it, and robot B is approximated by a circle or sphere of radius r_b . The centers of the approximate objects are \mathbf{a} and \mathbf{b} , respectively. Then the geometry collision-free constraint between A and B is approximated as:

$$d(\mathbf{a}, \mathbf{b}) - r_a - r_b \geq \delta \quad (4)$$

where $d(\mathbf{a}, \mathbf{b})$ denotes the Euclidean distance between the two points, and δ is a positive constant safety distance (δ has the same meaning in the whole paper). Usually, the robot and the obstacles are approximated by several overlapping circles or spheres to increase approximate precision, in this case, the geometry collision-free constraints are still straightforward.

Suppose plane obstacle A and plane robot B are approximated by convex polygons with n_a vertexes and n_b vertexes, respectively. The vertexes coordinates of the polygons are $\{\mathbf{a}_i\}$, $i = 1, 2, \dots, n_a$ and $\{\mathbf{b}_j\}$, $j = 1, 2, \dots, n_b$, respectively. Then the area criterion can describe the geometry collision-free constraints between A and B :

$$\begin{aligned} \sum_{j=1}^{n_b-1} S_{\Delta a_i b_j b_{j+1}} + S_{\Delta a_i b_{n_b} b_1} - S_B &\geq \delta, \quad i = 1, 2, \dots, n_a \\ \sum_{i=1}^{n_a-1} S_{\Delta b_j a_i a_{i+1}} + S_{\Delta b_j a_{n_a} a_1} - S_A &\geq \delta, \quad j = 1, 2, \dots, n_b \end{aligned} \quad (5)$$

where S_{Δ} is the area of a triangle, S_A and S_B are the areas of the two approximate polygons. Moreover, for convex polyhedrons in 3D environment, the volume criterion can derive the collision avoidance constraints for.

As we can see, for a pair of polygons, there are $n_a + n_b$ nonlinear constraints in (5), and this number is usually larger than the circle approximation method. However, in theory, a complex geometry can be incorporated by several polygons or polyhedrons in any precision and low overlapping rate with fewer approximate objects than the circle or sphere approximation method. And to achieve planning and control in scenarios with long and thin obstacles, the polygon or polyhedron approximation is more efficient. Therefore, the convex polygon or polyhedron approximation method will play an important role in complex and high-precision robot motion planning problems. However, the constraints in (5) are high nonlinear and redundant. Therefore, neither collision avoidance methods (4) and (5) can be used to general fast high-precision motion planning problems.

2.3 BOMP model

Consider vehicle state variable \mathbf{x} as $\mathbf{x} = (x, y, \theta, \alpha)$ and the control input $\mathbf{u} = (v, \omega)$, then vehicle kinematics model (1) can be abstracted as $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$. The optimal control problem tries to solve an optimal smooth trajectory corresponding to this differential equation constraints and with respect to a specified performance index. Consider a convex polyhedron robot B moves around a static convex polyhedron obstacle A (a case that a rectangular vehicle moves

around rectangular vehicle obstacles). Since the J_2 -function (Xiong 1987; Xiong and Ding 1989) acts as a distance function between A and B , $J_2 \geq \delta$ is added as an embedded constraint, the BOMP model is obtained:

$$\min \quad \beta g(\mathbf{x}(0), \mathbf{x}(t_f), t_f) + (1 - \beta) \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (6)$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (6a)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \quad (6b)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) \in \mathbf{X}_f \quad (6c)$$

$$J_2(\mathbf{x}) \geq \delta \quad (6d)$$

Equation (6a) represents vehicle dynamics constraints. Equation (6b) describes constraints on vehicle state and control variables, e.g., constraints (2). And Eq. (6c) corresponds to the parking initial and final state constraints (3), where \mathbf{x}_0 and \mathbf{X}_f are vehicle initial state and final stopping region, respectively. $J_2(\mathbf{x})$ represents collision avoidance constraint on vehicle state. To simplify the understanding of the BOMP model, the exact J_2 -function form is omitted here and can be found in the appendix. The readers just understand that the constraint (6d) acts the same role as (4) or (5), but J_2 is the optimal value of a linear programming problem. The cost function $g(\mathbf{x}(0), \mathbf{x}(t_f), t_f)$ depends only on the initial state $\mathbf{x}(0)$, final state $\mathbf{x}(t_f)$ and the completion time t_f , while $L(\mathbf{x}(t), \mathbf{u}(t))$ describes some objective along the trajectory such as the energy consumption and $0 \leq \beta \leq 1$ is the weight. A particular performance index is to minimize the weight sum of completion time and energy consumption, and it's used in Sect. 4.1.

$$t_f + \int_0^{t_f} v^2(t) dt \quad (7)$$

3 The BOMP model solution

3.1 BOMP model analysis

The BOMP model contains an optimization problem within the constraints of the upper optimal control problem. Moreover, the lower optimization problem depends on the continuous robot state trajectory such that it is also infinite dimensional. To our knowledge, this particular model has not appeared in the literature, but the complexity of this model can be seen indirectly from some other problems. In Benita and Mehrlitz (2016), a relative simple bilevel optimal control problem is considered. The upper level is an optimal control problem concerning ordinary differential equations, control constraints, initial and final state constraints, and the lower

level problem depends only on the final state of the physical system and is finite dimensional. In this case, nonsmooth analysis, optimization in Banach spaces and bilevel optimization (Sinha et al. 2017, 2018) are used to derive the necessary linearized Pontryagin-type optimality conditions. By nonconvex, non-differentiable or possibly disconnected, the hierarchical bilevel optimization problem is intrinsically difficult. Even for the most straightforward linear-linear bilevel optimization problem, it was proven to be strong NP-hard (Hansen et al. 1992) and that merely evaluating the optimality is also NP-hard (Vicente et al. 1994).

Two significant challenges are classified from the BOMP model, the bilevel problem, and the optimal control problem. Optimal control has experienced considerable development in mathematics and engineering. In particular, pseudospectral optimal control (PSOC) method (Ross and Karpenko 2012) satisfies the differential equations globally and treats integral by an implicit Runge–Kutta method, such that this method achieves high order accuracy, high order stability, and exponential convergence speed. Therefore, the PSOC method is selected. For the bilevel optimization problem, the Karush–Kuhn–Tucker (KKT) reformulation of the lower-level optimization problem is always used to construct a traditional single level optimization problem (Albrecht et al. 2012). The KKT reformulation can not be applied to the BOMP model seemingly since the BOMP problem is an optimal control problem rather than an optimization problem. However, the PSOC algorithm discretizes the whole variables and approximates them as polynomials, such that it actually solves a large scale sparse nonlinear optimization problem (NLP). In this case, the KKT reformulation can be directly used to the discretized NLP. Furthermore, differential equations do not occur in the lower level of the BOMP model, the continuous KKT reformulation (the Lagrange multipliers are also trajectory functions of time) concept can be utilized unambiguously.

However, due to the nonconvexities in the KKT conditions, even the upper-level problem is also convex, the converted problem is still hard to solve. Besides, the complementary constraint is intrinsically combinatorial and makes the extent of violation of KKT conditions in a small neighborhood of the KKT point nonsmooth (Dutta et al. 2013). Hence, the convergence property in the neighborhood of the optimal value is poor, and it cannot provide efficient information to determine whether current value is close enough to the optimal value. Dutta et al. (2013) proposed a modified approximate KKT (MAKKT) theory that the KKT conditions are relaxed and the violation in the neighborhood is smooth enough. Since the KKT conditions are relaxed, an iterative strategy is used to decrease the relaxation factor to approach the optimal value. Therefore, the MAKKT theory and iterative convergence strategy are selected in this paper.

Table 1 The IPOPT algorithm options

| Parameter | Description | Setting |
|-----------|---|------------|
| tol | Desired convergence tolerance | $1e^{-12}$ |
| a_tol | Acceptable convergence tolerance | $1e^{-16}$ |
| cv_tol | Desired/acceptable threshold for the constraint violation | $1e^{-12}$ |
| acv_tol | | |
| c_tol | Desired/acceptable threshold for the complementarity conditions | $1e^{-4}$ |
| ac_tol | | |
| d_tol | Desired/acceptable threshold for the dual infeasibility | 1 |
| ad_tol | | |
| solver | Linear solver used for step computation | MA86 |

3.2 The BOMP algorithm

Given the problematic nature of the BOMP model, it is helpful to reduce the overall bilevel optimal control problem to a traditional single level optimal control problem. According to the optimality conditions of linear programming (Wright and Nocedal 1999), the continuous MAKKT form of the lower problem (12) is:

$$\begin{cases} \mathbf{Q}(\mathbf{x})\mathbf{p}(t) = \mathbf{b}, \quad \mathbf{p}(t) \geq \mathbf{0} & (8a) \\ \|\mathbf{c} - \lambda(t) + \mathbf{Q}^T(\mathbf{x})\mathbf{v}(t)\| \leq \sqrt{\epsilon}, \quad \lambda^T(t)\mathbf{p}(t) \leq \epsilon & (8b) \\ \lambda(t) \geq \mathbf{0} & (8c) \end{cases}$$

where the Lagrange multipliers $\lambda(t) \in \mathfrak{R}^m$ and $\mathbf{v}(t) \in \mathfrak{R}^m$ correspond to the constraints $\mathbf{p}(t) \geq \mathbf{0}$ and $\mathbf{Q}(\mathbf{x})\mathbf{p}(t) = \mathbf{b}$, respectively. Equations (8a) are feasible conditions, (8b) are equilibrium and complementary conditions and $\epsilon > 0$ is the approximate relaxation. A point \mathbf{p} satisfying these constraints is called an ϵ -MAKKT point (or ϵ approximate optimal point), and when $\epsilon = 0$ it's actually the KKT point (or optimal point).

Since in the BOMP model the optimal value $J_2(\mathbf{x})$ of the lower level optimization problem is also restricted, the relation between the optimal value and the approximate optimal value also needs to be considered. The Theorem 3.5 in (Dutta et al. 2013) helps the analysis, that is, at an ϵ -MAKKT point, the objective value $\mathbf{c}^T\mathbf{p}(t)$ is at least ϵ larger than $J_2(\mathbf{x})$. Consequently, the converted single level optimal motion planning problem for (6) can be formulated as follow:

$$\min \quad \beta g(\mathbf{x}(0), \mathbf{x}(t_f), t_f) + (1 - \beta) \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t))dt \quad (9)$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (9a)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \quad (9b)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) \in \mathbf{X}_f \quad (9c)$$

$$\mathbf{c}^T\mathbf{p}(t) \geq \epsilon + \delta \quad (9d)$$

$$\mathbf{Q}(\mathbf{x})\mathbf{p}(t) = \mathbf{b}, \quad \mathbf{p}(t) \geq \mathbf{0}, \quad \lambda(t) \geq \mathbf{0} \quad (9e)$$

$$\|\mathbf{c} - \lambda(t) + \mathbf{Q}^T(\mathbf{x})\mathbf{v}(t)\| \leq \sqrt{\epsilon}, \quad \lambda^T(t)\mathbf{p}(t) \leq \epsilon \quad (9f)$$

The relationship between the MAKKT and the exact KKT optimality conditions is as follow: if a sequence of points $\{\mathbf{p}_k\}$ converge to a point $\bar{\mathbf{p}}$ where the Mangasarian Fromovitz constraint qualification is also satisfied, each point \mathbf{p}_k is an ϵ_k -MAKKT point, and $\epsilon_k \rightarrow 0$ as $k \rightarrow \infty$, then the point $\bar{\mathbf{p}}$ is a KKT point. Therefore, in order to solve problem (6), a decreasing sequence ϵ should be applied to problem (9) and a sequence of approximate optimal trajectory is solved to converge to the exact optimal trajectory.

Combining the above theories, an iterative BOMP algorithm framework is formulated for robot trajectory generation and optimization mission. The algorithm details are shown in Algorithm 1, where the trajectory τ is composed of state trajectory $\mathbf{x}(t)$, control trajectory $\mathbf{u}(t)$, and $\mathbf{p}(t)$, $\lambda(t)$, $\mathbf{v}(t)$ in the J_2 -function. The PSOC method solves problem (9). To limit the length of this paper, the PSOC method details are omitted, and readers can consult the reference paper (Ross and Karpenko 2012). The IPOPT (Wächter and Biegler 2006) algorithm of version 3.12.9 is used to solve the discrete NLP. The IPOPT algorithm is designed with special options, such as convergence tolerance $1e^{-12}$, MA86 linear solver and the acceptable termination conditions are more strict than the desired ones to avoid algorithm early termination. The IPOPT algorithm options are listed in Table 1.

Algorithm 1 The BOMP algorithm

Input: δ , an ϵ sequence $\{0.1, 0.01, 0.001, \dots\}$, an initial trajectory guess τ_0 , and the discrete nodes number K in the PSOC method.

Output: the optimal trajectory τ^* .

- 1: Let $k = 0, K = 15$.
- 2: **loop**
- 3: Using ϵ_k to construct the problem (9).
- 4: Solve problem (9) by the PSOC method with the initial guess τ_k to get a new trajectory τ_{k+1} .
- 5: **if** the optimal value changes small enough **then**
- 6: Let $\tau^* = \tau_{k+1}$ and break.
- 7: Let $k \leftarrow k + 1$.

4 The BOMP model verifications

In this section, the simulations of the BOMP algorithm in autonomous parking problem are shown. The computational and precision benefits of the BOMP model over the area criterion (AC) based model (Li and Shao 2015) and the circle approximation method are demonstrated. And a real experiment on Turtlebot3 is conducted. C++ code is programmed in Linux system, and simulations are conducted on an Intel Core i7-7700K CPU with 8GB RAM that runs at 4.20GHz.

4.1 Simulation in autonomous parking

In this application, four scenarios and three cases in each scenario are designed to reflect the generality, the robustness and the advantages of the BOMP algorithm. Figure 3 illustrates the four scenarios and one case in each scenario. Red vehicles are obstacles, the black vehicle denotes its initial location, and the dashed vehicle denotes the ideal parking. All the four scenarios correspond to irregularly placed obstacles that can be encountered easily in daily life and the differences between cases in each scenario are only the

initial configurations of the vehicle (see Table 2, the abbreviations S, O, C represent the scenario, the obstacle and the case, respectively). For simplicity, each obstacle is the same size as the vehicle and collision avoidance is equivalent to the plane rectangle pair is collision free. The transformation parameters (x, y, θ) of the obstacle coordinate system (it is originated at the blue corner point of the obstacle) relative to the world coordinate system are used to represent each obstacle compactly. These transformation parameters (x, y, θ) are listed in Table 2.

The specified parameters for the parking spot, the vehicle and the Algorithm 1’s input are listed in Table 3, where the vehicle size parameters, mechanical and physical constraints parameters are originated from Li and Shao (2015). As for the initial trajectory guess τ_0 in Algorithm 1, it is specified very easily. Guess the parking completion time arbitrarily, then τ_0 is composed of $x(t) = x_0, y(t) = y_0, \theta(t) = \theta_0, \alpha(t) = 0, v(t) = 0, \omega(t) = 0, p(t) = \mathbf{0}, \lambda(t) = \mathbf{0}, v(t) = \mathbf{0}$ for any t . Under these conditions, the BOMP algorithm solves all these 4×3 problems. The computation results are shown in Table 4 and Figs. 4, 5, 6 and 7.

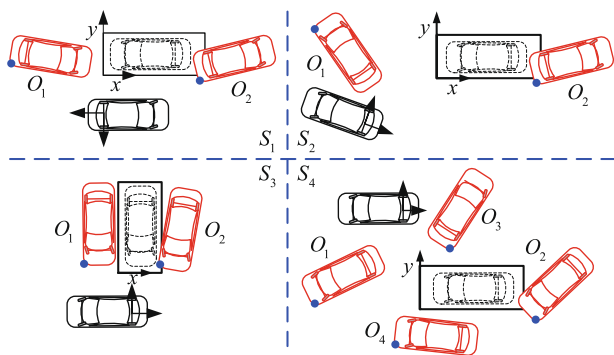


Fig. 3 Four scenarios used to conduct the simulations. Red vehicles are obstacles, the black vehicle denotes the initial location, the dashed vehicles illustrate the ideal parking and each obstacle is the same size as the vehicle. The first two scenarios demonstrate the parallel parking and the third one demonstrates the vertical parking. While the last scenario is not usual in daily life, it’s just designed to show the algorithm’s capability (color figure online)

Table 2 Transformation parameters (x, y, θ) of each obstacle with respect to the world coordinate system and the vehicle initial configuration (x_0, y_0, θ_0) in each case

| | | | | |
|-------|-------|----------------------|-------|-----------------------|
| S_1 | O_1 | $(-5.6, 0.6, -10.0)$ | C_1 | $(-6.0, -2.75, 0.0)$ |
| | | | C_2 | $(0.0, -2.75, 0.0)$ |
| | O_2 | $(5.8, -0.5, 13.0)$ | C_3 | $(0.0, -2.75, 180.0)$ |
| S_2 | O_1 | $(-7.0, 2.8, -55.0)$ | C_1 | $(-6.0, -2.0, 0.0)$ |
| | | | C_2 | $(-8.0, 0.0, -30.0)$ |
| | O_2 | $(5.8, -0.5, 13.0)$ | C_3 | $(-4.0, -3.0, -36.0)$ |
| S_3 | O_1 | $(-2.2, 0.4, 0.0)$ | C_1 | $(-6.0, -2.0, 0.0)$ |
| | | | C_2 | $(1.0, -2.0, 0.0)$ |
| | O_2 | $(2.3, 0.3, -5.0)$ | C_3 | $(-3.0, -2.0, 180.0)$ |
| S_4 | O_1 | $(-6.3, 0.3, 20.0)$ | C_1 | $(-5.0, 7.5, 0.0)$ |
| | O_2 | $(6.8, -1.0, 40.0)$ | C_2 | $(-5.0, 5.0, -10.0)$ |
| | O_3 | $(2.0, 3.6, 60.0)$ | C_3 | $(-1.0, 6.0, 0.0)$ |
| | O_4 | $(-1.5, -2.2, -3.0)$ | | |

The angle unit here is degree, but at other place it is radian

Table 3 The specified parameters for the parking spot, the vehicle and Algorithm 1’s input

| Parameter | Description | Setting |
|-------------------|--|------------|
| SL | Parking spot length | 6.00 m |
| SW | Parking spot width | 2.50 m |
| l | Wheel base length | 2.800 m |
| l_1 | Front overhang length | 0.960 m |
| l_2 | Rear overhang length | 0.929 m |
| W | Vehicle width | 1.942 m |
| v_{\max} | Bound of velocity | 2.00 m/s |
| α_{\max} | Bound of steering angle | 0.714 rad |
| ω_{\max} | Bound of angular velocity | 1.00 rad/s |
| ϵ_p | Bound of terminal position error | 0.10 m |
| ϵ_θ | Bound of terminal angle error | 0.17 rad |
| δ | The safety pseudodistance of the J_2 -function | 0.05 |

Table 4 The computation results of the BOMP and the AC algorithms, where T is the computation time and v^* is the optimal value

| | BOMP | | AC | |
|-------|---------|-------|---------|-------|
| | T (s) | v^* | T (s) | v^* |
| S_1 | | | | |
| C_1 | 3.15 | 15.71 | 149.04 | 16.98 |
| C_2 | 2.36 | 12.46 | 154.84 | 14.64 |
| C_3 | 3.81 | 17.05 | 141.62 | 17.58 |
| S_2 | | | | |
| C_1 | 4.24 | 15.33 | 144.96 | 15.26 |
| C_2 | 4.10 | 19.02 | × | × |
| C_3 | 3.47 | 15.84 | 127.66 | 14.88 |
| S_3 | | | | |
| C_1 | 4.71 | 16.57 | × | × |
| C_2 | 4.25 | 13.84 | 203.64 | 13.57 |
| C_3 | 4.00 | 17.73 | 208.51 | 18.46 |
| S_4 | | | | |
| C_1 | 7.01 | 16.40 | × | × |
| C_2 | 7.20 | 14.54 | × | × |
| C_3 | 8.60 | 15.84 | × | × |

The symbol × denotes that the AC algorithm fails to solve the corresponding problem

Then the computation complexity comparison between the BOMP algorithm and the AC algorithm is made. Given a point b and four corner points A, B, C, D of a rectangle in the plane, the AC collision avoidance constraint used in this paper is:

$$\frac{S_{\triangle bAB} + S_{\triangle bBC} + S_{\triangle bCD} + S_{\triangle bDA}}{S_{\square ABCD}} \geq 1.025 \tag{10}$$

where S_{\triangle} and S_{\square} denote the area of a triangle and a rectangle, respectively. And there are eight nonlinear constraints

(10) for a pair of plane rectangles collision avoidance problem. Then replacing (6d) by this constraint, the AC based motion planning problem can be obtained. Since in Li and Shao (2015), the IPOPT options and the computation time were not shown, the algorithm options in Table 1 are used. And the initial guess τ_0 is selected the same way $x(t) = x_0, y(t) = y_0, \theta(t) = \theta_0, \alpha(t) = 0, v(t) = 0, \omega(t) = 0$. However, the AC algorithm does not solve the whole autonomous parking problems. The results are shown in Table 4.

From Table 4, we can see that the shortest and the longest time of the BOMP algorithm to find an optimal collision-free trajectory for the 12 problems are 2.36 and 8.60 s, respectively. While for the AC algorithm in the solved problems, they are 127.66 and 208.51 s. The reasons why the BOMP algorithm has significant superiorities in computation speed than the AC algorithm may be:

- J_2 -function collision avoidance is linear programming, while the AC based collision avoidance method needs to calculate the distance between points, then the area of a triangle, such that it has a high degree of non-linearity. Linear programming is the most well-studied problem, and finding the solution of it is intrinsically much faster than the problem with nonlinear constraints.
- Variables in the J_2 -function make the BOMP algorithm only concern the closest components implicitly. However, the AC method needs to consider the eight points between a pair of rectangles simultaneously and explicitly, whose optimization direction is difficult to be determined.
- The BOMP algorithm finds a sequence of solution progressively approaching the optimal solution such that it significantly reduces the difficulty to solve the problem, while the AC method uses a global step to find the solution which tends to break down in complicated problems.

Limited by the length of this paper, the optimized vehicle motion and the optimized trajectories of state and control variables by the BOMP algorithm are just partly plotted in Figs. 4, 5, 6 and 7. The results demonstrate that the BOMP algorithm is capable of handling parking motion planning problems successfully and efficiently. Besides, the BOMP algorithm can autonomously park a vehicle in a much more complicated scenario that needs several maneuvers like a skillful driver. Meanwhile, the terminal orientation deviation and the position deviation are considered, so the vehicle stops as parallel and close as to parking spot central axis. It is worth emphasizing that all these 4×3 simulations are conducted without adjustment of any algorithm options, like ϵ , and this shows the robustness, generality, and unification of the BOMP model and the BOMP algorithm. However,

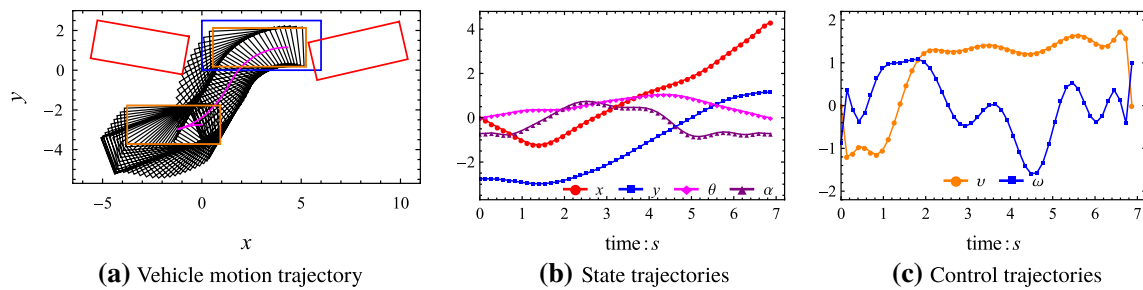


Fig. 4 Optimization results for scenario 1 case 2 by the BOMP algorithm. In **a**, the orange rectangles represent the vehicle initial and final locations, and the magenta curve shows the motion trajectory of vehicle rear wheel axis mid-point. The state trajectories in **b** are very

smooth, while the control trajectories in **c** are oscillating. The state and control trajectories in all other problems have the same characteristics (color figure online)

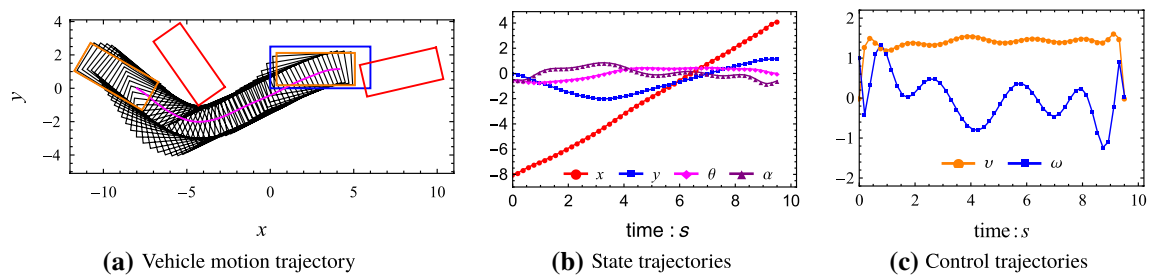


Fig. 5 Optimization results for scenario 2 case 2 by the BOMP algorithm

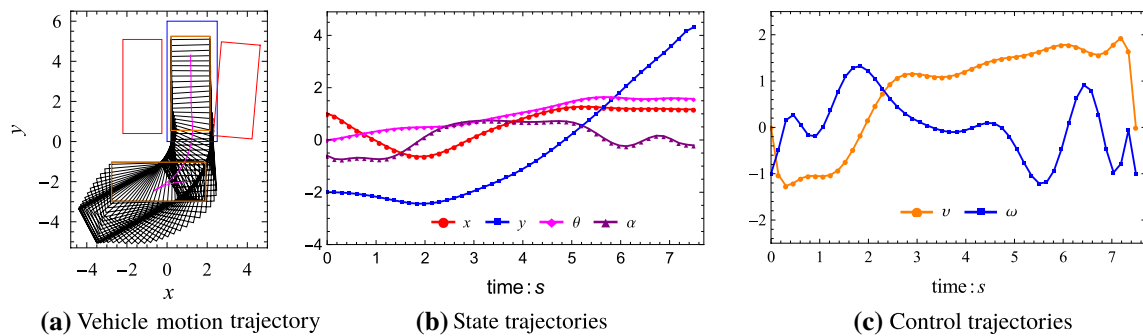


Fig. 6 Optimization results for scenario 3 case 2 by the BOMP algorithm

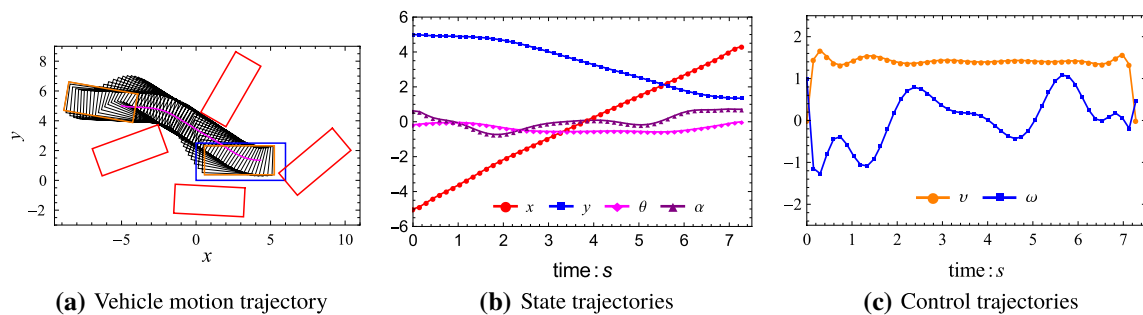


Fig. 7 Optimization results for scenario 4 case 2 by the BOMP algorithm

because the vehicle kinematics, mechanical and physical constraints in our application is similar to that of Li and Shao (2015), the control variables smoothness issue and terminal steering angle issue also exist in the BOMP algorithm, readers may consult that paper for the reasons. These issues are inherent in the modeling process and need some unique but not complicated techniques to overcome. Nevertheless, this paper focuses on the BOMP model and the solution to it, so these issues are left to resolve.

4.2 Polygon VS circle approximation

One motion planning problem in scenario 3 of the previous section is used to illustrate the different impacts of polygon and circle approximation methods. The planning task is to let the vehicle move from the initial configuration $(1.0, -2.0, 0.0)$ to the final configuration $(1.0, 10.0, \pi/2.0)$. Figure 8a shows the result of the BOMP algorithm, and the vehicle and obstacles are still represented as rectangles. Figure 8b shows the result of circle approximation method. The vehicle and the obstacles are approximated by three overlapping circles of radius 1.3 m (the red circles), respectively. The results demonstrate that the circle approximation makes the object expanding, and the vehicle can not pass through narrow passages. To approximate the object exactly, there should be many more different radius circles, and this will increase the computation time dramatically. The drawbacks of circle approximation will become increasingly severe if the ratio of rectangle width and length reduces further. Because of the high approximation precision and low overlapping rate of the polyhedron or polygon approximation method and the simplicity of the J_2 -function, the BOMP model will compensate motion planning methods in many domains.

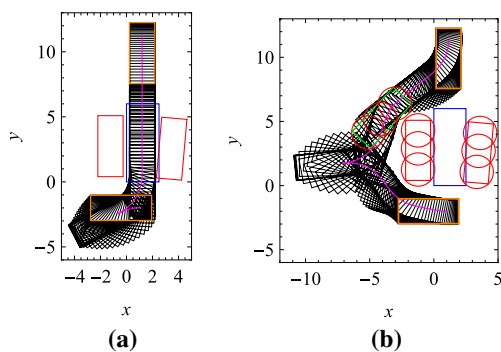


Fig. 8 The vehicle motion obtained by the BOMP algorithm and the circle approximation method, respectively. The green rectangle in **b** represents the vehicle at an instant moment, and the three overlapping circles represent the approximation of the vehicle and the obstacles (color figure online)

4.3 Physical experiment

We performed a physical experiment involving a mobile robot (TurtleBot3) to verify the correctness and effectiveness of the BOMP model (see in Fig. 9). To simplify the experiment, we only consider the robot kinematics, and the robot moved at very low speed. Using SLAM technique, the environment geometry can be obtained and then through convex decomposition (Mamou and Ghorbel 2009) the polygons information are available. Then given an initial robot location, the global control commands (linear and angular speed) are calculated by the BOMP algorithm. In this experiment, the time-optimal trajectory is sought, and the global control variables trajectories corresponding to Fig. 9 are shown in Fig. 10. When the robot is executing the control commands, the dynamic uncertainty and drift will accumulate the driving error. When this error exceeds tolerance, replanning process is triggered. The details of robot odometry, localization, control framework, error accumulation calculation, and replanning cycles will be presented in our future works. The videos of this experiment are available at <http://www.hust.edu.cn>.



Fig. 9 The environment setup for this experiment. There are seven obstacles in a very small area ($1.5 \text{ m} \times 1.5 \text{ m}$) and the goal of all the experiments is to let Turtlebot3 stop at the center of the white square which is $0.5 \text{ m} \times 0.5 \text{ m}$

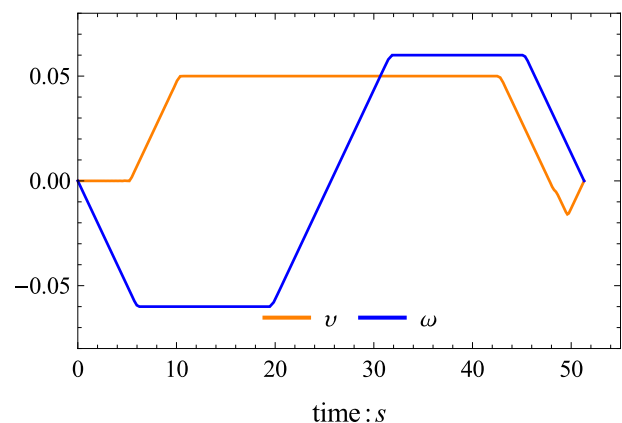


Fig. 10 Given robot initial location, the calculated global control variables trajectories by the BOMP algorithm

5 Conclusion and prospects

In this paper, we have proposed a general and unified BOMP model for robot trajectory generation and optimization in obstacles environment, which makes the optimal control method a practical approach for complex high-precision robot motion planning problems. The upper-level optimal control is designed for robot nonlinear dynamics, while the lower-level J_2 -function is for geometry collision-free constraint. Simulations in complex autonomous parking scenarios and experiment on Turtlebot3 demonstrate the computation superiorities and efficiency of the BOMP model. The highlights of this paper lie in the following aspects.

1. The BOMP model utilizes the J_2 -function linear programming to avoid collision. Because of the simplicity of the J_2 -function, the BOMP model can solve robot motion planning problems in very high speed and high precision.
2. The BOMP model is an open framework that can handle any user-specified constraints. And the BOMP model is dimensionality independent. Therefore, this model can be utilized in manipulators, vehicles and humanoid robots, etc.
3. The BOMP algorithm makes full use of the convergence property of the MAKKT theory which makes the BOMP algorithm very fast.

However, many aspects need to be studied deeply and comprehensively. First, one main issue is the computation efficiency for robots moving in cluttered and obstacles intensive environment. One possible solution is to combine the previous research theory, like sampling based methods. While in real applications, some simple techniques can be used. Take the parking problem as an example (Li et al. 2016), make finite classifications of the parking cases and calculate the trajectory offline to construct a database. Then in real parking process, pick out a closest recorded solution to the current situation to initialize the algorithm and finally solve the problem online. Second, motion planning for robots in dynamical environment with moving obstacles are regarded as the most difficult, important and significant research field. Whereas the BOMP model only solves the static problem at present, there are still extensive works to adopt the obstacles movement prediction and estimation theory into the BOMP framework. Third, with appropriate objectives and constraints in different levels, the multilevel problems have a close connection to multiple objectives optimization problems. However, solving the multilevel problem is much more difficult than the bilevel problem, and the related theories have not been proposed. All these three unresolved issues are challenging and meaningful for the development of intelligent robots and will be discussed in our future works. Moreover,

despite the simulation results and simple experiment on Turtlebot3 in this paper, extensive work is needed to conduct experiments on real complex applications and to verify the proposed algorithm.

Acknowledgements This work was supported by the National Science Foundation of China under Grant 51327801.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix

The J_2 -function represents one collision checking method between convex polyhedrons in any dimensional spaces including the convex polygons in 2D environment. Suppose obstacle A and robot B are approximated by convex polyhedrons with n_a vertexes and n_b vertexes, respectively. The vertexes coordinates are $\{a_i\}$, $i = 1, 2, \dots, n_a$ and $\{b_j\}$, $j = 1, 2, \dots, n_b$. Where $a_i, b_j \in \mathcal{R}^m$ and m is the dimensionality. Then the J_2 -function computes the minimum pseudo-distance between A and B , and $J_2 > 0$ denotes collision free:

$$J_2(A, B) = \min \sum_{k=1}^{m+2} z_k \quad (11)$$

$$\text{s.t. } Ar - Bs + z = \mathbf{0}, \quad (11a)$$

$$\sum_{i=1}^{n_a} r_i + z_{m+1} = 1, \quad \sum_{j=1}^{n_b} s_j + z_{m+2} = 1, \quad (11b)$$

$$r \geq \mathbf{0}, s \geq \mathbf{0}, z \geq \mathbf{0}, z_{m+1} \geq 0, z_{m+2} \geq 0. \quad (11c)$$

where $r = [r_1 \ r_2 \ \dots \ r_{n_a}]^T$, $s = [s_1 \ s_2 \ \dots \ s_{n_b}]^T$, and $z = [z_1 \ z_2 \ \dots \ z_m]^T$. In constraint (11a), the symbols A and B are actually matrixes composed of the points coordinates, i.e., $A = [a_1 \ a_2 \ \dots \ a_{n_a}]$ and $B = [b_1 \ b_2 \ \dots \ b_{n_b}]$. And the vector inequality is to be understood element-wise.

The J_2 -function (11) can be wrote as a standard linear programming problem:

$$J_2(A, B) = \min \ c^T p \quad (12)$$

$$\text{s.t. } Qp = b, \quad p \geq \mathbf{0}.$$

where the optimization variable is $p = [r^T \ s^T \ z^T \ z_{m+1} \ z_{m+2}]^T$, and the notations Q, b, c can be constructed appropriately. $p, c \in \mathcal{R}^{n'}$, $Q \in \mathcal{R}^{m' \times n'}$, $b \in \mathcal{R}^{m'}$ where $n' = n_a + n_b + m + 2$ and $m' = m + 2$.

Suppose obstacle A is static and robot B is moving, then the matrix A in (11a) is constant, and B depends on robot state x . Therefore, in robot trajectory optimization problem we can construct the collision avoidance constraint $J_2(x) \geq \delta$, the matrix Q depends on x and p is a time-dependent variable.

References

- Albrecht, S., Leibold, M., Ulbrich, M.: A bilevel optimization approach to obtain optimal cost functions for human arm movements. *Numer. Algebra Control Optim.* **2**(1), 105–127 (2012)
- Benita, F., Mehltitz, P.: Bilevel optimal control with final-state-dependent finite-dimensional lower level. *SIAM J. Optim.* **26**(1), 718–752 (2016)
- Betts, J.T.: *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, vol. 19. Siam, Al Burhama (2010)
- Bryson, A.E.: *Applied Optimal Control: Optimization, Estimation and Control*. Routledge, Abingdon (2018)
- Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J.: Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.* **29**(5), 485–501 (2010)
- Du, X., Tan, K.K.: Autonomous reverse parking system based on robust path generation and improved sliding mode control. *IEEE Trans. Intell. Transp. Syst.* **16**(3), 1225–1237 (2015)
- Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**(3), 497–516 (1957)
- Dutta, J., Deb, K., Tulshyan, R., Arora, R.: Approximate kkt points and a proximity measure for termination. *J. Glob. Optim.* **56**(4), 1463–1499 (2013)
- González, D., Pérez, J., Milanés, V., Nashashibi, F.: A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **17**(4), 1135–1145 (2016)
- Haddad, M., Khalil, W., Lehtihet, H.: Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint. *IEEE Trans. Robot.* **26**(5), 954–962 (2010)
- Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. *SIAM J. Sci. Comput.* **13**(5), 1194–1217 (1992)
- Kant, K., Zucker, S.W.: Toward efficient trajectory planning: the path-velocity decomposition. *Int. J. Robot. Res.* **5**(3), 72–89 (1986)
- Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **5**(1), 90–98 (1986)
- LaValle, S.M.: *Planning Algorithms*. Cambridge University Press, Cambridge (2006)
- Li, B., Shao, Z.: A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles. *Knowl. Based Syst.* **86**, 11–20 (2015)
- Li, B., Wang, K., Shao, Z.: Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. *IEEE Trans. Intell. Transp. Syst.* **17**(11), 3263–3274 (2016)
- Likhachev, M., Ferguson, D.: Planning long dynamically feasible maneuvers for autonomous vehicles. *Int. J. Robot. Res.* **28**(8), 933–945 (2009)
- Liu, W., Li, Z., Li, L., Wang, F.Y.: Parking like a human: a direct trajectory planning solution. *IEEE Transactions on Intelligent Transportation Systems* **18**(12), 3388–3397 (2017)
- Mamou, K., Ghorbel, F.: A simple and efficient approach for 3d mesh approximate convex decomposition. In: *Proceedings of IEEE International Conference Image Processing (ICIP)*, IEEE, pp. 3501–3504 (2009)
- Muller, B., Deutscher, J., Grodde, S.: Continuous curvature trajectory design and feedforward control for parking a car. *IEEE Trans. Control Syst. Technol.* **15**(3), 541–553 (2007)
- Nilsson, J., Brännström, M., Coelingh, E., Fredriksson, J.: Lane change maneuvers for automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **18**(5), 1087–1096 (2016)
- Oetiker, M.B., Baker, G.P., Guzzella, L.: A navigation-field-based semi-autonomous nonholonomic vehicle-parking assistant. *IEEE Trans. Veh. Technol.* **58**(3), 1106–1118 (2009)
- Rasekhipour, Y., Khajepour, A., Chen, S.K., Litkouhi, B.: A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Trans. Intell. Transp. Syst.* **18**(5), 1255–1267 (2016)
- Reeds, J., Shepp, L.: Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **145**(2), 367–393 (1990)
- Robinson, D.R., Mar, R.T., Estabridis, K., Hewer, G.: An efficient algorithm for optimal trajectory generation for heterogeneous multi-agent systems in non-convex environments. *IEEE Robot. Autom. Lett.* **3**(2), 1215–1222 (2018)
- Ross, I.M., Karpenko, M.: A review of pseudospectral optimal control: from theory to flight. *Annu. Rev. Control* **36**(2), 182–197 (2012)
- Sinha, A., Malo, P., Deb, K.: Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. *Eur. J. Oper. Res.* **257**(2), 395–411 (2017)
- Sinha, A., Malo, P., Deb, K.: A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Trans. Evolut. Comput.* **22**(2), 276–295 (2018)
- Tazaki, Y., Okuda, H., Suzuki, T.: Parking trajectory planning using multiresolution state roadmaps. *IEEE Trans. Intell. Veh.* **2**(4), 298–307 (2017)
- Upadhyay, S., Ratnoo, A.: A point-to-ray framework for generating smooth parallel parking maneuvers. *IEEE Robot. Autom. Lett.* **3**(2), 1268–1275 (2018)
- Vicente, L., Savard, G., Júdice, J.: Descent approaches for quadratic bilevel programming. *J. Optim. Theory Appl.* **81**(2), 379–399 (1994)
- Vorobieva, H., Glaser, S., Minoiu-Enache, N., Mammari, S.: Automatic parallel parking in tiny spots: path planning and control. *IEEE Trans. Intell. Transp. Syst.* **16**(1), 396–410 (2015)
- Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program* **106**(1), 25–57 (2006)
- Wang, H., Chen, Y., Souères, P.: A geometric algorithm to compute time-optimal trajectories for a bidirectional steered robot. *IEEE Trans. Robot.* **25**(2), 399–413 (2009)
- Wright, S., Nocedal, J.: *Numerical optimization*, vol. 35(67–68), p. 7. Springer Science, New York (1999)
- Xiong, Y.: Generalized criterion for minimum zone and computer arbitration. *Acta Metrol. Sin.* **8**(2), 1–8 (1987)
- Xiong, Y., Ding, H.: General criterion and control strategy of collision-free movement for manipulators. *Int. J. Robot. Autom.* **4**(2), 75–80 (1989)
- Xu, D., Shi, Y., Ji, Z.: Model-free adaptive discrete-time integral sliding-mode-constrained-control for autonomous 4wv parking systems. *IEEE Trans. Ind. Electron.* **65**(1), 834–843 (2018)
- Ye, Z., Li, K., Stapelbroek, M., Savelsberg, R., Günther, M., Pischinger, S.: Variable step-size discrete dynamic programming for vehicle speed trajectory optimization. *IEEE Trans. Intell. Transp. Syst.* **99**, 1–9 (2018)
- Zips, P., Böck, M., Kugi, A.: Optimisation based path planning for car parking in narrow environments. *Robot. Auton. Syst.* **79**, 1–11 (2016)



Shenglei Shi received a B.S. degree in mechanical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2016. He is currently pursuing a Ph.D. degree at HUST. His current research interests include motion planning and trajectory optimization in robotics.



Youlun Xiong graduated from the Department of Mechanical Engineering and Postgraduate School at Xiāfan Jiatong University, Xiāfan, China, in 1962 and 1966, respectively. From 1966 to 1980 he was employed by the Department of Mechanical Engineering at Huazhong University of Science and Technology (HUST), Wuhan, China. From 1980 to 1982, he was a visiting scholar in the Department of Control Engineering at Sheffield University, Sheffield, UK. In 1982, he returned to the Department of

Mechanical Engineering at HUST, where he is a Professor. From 1988 to 1989, he was a visiting professor in the Department of Aeronautical and Mechanical Engineering at the University of Salford, Salford, UK. In 1995 he was elected as Academician of Chinese Academy of Sciences. His research interests include robotics, precision measurement, intelligence manufacturing and manufacturing automation.



Jiankui Chen received the B.S. degree in vehicle-operation engineering from the Wuhan University of Technology, Wuhan, China, in 2001, and the M.S. and the Ph.D. degrees in mechatronic engineering from Huazhong University of Science and Technology (HUST), Wuhan, in 2006 and 2010, respectively. He was a Post-Doctoral Fellow with the Department of Control Science and Engineering, HUST, in 2011. He is currently a Lecturer with the State Key Laboratory of Digital Manufacturing Equipment and Technology, HUST. His current research interests include flexible electronics manufacturing and roll-to-roll Web transport system control.



Caihua Xiong received the Ph.D. degree in mechanical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1998. From 1999 to 2003, he was with City University of Hong Kong and The Chinese University of Hong Kong as a Postdoctoral Fellow, and with Worcester Polytechnic Institute, Worcester, MA, USA, as a Research Scientist. He is a Chang Jiang Professor with HUST. He has received the National Science Fund for Distinguished Young Scholars of China. His research interests include biomechatronic prostheses, rehabilitation robotics, and robot motion planning and control.