

# A suboptimal and analytical solution to mobile robot trajectory generation amidst moving obstacles

Jun Peng · Wenhao Luo · Weirong Liu ·  
Wentao Yu · Jing Wang

Received: 11 May 2013 / Accepted: 3 January 2015 / Published online: 4 February 2015  
© Springer Science+Business Media New York 2015

**Abstract** In this paper, we present a suboptimal and analytical solution to the trajectory generation of mobile robots operating in a dynamic environment with moving obstacles. The proposed solution explicitly addresses both the robot kinodynamic constraints and the geometric constraints due to obstacles while ensuring the suboptimal performance to a combined performance metric. In particular, the proposed design is based on a family of parameterized trajectories, which provides a unified way to embed the kinodynamic constraints, geometric constraints, and performance index into a set of parameterized constraint equations. To that end, the suboptimal solution to the constrained optimization problem can be analytically obtained. The solvability conditions to the constraint equations are explicitly established, and the proposed solution enhances the methodologies of real-time path

planning for mobile robots with kinodynamic constraints. Both the simulation and experiment results verify the effectiveness of the proposed method.

**Keywords** Nonholonomic mobile robots · Trajectory generation · Moving obstacles · Kinodynamic constraints · Optimization

## 1 Introduction

During the past decades, trajectory generation for mobile robots moving in a dynamic environment has received considerable attention. Two major concerns towards that topic are problems of feasibility and optimality (LaValle 2006). The feasibility is mainly concerned with the kinematic constraints of robots such as nonholonomic constraints, or kinodynamic constraints such as velocity and acceleration bounds (Li and Canny 1992). It is desirable for the robot to generate a feasible trajectory on-line that also satisfies certain optimality requirements.

To consider robot kinematic constraints, some canonical methods as Kim and Khosla (1992), Borenstein and Koren (1991) that directly focus on geometric space have been revised by alternative techniques. One approach is to mainly study on the path planner towards nonholonomic constraints and try to obtain the steering method to drive a robot to a pre-determined configuration. Without obstacles, discretizing control method is proposed in Barraquand and Latombe (1991), Divelbiss and Wen (1997) to integrate the motion equations to obtain a feasible path. A reverse motion based case has been adapted to obstacle-cluttered environment to optimize the path length (Bicchi et al. 1995). In Duleba and Sasiadek (2003) Newton algorithm and Jacobian matrix are used to solve energy optimization without obstacles. More-

---

Jun Peng (first author) and Wenhao Luo (second author) contributed equally to this work.

---

J. Peng · W. Liu (✉) · W. Yu  
School of Information Science and Engineering, Central South University and Hunan Engineering Laboratory for Advanced Control and Intelligent Automation, Changsha 410083, China  
e-mail: weirong\_liu@126.com; frat@csu.edu.cn

J. Peng  
e-mail: pengj@csu.edu.cn

W. Yu  
e-mail: wentaoyu@gmail.com

W. Luo  
Robotics Institute, School of Computer Science,  
Carnegie Mellon University, Pittsburgh, PA 15213, USA  
e-mail: luo@cmu.edu; whluo12@gmail.com

J. Wang  
Department of Electrical and Computer Engineering,  
Bradley University, Peoria, IL 61525, USA  
e-mail: jingwang@bradley.edu

over, the planning problem can be directly considered as an optimal control problem using Hamiltonian equations and Pontriagin's maximum principle in [Reister and Pin \(1994\)](#), [Balkcom and Mason \(2002\)](#). Input parameterizations are used in [Murray and Sastry \(1993\)](#), [Tilbury and Murray \(1995\)](#) to design steering input and represent the trajectories by sinusoidal, polynomial, or piecewise constant functions to smooth control inputs. Alternative methods ([Dong and Guo 2005](#); [Fliess et al. 1995](#)) generate smooth trajectories based on differential flatness approach. However, moving obstacles are seldom considered in those works, and optimality could be hard to formulate with kinematic constraints ([LaValle 2006](#)). Recent work in [Bhattacharya et al. \(2012\)](#) proposes a graph-search based method to plan homologous trajectories with topological constraints in Euclidean configuration spaces.

Approaches that directly plan nonholonomic trajectories becomes a new topic. This kind of solutions concentrates on trajectory generation and corresponding steering control based on the property of nonholonomic systems. Probabilistic roadmap methods (PRM) ([Kindel et al. 2000](#); [Ladd and Kavraki 2004](#); [van den Berg and Overmars 2007](#)) and rapidly-exploring random trees (RRTs) methods ([Cheng et al. 2008](#); [ucan and Kavraki 2012](#); [Karaman and Frazzoli 2010](#); [LaValle and Kuffner 1999](#)) are introduced to search for feasible paths and incorporate both vehicle's kinematic and dynamic constraints. In these works obstacle-free space is divided in terms of random samples and connection of these samples form the corresponding feasible trajectory. Grid-based discretization method is also used in [ucan and Kavraki \(2012\)](#) to estimate the coverage of the state space and could help to detect less-explored areas with complex dynamics. Most of such algorithms mainly discuss about the robot moving in static environment, and plenty of moving obstacles could make it hard for random trees to converge to a certain optimal path.

Recently, another kind of search-based method using "lattice graphs" has been proven to be efficient and attractive working in complex dynamic environments. The concept is first developed in [Pivtoraiko et al. \(2009\)](#), wherein the idea is to divide the configuration space into a set of cells (reachable states) and construct a state lattice at first for graph searches using heuristic search algorithms such as A\* and D\* Lite ([Koenig and Likhachev 2002](#)), and subsequently search for a specific combination of actions/edges (control/motion primitives) sequences in the action space to drive the robot along the feasible path. In order to address real-time planning for high-speed robot moving over large distances in dynamic environment, [Likhachev and Ferguson \(2009\)](#) introduced a novel multi-resolution lattice state/action space and employ the incremental search algorithms such as Anytime Dynamic A\* (AD\*) to more efficiently find global and feasible suboptimal solutions among obstacles. However, the resulting purely spatial trajectories by [Likhachev and Ferguson \(2009\)](#) may

fail to deal well when operating in cluttered environments full of dynamic obstacles, since it does not consider time as a state variable. Considering such a problem, [Kushleyev and Likhachev \(2009\)](#) proposes a novel data structure, "a time-bounded lattice", to merge both short-term planning with concerns of time and long-term planning without time. Leveraging the time horizon from the time-bounded lattice idea ([Kushleyev and Likhachev 2009](#); [Narayanan 2012](#)) proposes an anytime planners to find an initial solution quickly and then continue to optimize it as time allows. In those methods, however, as pointed out in [Likhachev and Ferguson \(2009\)](#), [Howard \(2007\)](#) no explicit representation of curvature and the rigidity of the discretized motion-primitives may result in discontinuous motions at the junctions between motion primitives and locally less optimal performance for the rendered paths.

Compared to the search-based methods, the real-time approaches of analytical motion planning are in a way more computational simple and efficient for working in local dynamic environment. A new parametric solution has been proposed in [Qu et al. \(2004\)](#) to analytically consider both kinematic constraints and moving obstacles with the utilization of the chained form/differential flatness ([Lvine 2009](#); [Luca et al. 1998](#)), and the *velocity obstacles* method in [Fiorini and Shiller \(1993\)](#), [Shiller et al. \(2001\)](#). The concept of 'Velocity Obstacle' is extended in [Wu and How \(2012\)](#) to incorporate moving obstacles that have constrained dynamics but move unpredictably. Based on [Qu et al. \(2004\)](#), extended works in [Yang et al. \(2005, 2010\)](#) have introduced energy-optimal and path length-optimal methods. For those methods, however, lack of considering robot kinodynamic constraints and special need for intermediate configuration to overcome the vertical singularity could weaken both feasibility and optimal effectiveness of the generated paths. In [Guo and Tang \(2008\)](#), [Yuan and Shim \(2011\)](#), [Guo et al. \(2007\)](#), [Hashim and Tien-Fu \(2009\)](#) different trajectory models are used to avoid curvature discontinuities and vertical singularities, while the optimization is not completely solved in [Guo and Tang \(2008\)](#). Parameters of the robot dynamic model in [Yuan and Shim \(2011\)](#) is often difficult to obtain. [Liu and Sun \(2011\)](#) provides a special cost function to present optimal options whereas it can only deal with statistic obstacles with prior knowledge of environment. Further research on higher dimensional states such as velocity and acceleration (kinodynamic) and complete optimization are needed to better extend application of analytical solutions.

In this paper, we propose a suboptimal and analytical solution for mobile robot on-line trajectory generation in the presence of moving obstacles. First, we parameterize the family of trajectories by two time-variant polynomials and based on the flatness differential property as well as an explicit consideration of robot kinematic constraint, high-dimensional robot boundary states are substituted to make

trajectories controlled by two freely adjustable parameters. As a result, importantly, the kinodynamic constraints can be handled together with geometric constraints due to obstacles, and the trajectory generation problem can then be recast as optimizing the two free parameters based on a set of constraint equations. By defining two optimal indices to quantify the minimum control energy consumption and the shortest path length, respectively, the suboptimal solution in terms of the two parameters can be obtained in closed form, allowing highly efficient on-line computation. Particularly, the optimization process is simplified and analytically solved in the proposed 2D parameter space, leading to improved suboptimal solutions that render good suboptimal and feasible trajectories in real-time.

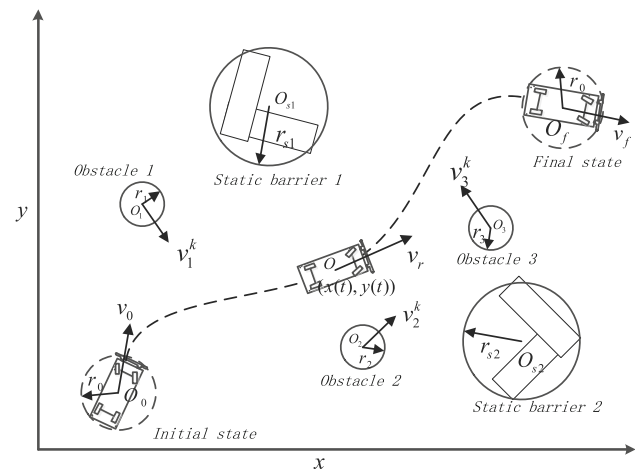
This paper further extends the work in [Qu et al. \(2004\)](#), [Guo and Tang \(2008\)](#), and [Guo et al. \(2007\)](#). Compared with those existing results, the main contributions of this paper are as follows. (1) Higher-dimensional states such as velocity and acceleration are considered for the robot, allowing to explicitly address the kinodynamic constraints, and to avoid singularity problem as well as discontinuous or incompatible control at boundary states. (2) Optimization problem is flexibly and analytically addressed on a 2D parameter space by considering an adjustable combined metric consisting of both energy optimality and length optimality, making the obtained 2D analytical suboptimal solutions in principle outweigh the 1D solutions in the existing works and still run in real-time. (3) Solvable conditions are established with the discussion of remedies for the possible unsolvable constraint equations.

This paper is organized as follows. Section 2 formulates the trajectory generation problem. In Sect. 3, two optimal performance indices of the trajectories are introduced and integrated with a combined solution form without addressing constraints. In Sect. 4, kinodynamic constraints and the collision avoidance are taken into account. Framework of the suboptimal solutions on the parameter space are clearly illustrated. Simulation and experiment results are included in Sect. 5 to illustrate the effectiveness of the proposed method. Section 6 concludes the paper.

## 2 Problem formulation

Consider the trajectory generation problem for a car-like mobile robot. The mobile robot is represented by a circum-circle and conforms to nonholonomic constraints. The front wheels of the mobile robot are steering wheels and the rear wheels are driving wheels with a fixed forward orientation, and the mathematical model of the robot is described by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \varphi / l \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (1)$$



**Fig. 1** The mobile robot moving in the dynamic environment

where  $(x, y)$  represents the Cartesian coordinates of the middle point of the rear wheel axle,  $\theta$  is the orientation of the robot body with respect to the  $X$ -axis,  $\varphi$  is the steering angle,  $l$  is the distance between the front and rear wheel-axle centers,  $u_1$  the linear velocity of the driving wheels and  $u_2$  the steering velocity of the guiding wheels. Note that  $\varphi \in (-\pi/2, \pi/2)$  due to the structure constraint of the robot.

Assume that the robot moves in a working region as shown in Fig. 1, where moving obstacles are represented by simple circumcircles.

In this paper, the objective is to design a trajectory generation method to steer the robot moving from the starting point  $O_0(x_0, y_0)$  at the initial time  $t_0$  to the final point  $O_f(x_f, y_f)$  at the final time  $t_f = t_0 + T$  while satisfying kinodynamic constraints and certain optimality requirements, where  $T$  is a given constant.

## 3 An optimal and analytical solution framework

### 3.1 Trajectory parameterization

The proposed trajectory generation solution is based on the typical trajectory parameterization method, as studied in [Qu et al. \(2004\)](#), [Guo et al. \(2007\)](#). Specifically, we parameterize the family of trajectories of the robot as two polynomials in terms of time  $t$

$$\begin{aligned} x(t) &= [c_0 \ c_1 \ c_2 \ \dots \ c_p] f(t) \\ y(t) &= [d_0 \ d_1 \ d_2 \ \dots \ d_p] f(t) \end{aligned} \quad (2)$$

with  $f(t) = [1 \ t \ t^2 \ t^3 \ \dots \ t^p]^T$ , integer  $p > 0$  and  $c_i, d_i, i = 0, \dots, p$  are some constants to be determined to address the system model constraints, collision avoidance and optimal performance in a unifying way.

The polynomial order  $p$  plays an important role in solving the posed optimal trajectory generation problem. In essence,

$p = 5$  suffices to determine a unique solution for the parameterized trajectory defined in (2) based on the available trajectory boundary conditions at  $t_0$  and  $t_f$ . In this paper  $p$  is specified by 6, which then releases two more free parameters  $c_6$  and  $d_6$  so as to deal with additional constraints from collision avoidance and optimization.

Assume that the initial driving velocity  $v_0 \triangleq u_1(t_0)$ , the initial acceleration  $a_0 \triangleq \dot{u}_1(t_0)$ , the final driving velocity  $v_f \triangleq u_1(t_f)$ , and the final acceleration  $a_f \triangleq \dot{u}_1(t_f)$  are known. Thus, together with the initial and final state boundary conditions  $q_0 = [x_0, y_0, \theta_0, \varphi_0]^T$  and  $q_f = [x_f, y_f, \theta_f, \varphi_f]^T$ , we can construct a higher-dimensional boundary conditions such as  $q_0^* = [x_0, y_0, \theta_0, \varphi_0, v_0, a_0]^T$  and  $q_f^* = [x_f, y_f, \theta_f, \varphi_f, v_f, a_f]^T$ . Considering the kinematic model of the mobile robot, then we have

$$\begin{aligned} \frac{dx}{dt} \Big|_{t_0} &= v_0 \cos \theta_0, & \frac{dy}{dt} \Big|_{t_0} &= v_0 \sin \theta_0 \\ \frac{d^2x}{dt^2} \Big|_{t_0} &= a_0 \cos \theta_0 - \frac{v_0^2 \tan \varphi_0}{l} \sin \theta_0 \\ \frac{d^2y}{dt^2} \Big|_{t_0} &= a_0 \sin \theta_0 + \frac{v_0^2 \tan \varphi_0}{l} \cos \theta_0 \\ \frac{dx}{dt} \Big|_{t_f} &= v_f \cos \theta_f, & \frac{dy}{dt} \Big|_{t_f} &= v_f \sin \theta_f \\ \frac{d^2x}{dt^2} \Big|_{t_f} &= a_f \cos \theta_f - \frac{v_f^2 \tan \varphi_f}{l} \sin \theta_f \\ \frac{d^2y}{dt^2} \Big|_{t_f} &= a_f \sin \theta_f + \frac{v_f^2 \tan \varphi_f}{l} \cos \theta_f \end{aligned} \quad (3)$$

To further address the dynamically changing environment, we assume the trajectory (2) is piecewise parameterized and it is updated at each sampling instant. The entire maneuver time is  $T = t_f - t_0$  and sampling time-point (trajectory refinement time-point) is chosen to be at  $t_0, t_0 + T_s, t_0 + 2T_s, \dots, t_0 + kT_s, \dots, t_0 + (\bar{k} - 1)T_s$ , where  $k = 0, 1, \dots, \bar{k} - 1$  and  $T_s$  is the sampling interval unit that is chosen based on the relative speeds of the robot and obstacles.  $\bar{k}$  is the quotient of  $T/T_s$ . At each refinement time-point  $t = t_0 + kT_s$ , velocity alteration of each obstacle is detected, and thus trajectory parameters  $c_i^k$  and  $d_i^k$  ( $i = 0, \dots, 6$ ) will be updated. In other words, the first time the trajectory (the parameters) is computed for the entire time interval  $[t_0, t_f]$ . But after time  $t = t_0 + kT_s$ , new information about obstacles are obtained, and the parameters are re-computed for a trajectory between  $[t_0 + kT_s, t_f]$ . The following theorem cited from Guo et al. (2007) defines the family of parameterized trajectories.

**Theorem 1** For  $t \in [t_k, t_f]$ ,  $t_k = t_0 + kT_s$ , the parameterized trajectory for the robot can be described as

$$\begin{aligned} x(t) &= \bar{f}(t)(G^k)^{-1} (E^k - H^k c_6^k) + c_6^k t^6 \\ y(t) &= \bar{f}(t)(G^k)^{-1} (F^k - H^k d_6^k) + d_6^k t^6 \end{aligned} \quad (4)$$

where  $\bar{f}(t) = [1 \ t \ t^2 \ t^3 \ t^4 \ t^5]$ , and

$$\begin{aligned} G^k &= \begin{bmatrix} 1 & t_k & t_k^2 & t_k^3 & t_k^4 & t_k^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_k & 3t_k^2 & 4t_k^3 & 5t_k^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_k & 12t_k^2 & 20t_k^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \\ E^k &= \left[ x_k \ x_f \ \frac{dx}{dt} \Big|_{t_k} \ \frac{dx}{dt} \Big|_{t_f} \ \frac{d^2x}{dt^2} \Big|_{t_k} \ \frac{d^2x}{dt^2} \Big|_{t_f} \right]^T \\ F^k &= \left[ y_k \ y_f \ \frac{dy}{dt} \Big|_{t_k} \ \frac{dy}{dt} \Big|_{t_f} \ \frac{d^2y}{dt^2} \Big|_{t_k} \ \frac{d^2y}{dt^2} \Big|_{t_f} \right]^T \\ H^k &= \left[ t_k^6 \ t_f^6 \ 6t_k^5 \ 6t_f^5 \ 30t_k^4 \ 30t_f^4 \right]^T \end{aligned} \quad (5)$$

*Proof* The proof is straightforward, and directly follows from (2) and the boundary conditions given in (3) with the replacement of  $t_0$  by  $t_k$  for the considered time instant  $t_k$ .  $\square$

It follows from the parameterized trajectory in (4) that the trajectory generation problem boils down to solve for  $c_6^k$  and  $d_6^k$  based on constraints due to system model, collision avoidance and performance. Moreover, once the specific trajectory is obtained, steering control of the robot could always be solvable. Considering robot kinematic constraints, we can obtain the states and steering inputs at each time instant as follows.

$$\begin{aligned} \theta &= \arctan \frac{dy}{dx}, \quad \cos \theta = \sqrt{\frac{1}{1 + \left(\frac{dy}{dx}\right)^2}} \\ \varphi &= \arctan \left( l \cos^3 \theta \cdot \frac{d^2y}{dx^2} \right), \quad u_1 = \pm \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} \\ u_2 &= lu_1 \left[ \frac{(\ddot{y}(t)\dot{x}(t) - \ddot{x}(t)\dot{y}(t))u_1^2}{u_1^6 + l^2(\ddot{y}(t)\dot{x}(t) - \ddot{x}(t)\dot{y}(t))^2} \right. \\ &\quad \left. - \frac{3(\ddot{y}(t)\dot{x}(t) - \ddot{x}(t)\dot{y}(t))(\dot{x}(t)\ddot{x}(t) + \dot{y}(t)\ddot{y}(t))}{u_1^6 + l^2(\ddot{y}(t)\dot{x}(t) - \ddot{x}(t)\dot{y}(t))^2} \right] \end{aligned} \quad (6)$$

*Remark 1* Note that the sign of driving input  $u_1$  depends on the choice of executing the trajectory with forward or backward car motion, respectively.  $\square$

*Remark 2* Different from Qu et al. (2004), Guo and Tang (2008), Guo et al. (2007), by (3) we further consider the relation between robot kinematic model and the higher-dimensional robot states  $\dot{x}$ ,  $\dot{y}$ ,  $\ddot{x}$  and  $\ddot{y}$  derived from the time-parametric trajectory, and hence we are able to guarantee that the given general robot boundary conditions including velocity and acceleration are well consistent with the ones derived directly by the computed trajectory model, leading to compatible planning at boundary states. The computation for intermediate boundary conditions is as follows. For  $t \in [t_k, t_f]$ , the boundary conditions for computing  $E^k$ ,  $F^k$  and  $H^k$  are obtained as



$$\begin{aligned} \frac{dx}{dt} \Big|_{t_k} &= v_k \cos \theta_k, & \frac{dy}{dt} \Big|_{t_k} &= v_k \sin \theta_k \\ \frac{d^2x}{dt^2} \Big|_{t_k} &= a_k \cos \theta_k - \frac{v_k^2 \tan \varphi_k}{l} \sin \theta_k \\ \frac{d^2y}{dt^2} \Big|_{t_k} &= a_k \sin \theta_k + \frac{v_k^2 \tan \varphi_k}{l} \cos \theta_k \end{aligned} \quad (7)$$

where  $v_k \triangleq u_1(t_k)$ ,  $a_k \triangleq \dot{u}_1(t_k)$ ,  $\theta_k$ , and  $\varphi_k$  are calculated according to (6) by using the parameterized trajectory  $x(t)$  and  $y(t)$  in terms of  $c_6^{k-1}$  and  $d_6^{k-1}$  at the previous sampling instant.  $\square$

### 3.2 An optimal solution framework

The piecewise-constant parameterized trajectory in (4) defines a family of trajectories given different values of  $c_6^k$  and  $d_6^k$ . By selecting  $c_6^k$  and  $d_6^k$  according to certain criteria, a feasible, collision free and performance-guaranteed trajectory could be analytically obtained. In this section, we first present for the optimal and analytical solution to trajectory generation of robot moving from the initial point to the final point in a workspace without obstacles. Then, Sect. 4 will deal with the cases in the presence of robot kinodynamic constraints and moving obstacles under the solution framework to be designed in this section.

#### 3.2.1 A combined performance index

To generate an optimal trajectory, it is common to introduce some performance indices related to control energy or the traveled path length. For instance, minimum energy-related performance index and shortest path-related performance index can be defined as follows, respectively,

$$J_k^{E1}(c_6^k, d_6^k) = \int_{t_k}^{t_f} \left( \left( \frac{u_1}{\rho} \right)^2 + u_2^2 \right) dt \quad (8)$$

$$J_k^{L1}(c_6^k, d_6^k) = \int_{x_k}^{x_f} \sqrt{1 + \left( \frac{dy}{dx} \right)^2} dx \quad (9)$$

where  $\rho$  is the radius of rear wheel. It follows from (4) and (6) that  $J_k^{E1}$  and  $J_k^{L1}$  are nonlinear functions of  $c_6^k$  and  $d_6^k$ . While standard numerical method may be pursued to seek the solutions for  $c_6^k$  and  $d_6^k$  which minimize  $J_k^{E1}$  and  $J_k^{L1}$ , it is usually computationally expensive and may not be suitable for real time trajectory generation. Instead, we prefer to have an analytical solution. In this paper, we propose a new combined performance index which could lead to an optimal and analytical solution to trajectory generation of mobile robots.

The proposed combined performance index is defined as

$$J_k = \omega_1 J_k^{E2} + \omega_2 J_k^{L2}, \quad s.t. \quad \omega_1 + \omega_2 = 1 \quad (10)$$

where  $\omega_1$  and  $\omega_2$  are the corresponding weights of minimum-energy index  $J_k^{E2}$  and shortest-path index  $J_k^{L2}$ ,

$$J_k^{E2}(c_6^k, d_6^k) = \frac{1}{\rho^2} \int_{t_k}^{t_f} (\dot{x}^2 + \dot{y}^2) dt \quad (11)$$

and

$$J_k^{L2}(c_6^k, d_6^k) = \int_{t_k}^{t_f} [(x - x')^2 + (y - y')^2] dt \quad (12)$$

with

$$x' = \frac{x_f - x_k}{t_f - t_k}(t - t_k) + x_k, \quad y' = \frac{y_f - y_k}{t_f - t_k}(t - t_k) + y_k$$

It should be noted that  $J_k^{E2}$  and  $J_k^{J2}$  are analogies to  $J_k^{E1}$  and  $J_k^{J1}$ , respectively. In particular, for minimum-energy related performance index, instead of putting penalty on both controls  $u_1$  and  $u_2$ , only  $u_1$  appears in the definition of  $J_k^{E2}$ . This is reasonable since in most cases, steering control input  $u_2$  is small and driving velocity  $u_1$  dominates. More importantly,  $J_k^{E2}$  is quadratic in terms of  $c_6^k$  and  $d_6^k$ , and it can be solved analytically. As well, for shortest-path related performance index,  $J_k^{L2}$  in (12) is applied, which measures the closeness of the trajectory  $(x(t), y(t))$  to the straight line connecting points  $(x_k, y_k)$  and  $(x_f, y_f)$ . Apparently, by minimizing  $J_k^{L2}$ , we will be able to obtain the near shortest trajectory through finding the analytical solution for  $c_6^k$  and  $d_6^k$ .

In this paper, we take into account both minimum-energy related performance index and shortest-path related performance index in a unified way by employing  $J_k$  in (10). Weights  $w_1$  and  $w_2$  are used to more flexibly adjust the preference for different performance requirements. To this end, the optimization problem becomes

$$\min J_k(c_6^k, d_6^k) \quad (13)$$

#### 3.2.2 Analytical solution

**Theorem 2** At each time instant  $t_k$ , the optimization problem (13) is solvable, and its solutions are

$$\begin{cases} c_6^{k*} = \frac{\omega_1 n_2^k c_6^{kE*} + \rho^2 \omega_2 p_2^k c_6^{kL*}}{\omega_1 n_2^k + \rho^2 \omega_2 p_2^k} \\ d_6^{k*} = \frac{\omega_1 n_2^k d_6^{kE*} + \rho^2 \omega_2 p_2^k d_6^{kL*}}{\omega_1 n_2^k + \rho^2 \omega_2 p_2^k} \end{cases} \quad (14)$$

where  $n_2^k$ ,  $p_2^k$ ,  $c_6^{kE*}$ ,  $d_6^{kE*}$ ,  $c_6^{kL*}$ ,  $d_6^{kL*}$  are given in equations (19), (25), (21), and (28), respectively.

*Proof* Since the combined performance index  $J_k$  in (10) consists of two independent terms  $\omega_1 J_k^{E2}$  and  $\omega_2 J_k^{L2}$ , we first consider those two terms separately, then combine their solutions into (10).

- (1) *The optimal solution to  $\min \omega_1 J_k^{E2}$* : Substituting the derivatives  $\dot{x}$  and  $\dot{y}$  of (4) into (11), we have

$$\begin{aligned} \omega_1 J_k^{E2} (c_6^k, d_6^k) &= \frac{\omega_1}{\rho^2} \int_{t_k}^{t_f} (\dot{x}^2 + \dot{y}^2) dt \\ &= \frac{\omega_1}{\rho^2} \left[ n_2^k \left( c_6^k + \frac{n_1^k}{2n_2^k} \right)^2 + n_2^k \left( d_6^k + \frac{n_3^k}{2n_2^k} \right)^2 \right. \\ &\quad \left. + (n_0^k + n_4^k) - \frac{(n_1^k)^2 + (n_3^k)^2}{4n_2^k} \right] \end{aligned} \quad (15)$$

where

$$\begin{aligned} n_0^k &= \int_{t_k}^{t_f} (\bar{f}'(G^k)^{-1} E^k)^2 dt \\ n_1^k &= 2 \int_{t_k}^{t_f} (6t^5 - \bar{f}'(G^k)^{-1} H^k) (\bar{f}'(G^k)^{-1} E^k) dt \\ n_2^k &= \int_{t_k}^{t_f} (6t^5 - \bar{f}'(G^k)^{-1} H^k)^2 dt \\ n_3^k &= 2 \int_{t_k}^{t_f} (6t^5 - \bar{f}'(G^k)^{-1} H^k) (\bar{f}'(G^k)^{-1} F^k) dt \\ n_4^k &= \int_{t_k}^{t_f} (\bar{f}'(G^k)^{-1} F^k)^2 dt \\ \bar{f}' &= \begin{bmatrix} 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \end{bmatrix} \end{aligned} \quad (16)$$

It follows from the last equation in (15) that  $\omega_1 J_k^{E2}$  is minimized if

$$c_6^{kE*} = -\frac{n_1^k}{2n_2^k}, \quad d_6^{kE*} = -\frac{n_3^k}{2n_2^k} \quad (17)$$

On the other hand, direct integration of (16) leads to

$$\begin{aligned} n_1^k &= \frac{(t_k - t_f)^7 \left( \frac{d^2 x}{dt^2} \Big|_{t_k} + \frac{d^2 x}{dt^2} \Big|_{t_f} \right)}{420} \\ &\quad - \frac{2(t_k - t_f)^6 \left( \frac{dx}{dt} \Big|_{t_k} - \frac{dx}{dt} \Big|_{t_f} \right)}{105} \end{aligned} \quad (18)$$

$$n_2^k = \frac{(t_f - t_k)^{11}}{770} \quad (19)$$

$$\begin{aligned} n_3^k &= \frac{(t_k - t_f)^7 \left( \frac{d^2 y}{dt^2} \Big|_{t_k} + \frac{d^2 y}{dt^2} \Big|_{t_f} \right)}{420} \\ &\quad - \frac{2(t_k - t_f)^6 \left( \frac{dy}{dt} \Big|_{t_k} - \frac{dy}{dt} \Big|_{t_f} \right)}{105} \end{aligned} \quad (20)$$

To this end, substituting (18), (19), and (20) into (17) yields

$$\begin{cases} c_6^{kE*} = \frac{22 \left( \frac{dx}{dt} \Big|_{t_k} - \frac{dx}{dt} \Big|_{t_f} \right)}{3(t_f - t_k)^5} + \frac{11 \left( \frac{d^2 x}{dt^2} \Big|_{t_k} + \frac{d^2 x}{dt^2} \Big|_{t_f} \right)}{12(t_f - t_k)^4} \\ d_6^{kE*} = \frac{22 \left( \frac{dy}{dt} \Big|_{t_k} - \frac{dy}{dt} \Big|_{t_f} \right)}{3(t_f - t_k)^5} + \frac{11 \left( \frac{d^2 y}{dt^2} \Big|_{t_k} + \frac{d^2 y}{dt^2} \Big|_{t_f} \right)}{12(t_f - t_k)^4} \end{cases} \quad (21)$$

- (2) *The optimal solution to  $\min \omega_2 J_k^{L2}$* : Similarly, by substituting the trajectory polynomials in (4) into (12), we obtain

$$\begin{aligned} \omega_2 J_k^{L2} (c_6^k, d_6^k) &= \omega_2 \int_{t_k}^{t_f} [(x - x')^2 + (y - y')^2] dt \\ &= \omega_2 \left[ p_2^k \left( c_6^k + \frac{p_1^k}{2p_2^k} \right)^2 + p_2^k \left( d_6^k + \frac{p_3^k}{2p_2^k} \right)^2 \right. \\ &\quad \left. + (p_0^k + p_4^k) - \frac{(p_1^k)^2 + (p_3^k)^2}{4p_2^k} \right] \end{aligned} \quad (22)$$

where

$$\begin{aligned} p_0^k &= \int_{t_k}^{t_f} (\bar{f}(G^k)^{-1} E^k - x')^2 dt \\ p_1^k &= 2 \int_{t_k}^{t_f} (t^6 - \bar{f}(G^k)^{-1} H^k) (\bar{f}(G^k)^{-1} E^k - x') dt \\ p_2^k &= \int_{t_k}^{t_f} (t^6 - \bar{f}(G^k)^{-1} H^k)^2 dt \\ p_3^k &= 2 \int_{t_k}^{t_f} (t^6 - \bar{f}(G^k)^{-1} H^k) (\bar{f}(G^k)^{-1} F^k - y') dt \\ p_4^k &= \int_{t_k}^{t_f} (\bar{f}(G^k)^{-1} F^k - y')^2 dt \\ \bar{f} &= \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \end{bmatrix} \end{aligned} \quad (23)$$

Direct integration of (23) leads to

$$\begin{aligned} p_1^k &= \frac{3(t_k - t_f)^8 \left( \frac{dx}{dt} \Big|_{t_f} - \frac{dx}{dt} \Big|_{t_k} \right)}{1540} \\ &\quad - \frac{(t_f - t_k)^9 \left( \frac{d^2 x}{dt^2} \Big|_{t_k} + \frac{d^2 x}{dt^2} \Big|_{t_f} \right)}{5544} \end{aligned} \quad (24)$$

$$p_2^k = \frac{(t_f - t_k)^{13}}{12012} \quad (25)$$

$$\begin{aligned} p_3^k &= \frac{3(t_k - t_f)^8 \left( \frac{dy}{dt} \Big|_{t_f} - \frac{dy}{dt} \Big|_{t_k} \right)}{1540} \\ &\quad - \frac{(t_f - t_k)^9 \left( \frac{d^2 y}{dt^2} \Big|_{t_k} + \frac{d^2 y}{dt^2} \Big|_{t_f} \right)}{5544} \end{aligned} \quad (26)$$

It follows from the last equation in (22) that  $\omega_2 J_k^{L_2}$  is minimized if

$$c_6^{kL*} = -\frac{p_1^k}{2p_2^k}, \quad d_6^{kL*} = -\frac{p_3^k}{2p_2^k} \quad (27)$$

To this end, substituting (24), (25) and (26) into (27), we have

$$\begin{aligned} c_6^{kL*} &= \frac{22\left(\frac{dx}{dt}\big|_{t_k} - \frac{dx}{dt}\big|_{t_f}\right)}{3(t_f - t_k)^5} + \frac{11\left(\frac{d^2x}{dt^2}\big|_{t_k} + \frac{d^2x}{dt^2}\big|_{t_f}\right)}{12(t_f - t_k)^4} \\ d_6^{kL*} &= \frac{22\left(\frac{dy}{dt}\big|_{t_k} - \frac{dy}{dt}\big|_{t_f}\right)}{3(t_f - t_k)^5} + \frac{11\left(\frac{d^2y}{dt^2}\big|_{t_k} + \frac{d^2y}{dt^2}\big|_{t_f}\right)}{12(t_f - t_k)^4} \end{aligned} \quad (28)$$

(3) *The optimal solution to min  $J_k$ :*

It follows from (15) and (22) that

$$\begin{aligned} J_k &= \omega_1 J_k^{E_2} + \omega_2 J_k^{L_2} \\ &= \frac{\omega_1 n_2^k}{\rho^2} \left[ \left( c_6^k + \frac{n_1^k}{2n_2^k} \right)^2 + \left( d_6^k + \frac{n_3^k}{2n_2^k} \right)^2 \right] \\ &\quad + \omega_2 p_2^k \left[ \left( c_6^k + \frac{p_1^k}{2p_2^k} \right)^2 + \left( d_6^k + \frac{p_3^k}{2p_2^k} \right)^2 \right] + \Delta^k \end{aligned} \quad (29)$$

where

$$\begin{aligned} \Delta^k &= \frac{\omega_1}{\rho^2} \left[ (n_0^k + n_4^k) - \frac{(n_1^k)^2 + (n_3^k)^2}{4n_2^k} \right] \\ &\quad + \omega_2 \left[ (p_0^k + p_4^k) - \frac{(p_1^k)^2 + (p_3^k)^2}{4p_2^k} \right] \end{aligned}$$

It follows that  $J_k$  is a second-order polynomial in terms of  $c_6^k$  and  $d_6^k$ , and its minimal value is achieved when

$$\frac{\partial J_k}{\partial c_6^k} = 0, \quad \frac{\partial J_k}{\partial d_6^k} = 0$$

That is,

$$c_6^{k*} = -\frac{\omega_1' \frac{n_1^k}{2n_2^k} + \omega_2' \frac{p_1^k}{2p_2^k}}{\omega_1' + \omega_2'}, \quad d_6^{k*} = -\frac{\omega_1' \frac{n_3^k}{2n_2^k} + \omega_2' \frac{p_3^k}{2p_2^k}}{\omega_1' + \omega_2'} \quad (30)$$

where  $\omega_1' = \frac{\omega_1 n_2^k}{\rho^2}$  and  $\omega_2' = \omega_2 p_2^k$ . To this end, noting the expressions in (17) and (27), equations (30) are equivalent to (14). This completes the proof.  $\square$

**Remark 3** Theorem 2 provides an analytical solution to find the optimal values  $c_6^{k*}$  and  $d_6^{k*}$  under the combined perfor-

mance index  $J_k$ . The weights  $\omega_1$  and  $\omega_2$  are used in  $J_k$  to weigh the relative importance between the minimum-energy and shortest-path performance. Particularly,  $J_k$  becomes the minimum-energy related performance index  $J_k^{E_2}$  when  $\omega_1 = 1$ , and  $J_k$  reduces to the shortest-path related performance index  $J_k^{L_2}$  for  $\omega_2 = 1$ .  $\square$

**Remark 4** It is for the purpose of easier computations to choose  $c_6^k$  and  $d_6^k$  as the free parameters. But the values of the rest parameters such as  $c_0^k, \dots, c_5^k$  and  $d_0^k, \dots, d_5^k$  are dependent on  $c_6^k$  and  $d_6^k$ . Although the dominant optimal solutions are referred to as the optimal value of  $c_6^k$  and  $d_6^k$ , actually the real optimized parameters that construct the whole trajectory are the combination of free parameters  $c_6^*$  and  $d_6^*$ , and the  $c_0^{k*}, \dots, c_5^{k*}, d_0^{k*}, \dots, d_5^{k*}$  that are subsequently derived from  $c_6^{k*}, d_6^{k*}$  with the boundary conditions. It is equivalent no matter which pair of parameters  $c_i^k$  and  $d_i^k$  is chosen as the two free parameters.  $\square$

#### 4 Suboptimal solution under kinodynamic constraints and presence of moving obstacles

The result in Sect. 3 provides a guideline to find the optimal solution in the free space. Under the same framework, in this section, we address the issue in the presence of kinodynamic constraints and moving obstacles.

##### 4.1 Kinodynamic constraints

Since we mainly consider kinematic issues of mobile robots, it is reasonable to demonstrate kinodynamic constraints by velocity and acceleration bounds. Then it is straightforward to consider kinodynamic constraints with the following inequations. We assume that absolute values of maximum bounds on both directions of velocity and acceleration (positive or negative) are the same, and they are denoted by  $v_{max}$  and  $a_{max}$ .

$$v_x^2(t) + v_y^2(t) \leq v_{max}^2 \quad (31)$$

$$a_x^2(t) + a_y^2(t) \leq a_{max}^2 \quad (32)$$

To this end, substituting derivations of (4) into (31) and (32), we could obtain the following Theorem.

**Theorem 3** For  $t \in [t_k, t_f]$ ,  $t_k = t_0 + kT_s$ , the kinodynamic constraints (31) and (32) can be satisfied with the following constraints on  $c_6^k$  and  $d_6^k$ .

$$\left( c_6^k + \frac{m_1^k(t)}{m_2^k(t)} \right)^2 + \left( d_6^k + \frac{m_3^k(t)}{m_2^k(t)} \right)^2 \leq \frac{v_{max}^2}{(m_2^k(t))^2} \quad (33)$$

$$\left( c_6^k + \frac{s_1^k(t)}{s_2^k(t)} \right)^2 + \left( d_6^k + \frac{s_3^k(t)}{s_2^k(t)} \right)^2 \leq \frac{a_{max}^2}{(s_2^k(t))^2} \quad (34)$$

where

$$\begin{aligned}
 m_1^k(t) &= \bar{f}'(G^k)^{-1} E^k \\
 m_2^k(t) &= 6t^5 - \bar{f}'(G^k)^{-1} H^k, \quad m_2^k(t) \neq 0 \\
 m_3^k(t) &= \bar{f}'(G^k)^{-1} F^k \\
 s_1^k(t) &= \bar{f}''(G^k)^{-1} E^k \\
 s_2^k(t) &= 30t^4 - \bar{f}''(G^k)^{-1} H^k, \quad s_2^k(t) \neq 0 \\
 s_3^k(t) &= \bar{f}''(G^k)^{-1} F^k \\
 \bar{f}' &= \begin{bmatrix} 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \end{bmatrix} \\
 \bar{f}'' &= \begin{bmatrix} 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix}
 \end{aligned} \quad (35)$$

*Proof* Notice that  $v_x(t) = \dot{x}(t)$ ,  $v_y(t) = \dot{y}(t)$ ,  $a_x(t) = \ddot{x}(t)$  and  $a_y(t) = \ddot{y}(t)$ . Then it is straightforward to substitute the differential form of (4) into (31) and (32), which could directly yield (33) and (34). This completes the proof.  $\square$

*Remark 5* It is possible that  $m_2^k(t)$  or  $s_2^k(t)$  is equal to zero. Consider expressions of these two in (35). By numerical analysis, such cases occur when  $t$  is close to  $\frac{t_f - t_0}{2}$  ( $m_2^k(t) = 0$ ),  $\frac{t_f - t_0}{4}$  and  $\frac{3}{4}(t_f - t_0)$  ( $s_2^k(t) = 0$ ), denoted by  $t_v^*$ ,  $t_a^*$  and  $t_a^{**}$  respectively. In this case, three additional inequalities should be obeyed as follows.

$$\begin{cases} (m_1^k(t))^2(t_v^*) + (m_3^k(t))^2(t_v^*) \leq v_{\max}^2 \\ (s_1^k(t))^2(t_a^*) + (s_3^k(t))^2(t_a^*) \leq a_{\max}^2 \\ (s_1^k(t))^2(t_a^{**}) + (s_3^k(t))^2(t_a^{**}) \leq a_{\max}^2 \end{cases} \quad (36)$$

It is noted that these three equations are irrelevant with  $c_6^k$ ,  $d_6^k$ , and can be precalculated to decide whether there exists feasible solutions.  $\square$

#### 4.2 Collision-avoidance criterion in dynamic environment

In Fig. 1 during one sampling interval  $t \in [t_0 + kT_s, t_0 + (k+1)T_s]$ , ( $k = 0, 1, \dots, \bar{k}$ ) robot is located at  $O^k(x(t), y(t))$  with radius  $R$  moving with velocity  $v_r \triangleq [\dot{x}(t) \ \dot{y}(t)]^T$  and the  $i$ th obstacle is located at  $O_i^k(x_i(t), y_i(t))$  with radius  $r_i$ . It can be assumed that for the  $i$ th obstacle, the detected velocity  $v_i^k$  remains constant during each sampling interval. The static barriers in Fig. 1 can also be considered as moving obstacles with zero velocities. Then relative velocity of robot and the  $i$ th obstacle is

$$v_{r,i}^k \triangleq v_r - v_i^k = \begin{bmatrix} v_{r,i,x}^k \\ v_{r,i,y}^k \end{bmatrix} = \begin{bmatrix} \dot{x} - v_{i,x}^k \\ \dot{y} - v_{i,y}^k \end{bmatrix} \quad (37)$$

From (37), we can consider moving obstacles as static. Then with velocity obstacles method (Shiller et al. 2011), during  $t \in [t_0 + kT_s, t_f]$  the distance between centers of robot and the  $i$ th obstacle must satisfy:

$$(x_i'(t) - x_i^k)^2 + (y_i'(t) - y_i^k)^2 \geq (r_i + R)^2 \quad (38)$$

where  $x_i'(t) = x(t) - v_{i,x}^k \tau$ ,  $y_i'(t) = y(t) - v_{i,y}^k \tau$  (relative position of the robot with respect to the static obstacle),  $\tau = t - (t_0 + kT_s)$ , for  $t \in [t_0 + kT_s, t_f]$ .

**Theorem 4** For  $t \in [t_k, t_f]$ ,  $t_k = t_0 + kT_s$ , the collision avoidance criterion for the  $i$ th obstacle in (38) can be satisfied with the following constraints on  $c_6^k$  and  $d_6^k$ .

$$\left( c_6^k + \frac{g_{1,i}^k(t)}{g_{2,i}^k(t)} \right)^2 + \left( d_6^k + \frac{g_{3,i}^k(t)}{g_{2,i}^k(t)} \right)^2 \geq \frac{(r_i + R)^2}{(g_{2,i}^k(t))^2} \quad (39)$$

where

$$\begin{aligned}
 g_{1,i}^k(t) &= \bar{f}(G^k)^{-1} E^k - v_{i,x}^k \tau - x_i^k \\
 g_{2,i}^k(t) &= t^6 - \bar{f}(G^k)^{-1} H^k, \quad g_{2,i}^k(t) \neq 0 \\
 g_{3,i}^k(t) &= \bar{f}(G^k)^{-1} F^k - v_{i,y}^k \tau - y_i^k \\
 \bar{f} &= \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \end{bmatrix}
 \end{aligned} \quad (40)$$

*Proof* The proof is straightforward by directly substituting  $x(t)$  and  $y(t)$  from (4) into (38), which thereby results in (39).

Furthermore, consider the condition of  $g_{2,i}^k(t) = 0$  or close to 0 and its implications. Since  $g_{2,i}^k(t)$  is the coefficient of all the terms containing  $c_6^k$  and  $d_6^k$  in (39),  $c_6^k$  and  $d_6^k$  are hence removed from the inequality and they no longer affect the collision avoidance. In this case, the resulting trajectory in (4) is degenerated into a fixed 5th-order parameterized polynomials. This is impossible except for the boundary time-point. In order to avoid this situation, an assumption is imposed that at boundary positions robot should not be so close to other obstacles. This assumption can also be held upon intermediate boundary conditions by expanding the safety distance in (38) from  $r_i + R$  (the sum of radius of robot and obstacle) to a bigger value, so as to keep enough distance for the robot from the obstacles along the whole path. This completes the proof.  $\square$

*Remark 6* When the relative speeds of the robot and the obstacles change considerably according to the newly updated obstacles' information, the current sampling interval  $T_s$  may be so large that it becomes rather ideal to assume that the motions of obstacles maintain constant within  $T_s$ . In this case the sampling interval  $T_s$  could be altered adaptively on-demand. For example, if the value of  $T_s$  appears to be large and no longer proper at the beginning of sampling time-point  $t_k = t_0 + kT_s$ , then the trick is to consider  $t_k$  as the new starting time-point  $t_0$ , and reorganize the rest maneuver time  $T' = t_f - (t_0 + kT_s)$  by dividing it into more time intervals on-demand than the previous  $\bar{k} - k$  time intervals. As a result, each new sampling interval can be small enough to approximately capture the motion changing of dynamic obstacles, and hence the general assumption of constant obstacles' motions within each sampling interval can still hold.  $\square$



### 4.3 The feasible value area on the parameter space for trajectory generation

Considering the constraints conditions (33), (34) and (39), it is interesting to find that they are all second-order polynomials in terms of  $c_6^k$  and  $d_6^k$ . For  $t \in [t_k, t_f]$ ,  $t_k = t_0 + kT_s$ , if we map those inequations on a parameter space of  $c_6^k - d_6^k$ , then they can be reformulated by the following specific circular areas respectively, and the candidate solution of  $c_6^k$  and  $d_6^k$  is represented by specific points on the parameter space. Denoting all these quadratic constraints for all discrete time-point  $t \in [t_k, t_f]$  by a general form of  $g(c_6^k, d_6^k)$ , we have

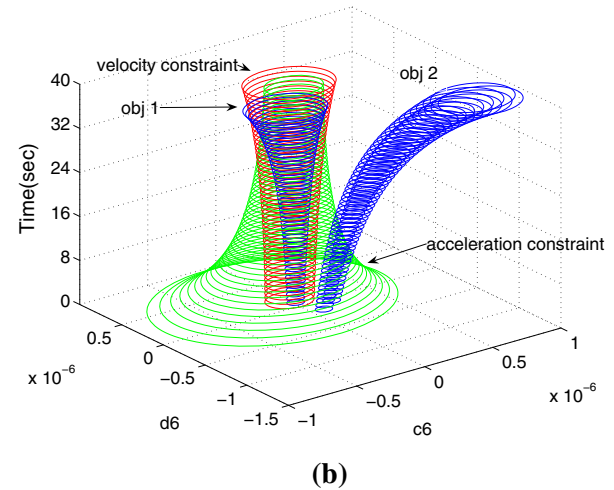
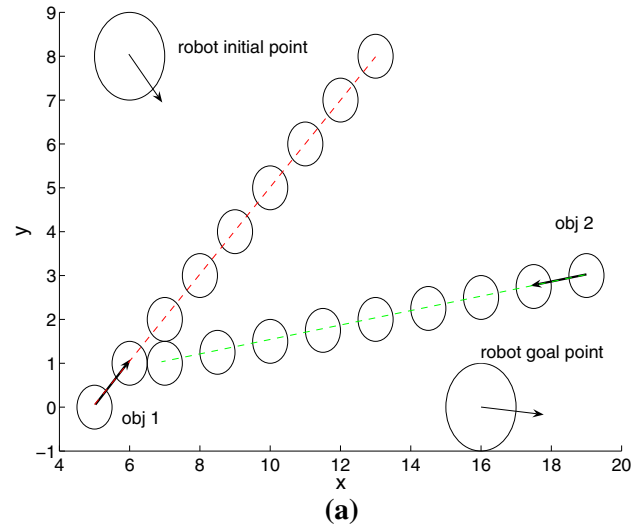
$$\begin{aligned} g(c_6^k, d_6^k) : \\ \Theta_{c,t}^k &= \{(c_6^k, d_6^k) | g_1(c_6^k, d_6^k) \leq 0\} \\ \Theta_{v,t}^k &= \{(c_6^k, d_6^k) | g_2(c_6^k, d_6^k) \leq 0\} \\ \Theta_{a,t}^k &= \{(c_6^k, d_6^k) | g_3(c_6^k, d_6^k) \leq 0\} \end{aligned} \quad (41)$$

where

$$\begin{aligned} g_1(c_6^k, d_6^k) &= -\left(c_6^k + \frac{g_{1,i}^k}{g_{2,i}^k}\right)^2 - \left(d_6^k + \frac{g_{3,i}^k}{g_{2,i}^k}\right)^2 + \frac{(r_i + R)^2}{(g_{2,i}^k)^2}, \\ i &= 1, 2, \dots \\ g_2(c_6^k, d_6^k) &= \left(c_6^k + \frac{m_1^k}{m_2^k}\right)^2 + \left(d_6^k + \frac{m_3^k}{m_2^k}\right)^2 - \frac{v_{max}^2}{(m_2^k)^2} \\ g_3(c_6^k, d_6^k) &= \left(c_6^k + \frac{s_1^k}{s_2^k}\right)^2 + \left(d_6^k + \frac{s_3^k}{s_2^k}\right)^2 - \frac{a_{max}^2}{(s_2^k)^2} \end{aligned}$$

The velocity constrained condition (33) corresponds to  $\Theta_{v,t}^k$ , which indicates the area inside the “velocity circles” centered at  $\left(\frac{m_1^k(t)}{m_2^k(t)}, -\frac{m_3^k(t)}{m_2^k(t)}\right)$  with radius  $\sqrt{\frac{v_{max}^2}{(m_2^k(t))^2}}$ . The acceleration constrained condition (34) corresponds to  $\Theta_{a,t}^k$ , which indicates the area inside the “acceleration circles” centered at  $\left(\frac{s_1^k(t)}{s_2^k(t)}, -\frac{s_3^k(t)}{s_2^k(t)}\right)$  with radius  $\sqrt{\frac{a_{max}^2}{(s_2^k(t))^2}}$ . For collision avoidance constraint (39), the  $i$ th obstacle corresponds to the “collision circles” that centered at  $\left(-\frac{g_{1,i}^k(t)}{g_{2,i}^k(t)}, -\frac{g_{3,i}^k(t)}{g_{2,i}^k(t)}\right)$  with radius  $\sqrt{\frac{(r_i + r_0)^2}{(g_{2,i}^k(t))^2}}$ . Thus  $\Theta_{c,t}^k$  denotes the area outside those “collision circles” from all the detected obstacles. Then a set of circular areas corresponding to the rest of maneuver time should be considered to make the generated trajectory meet the constraints.

To satisfy all the constrained conditions  $g(c_6^k, d_6^k)$ , the candidate parameter point  $(c_6^k, d_6^k)$  should be located at the intersection of those constrained areas (41) on the parameter space as follows (For  $t \in [t_k, t_f]$ ,  $t_k = t_0 + kT_s$ ).

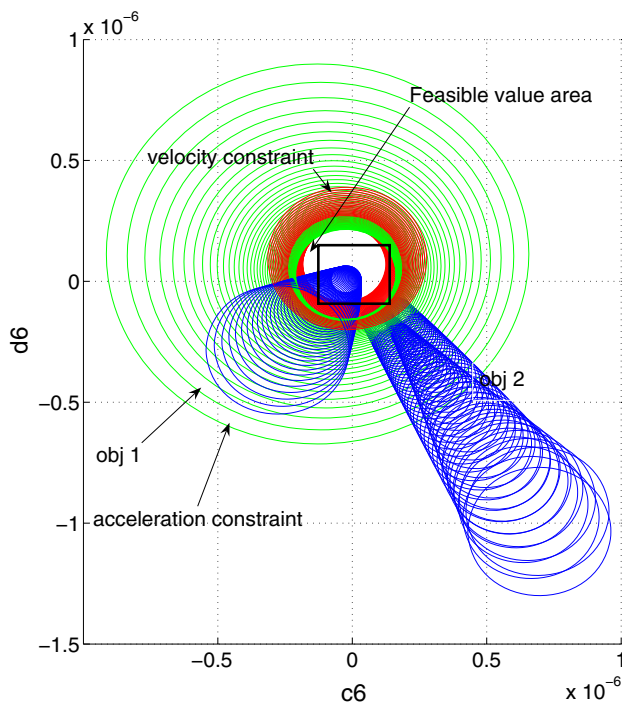


**Fig. 2** An example for constraints areas on parameter space. **a** The robot boundary positions in the presence of two moving obstacles 1 and 2. **b** Kinodynamic and collision avoidance constraints circles in the parameter space during the entire maneuver time

$$\Theta_f^k = \bigcap_{t \in [t_k, t_f]} (\Theta_{c,t}^k \cap \Theta_{v,t}^k \cap \Theta_{a,t}^k), \quad k = 1, 2, \dots, \bar{k} \quad (42)$$

To better clarify the feasible value area (42) on the parameter space, an example is given in Fig. 2.

In Fig. 2a, the mobile robot should move from initial point to goal point and avoid collision with those two moving obstacles. The entire operation time is 40s and obstacles motion remain constant. Then constraints areas in terms of  $c_6^k$  and  $d_6^k$  at each moment can be plotted in 3-dimensional Fig. 2b. Kinodynamic constraints areas  $\Theta_{v,t}^k$  and  $\Theta_{a,t}^k$  are inside red circles and green circles respectively (velocity and acceleration). Collision avoidance constraints areas  $\Theta_{c,t}^k$  are outside the blue circles of those two obstacles. Since obstacles' motion is constant in this case,  $c_6^k$  and  $d_6^k$  will not be updated and remain constant as  $c_6$  and  $d_6$ . Then for the entire maneuver time, candidate points  $(c_6, d_6, t)$  should be inside the red



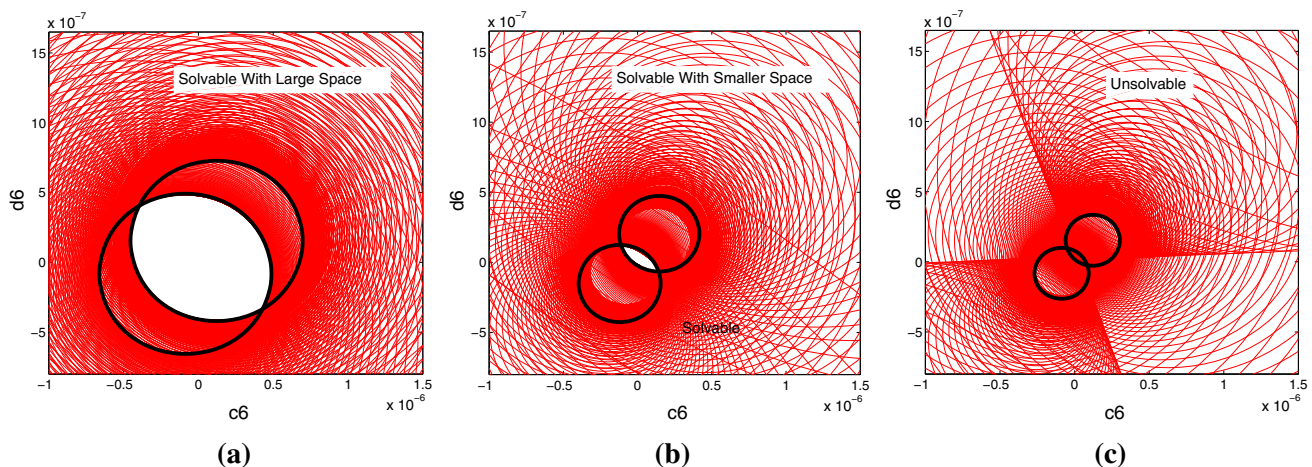
**Fig. 3** Constraints circle areas projection on  $c_6 - d_6$  plane

and green columnar areas, while maintaining outside the two blue columnar areas of two obstacles. Since that, the feasible value area  $\Theta_f^k$  for  $c_6$  and  $d_6$  in Fig. 2b can be found by projection on  $c_6 - d_6$  plane (the parameter space) and represented by the blank area marked in Fig. 3 (the intersections of areas of both kinodynamic constraints  $\Theta_{v,t}^k$ ,  $\Theta_{a,t}^k$  and collision avoidance  $\Theta_{c,t}^k$ ). To this end, parameters ( $c_6^k, d_6^k$ ) picked from such area could autonomously generate family of trajectories by substituting them to (4) and make them satisfy all the constraints.

**Solvable conditions** The existence of solutions mainly relies on the existence of intersections of “velocity” and “acceleration” circles during the entire maneuver period. If kinodynamic constraints are initially violated under the given maneuver time, robot should know in advance such that alternative solutions can be applied to continue the trajectory generation. To that end, we investigate intersections of “velocity circles” for instance (the process for “acceleration circles” is the same). In general, if we continuously lower  $v_{max}$ , distribution of “velocity circles” on the parameter space in the entire time span will change as illustrated in Fig. 4.

Figure 4 plots the “velocity circles” (red) at all the moments on the parameter space, and demonstrates typical distributions of “velocity circles” under different velocity bound, where common intersections are formed, if such one exists. The black marked circles represent converged circles where more velocity circles are densely distributed. As the velocity bound decreases, initial solvable intersections will eventually disappear, which indicates the unsolvable conditions under highly strict speed bound. Particularly, by numerical analysis, those converged black circles can be proved to correspond to around the moment when  $t = \left\{ \frac{t_f - t_0}{4}, \frac{3}{4}(t_f - t_0) \right\}$  (for velocity circles  $\Theta_{v,t}$ ) and  $t = \left\{ \frac{t_f - t_0}{8}, \frac{t_f - t_0}{2}, \frac{7}{8}(t_f - t_0) \right\}$  (for acceleration circles  $\Theta_{a,t}$ ) respectively. In order to simplify computation process, we can simply calculate velocity and acceleration circles around those moments to find the common intersection areas. After separately considering feasible intersections of those mapped kinodynamic circles, we can incorporate intersections of both of them to obtain the ultimate feasible solution area on the parameter space.

**Remark 7** In (42), the set of  $\Theta_f^k$  is assumed to be nonempty, which may not always be guaranteed during entire operation time. In fact, if  $\Theta_f^k$  is empty, it means that under given



**Fig. 4** Changing distribution of velocity circles under diversely lowered velocity bound. **a** Solvable under initial velocity bound. **b** Solvable with narrowed space under lowered velocity bound. **c** Unsolvable with no feasible intersections under largely lowered velocity bound

maneuver time, robot could not avoid all the obstacles while conforming current kinodynamic constraints. Intuitively, one way is to extend the maneuver time  $t_f$  such that robot could operate with lower velocity and longer detour to avoid moving obstacles.  $\square$

#### 4.4 Suboptimal solution to the constrained optimization problem on the parameter space

Since the kinodynamic constraints and collision avoidance criterion are recast by corresponding geometric areas on the parameter space, the optimization problem can be reformulated on the parameter space as follows.

$$\begin{aligned} \min J_k &= \omega_1 J_k^{E_2} + \omega_2 J_k^{L_2} \\ s.t. (c_6^k, d_6^k) &\in \Theta_f^k \end{aligned} \quad (43)$$

Recalling the last equation in (22) and (15) as well as the analytical solution (14), it is noticed that the contour of the performance index (43) is a series of circles centered at the time-invariant optimal point (14). Hence the candidate point of  $(c_6^k, d_6^k)$  should be located inside the feasible value area  $\Theta_f^k$ , while staying as close to the optimal point  $O^{k*}(c_6^{k*}, d_6^{k*})$  on the parameter space (constrained minimum distance problem). The suboptimal solution  $O^{k**}$  to the problem can be obtained by projecting  $c_6^*$  and  $d_6^*$  into the set  $\Theta_f^k$  in (42) based on

$$\begin{aligned} O^{k**} &:= \{(c_6^k, d_6^k) | \min \|c_6^k - c_6^{k*}\| + \|d_6^k - d_6^{k*}\|\} \\ \forall (c_6^k, d_6^k) &\in \Theta_f^k \end{aligned} \quad (44)$$

Intuitively, such a problem cannot be directly solved in closed form. Yet recalling the constraints  $g(c_6^k, d_6^k)$  in (41) and the problem (44), the suboptimal solution only lies in (1) optimal point  $O^{k*}$ , iff  $O^{k*} \in \Theta_f^k$ . (2) boundary of circular constraints area  $\Theta_f^k$ , Otherwise. To that end, in order to avoid the need for search and maintain the spirit of analytical solution emphasized in this paper, we could employ a set of lines passing through  $O^{k*}$  to approximate the possible locations of the solutions in closed form by only considering  $O^{k*}$  and all the intersection points between the lines and the circular constraints  $g(c_6^k, d_6^k)$ .

For the purpose of unbiased solutions, we consider  $2N$  lines that uniformly divide the parameter space originated at  $O^{k*}$  into  $4N$  equivalent angles, yielding to the angle between the  $j$ th line and the  $c_6$ -axis in the interval  $[-\frac{\pi}{2}, \frac{\pi}{2}]$  as  $\alpha_{j,2N} = -\frac{\pi}{2} + \frac{j\pi}{2N}$ , for  $j = 1, \dots, 2N$ . Then the expression for the  $j$ th line is written as

$$\begin{aligned} l_j(c_6^k, d_6^k) : \quad d_6^k &= (c_6^k - c_6^{k*}) \tan(\alpha_{j,2N}) + d_6^{k*}, \\ j &= 1, \dots, 2N - 1 \\ d_6^k &= d_6^{k*}, \quad j = 2N \end{aligned} \quad (45)$$

In particular, for  $j = N, 2N$ , the corresponding lines are  $c_6^k = c_6^{k*}$  and  $d_6^k = d_6^{k*}$  respectively, which are identical to lines used in Yang et al. (2010), Guo and Tang (2008) and there may not be solutions for the two specific lines. This is why the parameterization method in (45) is used to find out more potentially feasible solutions from more directions in the parameter space. In terms of analytically obtaining possible solutions, the trick is to find out a set of intersection points on each  $l_j$  due to constraints  $g(c_6^k, d_6^k)$ , and pick the desired one as the selected suboptimal point following the criterion (44).

Firstly, by substituting the line  $l_j$  in (45) into kinodynamic equation constraints  $g_i(c_6^k, d_6^k) = 0, i = 2, 3$  in (41), the upper and lower bounds  $\overline{D}_j^k(c_6^k, \overline{d}_{6,j}^k)$  and  $\underline{D}_j^k(c_6^k, \underline{d}_{6,j}^k)$  of possible solutions  $(c_6^k, d_6^k)$  on  $l_j$  can be obtained in closed form respectively as follows.

$$\begin{aligned} \begin{cases} \overline{c}_{6,j}^k = \min_{t \in [t_k, t_f]} (\overline{c}_{v,j}^k, \overline{c}_{a,j}^k) \\ \underline{d}_{6,j}^k = \min_{t \in [t_k, t_f]} (\underline{d}_{v,j}^k, \underline{d}_{a,j}^k) \end{cases} & \begin{cases} \overline{c}_{6,j}^k = \max_{t \in [t_k, t_f]} (\overline{c}_{v,j}^k, \overline{c}_{a,j}^k) \\ \underline{d}_{6,j}^k = \max_{t \in [t_k, t_f]} (\underline{d}_{v,j}^k, \underline{d}_{a,j}^k) \end{cases} \\ \Gamma_j^k := \{(c_6^k, d_6^k) | c_6^k \in [\underline{c}_{6,j}^k, \overline{c}_{6,j}^k], d_6^k \in [\underline{d}_{6,j}^k, \overline{d}_{6,j}^k], \\ \forall (c_6^k, d_6^k) \in l_j\} \end{aligned} \quad (46)$$

where  $\overline{c}_{v,j}^k, \overline{c}_{a,j}^k, \underline{d}_{v,j}^k, \underline{d}_{a,j}^k$  are the roots of bigger value in each pair of solutions. Likewise,  $\underline{c}_{v,j}^k, \underline{c}_{a,j}^k, \overline{d}_{v,j}^k, \overline{d}_{a,j}^k$  are the roots of smaller value (whenever roots exist). It is noted that all these roots are obtained in analytical forms. If  $\overline{c}_{6,j}^k < \underline{c}_{6,j}^k$  or  $\underline{d}_{6,j}^k < \overline{d}_{6,j}^k$ , it is concluded that under current kinodynamic constraints no solution exists, and the maneuver time should be extended if the robot still have to plan a feasible path.

Next, we consider the intersection points of  $l_j$  and the collision avoidance constraints  $\Theta_{c,t}^k$  so as to obtain a family of possible analytical suboptimal solutions along  $l_j$ . By substituting  $l_j$  into the collision avoidance equation constraints  $g_1(c_6^k, d_6^k) = 0$  in (41) due to the  $i$ th obstacle, we have the subset  $S_{i,j}^k = \{(c_{6,i,j}^k, \overline{d}_{6,i,j}^k), (c_{6,i,j}^k, \underline{d}_{6,i,j}^k)\}$  of possible solutions and the constraints interval  $\Phi_{i,j}^k$  for other possible solutions as follows.

$$\begin{aligned} \begin{cases} \overline{c}_{6,i,j}^k = \max_{t \in [t_k, t_f]} \overline{c}_{c,i,j}^k \\ \underline{d}_{6,i,j}^k = \max_{t \in [t_k, t_f]} \underline{d}_{c,i,j}^k \end{cases} & \begin{cases} \underline{c}_{6,i,j}^k = \min_{t \in [t_k, t_f]} \underline{c}_{c,i,j}^k \\ \overline{d}_{6,i,j}^k = \min_{t \in [t_k, t_f]} \overline{d}_{c,i,j}^k \end{cases} \\ \Phi_{i,j}^k := \{(c_6^k, d_6^k) | c_6^k \in [\underline{c}_{6,i,j}^k, \overline{c}_{6,i,j}^k], d_6^k \in [\underline{d}_{6,i,j}^k, \overline{d}_{6,i,j}^k], \\ \forall (c_6^k, d_6^k) \in l_j\} \end{aligned} \quad (47)$$

where  $\overline{c}_{c,i,j}^k, \overline{d}_{c,i,j}^k$  are the roots of bigger value in each pair of solutions. Likewise,  $\underline{c}_{c,i,j}^k, \underline{d}_{c,i,j}^k$  are the roots of smaller value (whenever roots exist). They can also be derived in closed form. Thus, considering the collision avoidance con-

straints due to all the obstacles as well as the upper and lower bounds, we have a finite set of possible suboptimal solutions  $S_j^k = \overline{D_j^k} \cup D_j^k \cup S_{i,j}^k \cup O^{k*}$  and the constraints interval  $\Phi_j^k = \bigcup \Phi_{i,j}^k$  on line  $l_j$  for  $i = 1, \dots$ . In order to pick up at most one desired suboptimal point  $O_j^{k**}$  from  $S_j^k$  along line  $l_j$ , we have

$$O_j^{k**} := \{(c_6^k, d_6^k) | \min \|c_6^k - c_6^{k*}\| + \|d_6^k - d_6^{k*}\|, (c_6^k, d_6^k) \in \overline{\Phi_{i,j}^k} \cap \Gamma_j^k, \forall (c_6^k, d_6^k) \in S_j^k\} \quad (48)$$

where  $\overline{\Phi_{i,j}^k}$  is the complement of set  $\Phi_{i,j}^k$ . It is noted that in order to simplify the computation,  $\overline{d_{6,j}^k}$ ,  $\overline{d_{6,i,j}^k}$  and  $\overline{d_{6,i,j}^k}$  in (46) and (47) can be directly obtained by substituting  $\overline{c_{6,j}^k}$ ,  $\overline{c_{6,i,j}^k}$  and  $\overline{c_{6,i,j}^k}$  into (45) respectively. Since the elements (possible suboptimal solutions) of the finite set  $S_j^k$  and the boundary points of both  $\overline{\Phi_{i,j}^k}$  and  $\Gamma_j^k$  are all analytically obtained, for each line  $l_j$  the suboptimal solution  $O_j^{k**}$  can then be derived by linear comparison of the elements in  $S_j^k$  and the boundary points of  $\overline{\Phi_{i,j}^k}$  and  $\Gamma_j^k$ , which is efficient and computationally simple for on-line planning.

Considering all the feasible suboptimal points  $O_j^{k**}$  on all the lines  $l_j$ , for  $j = 1, \dots, 2N$ , the desired suboptimal point  $O^{k**}$  can thus be obtained as follows.

$$O^{k**} := \{(c_6^k, d_6^k) | \min \|c_6^k - c_6^{k*}\| + \|d_6^k - d_6^{k*}\|, \forall (c_6^k, d_6^k) \in O_j^{k**}, j = 1, \dots, 2N\} \quad (49)$$

**Algorithm completeness and stability** As mentioned in Remark 7, if the problem in (44) is unsolvable (no  $O_j^{k**}$  can be found for all  $j = 1, \dots, 2N$ ), the algorithm will take alternative ways such as extending the maneuver time and rechecking until the problem becomes solvable prior to planning the path (4). Nonetheless, due to the complexity and uncertain future behavior of the dynamic obstacles, the existence of solutions to the constrained problem (44) cannot be guaranteed using the current framework. It should also be noted that since the trajectory is updated recursively during the entire maneuver time, if the obstacles perform a periodic motion, the algorithm may recompute a same set of trajectories at each refinement time-point that render oscillatory behaviors for the robot to keep flipping among them instead of converging to the goal state.

**Remark 8** It is noted that when computing the upper and lower bounds in (46) and the boundary points in (47), for simplification we just assume that the line  $l_j$  will always intersect with circular equation constraints of  $g_i(c_6^k, d_6^k) = 0$ ,  $i = 1, 2, 3$ , which may not be guaranteed in all the cases. If  $l_j$  have no intersection points with at least one circle of  $g_i(c_6^k, d_6^k) = 0$ ,  $i = 2, 3$  for a certain time-point  $t$ , it indicates that on the current line there is no solutions satisfying

all the kinodynamic constraints, and  $l_j$  should be directly abandoned without further consideration of (47) and (48) on it. Since the given set of lines (45) uniformly discretize all the directions in the parameter space originated from  $O^{k*}$ , considering the actual scale of the circles  $g_2 = 0$  and  $g_3 = 0$ , there will always be at least some intersections between them and certain lines. For the cases of no intersection points with the  $i$ th obstacle in  $g_1(c_6^k, d_6^k) = 0$  during the entire time  $t_f - t_k$ , the corresponding collision avoidance constraint always holds and becomes an inactive constraint, leading to that  $S_{i,j}^k$  contains only one element as  $(c_6^{k*}, d_6^{k*})$ .  $\square$

**Remark 9** The above framework works well in analytically solving the constrained optimization problem in (44). However, as a trade-off, such a method is based on a set of finite lines that impose an additional constraint as (45) to the suboptimal solutions, which may ignore solutions that actually exist while not satisfying the additional constraint. Here we introduce an alternative technique to numerically solve the problem (44), which maybe less efficient while could potentially find more feasible solutions. Rewriting the problem in the quadratic form and retrieving the condition of  $\Theta_j^k$  by the general expression  $g(c_6^k, d_6^k) \leq 0$ , we have the constrained quadratic optimization problem as follows.

$$\min J'_k = (c_6^k - c_6^{k*})^2 + (d_6^k - d_6^{k*})^2 \quad s.t. \quad g(c_6^k, d_6^k) \leq 0 \quad (50)$$

As mentioned before, such a problem cannot be directly solved in closed form, yet can also be solved as a nonlinear programming problem (NLP) by satisfying the inequality constraints  $g(c_6^k, d_6^k)$  at a finite number of discrete time-point within interval  $[t_k, t_f]$ . As illustrated in Morales et al. (2011), first we can obtain the Lagrangian of the problem as

$$L(c_6^k, d_6^k, \lambda) = J'_k - \lambda^T g(c_6^k, d_6^k) \quad (51)$$

where  $\lambda$  is the vector of lagrange multipliers for  $g(c_6^k, d_6^k)$ . Then the Karush Kuhn Tucker (KKT) conditions for the NLP (50) can be given by

$$\begin{aligned} \nabla J'_k - G(c_6^k, d_6^k)^T \lambda &= 0 \\ g(c_6^k, d_6^k) &\leq 0 \\ \lambda &\geq 0 \\ \lambda^T g(c_6^k, d_6^k) &= 0 \end{aligned} \quad (52)$$

where  $G$  is the Jacobian matrix of  $g$ . The solution of the KKT equations is the basis to many nonlinear programming algorithms. These algorithms attempt to compute the Lagrange multipliers directly. Here we solve the problem with the iterative approach Sequential Quadratic Programming algorithm (SQP) from SQP software packages such as Matlab Optimization Toolbox and SQPlab (Bonnans 2006). Such an algorithm is one of the most effective methods for nonlinearly constrained optimization problems. The idea is



that at each iteration, a locally quadratic approximation of the objective is made using a quasi-Newton updating method and a quadratic programming (QP) subproblem is generated with locally linearized constraints whose solution is then used for a line search procedure (Morales et al. 2011; John 2014). Since SQP can deal with infeasible initializations, we can directly set up the initial searching point by the optimal solution  $O^{k*}(c_6^{k*}, d_6^{k*})$ .

It is worth noting that besides the complexity of the inequality constraints due to (41), the computational complexity and the quality of solutions also largely depends on the pre-defined stepsize and tolerance for searching, which maybe hard to properly determine given the complicated obstacle information and different task settings. Given the good analytical property of the results in (49), the method of SQP could be considered as a complementary technique in the extreme situation that all the feasible solutions to (44) are only located in the positions that do not satisfy the additional constraint (45).  $\square$

## 5 Simulation and experiment

In this section, we will show both the simulation and experiment results on a car-like robot to analyze the steering control and their optimal performance. The proposed suboptimal method and other analytical solutions are compared in scenarios with and without obstacles. To demonstrate the flexibility of the proposed method to handle a wide variety of situations, we also extend simulation scenarios to incorporate highly cluttered environment with both static and moving obstacles. In particular, results of solutions obtained before and after remedies to unsolvable situation are presented to verify the effectiveness of the method with remedies.

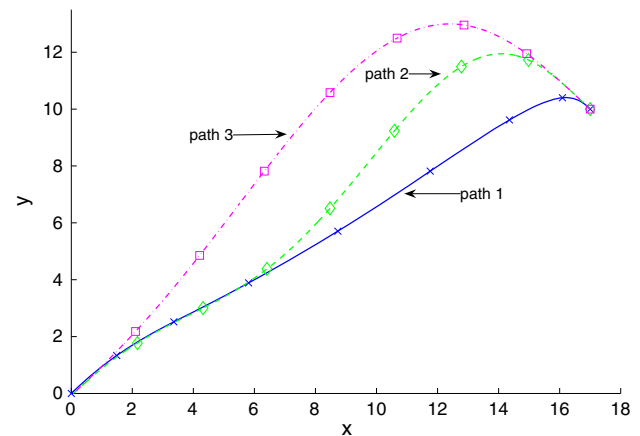
### 5.1 Optimal trajectory generation and comparison without obstacles

In this part, we consider robots moving in a free environment without obstacles.

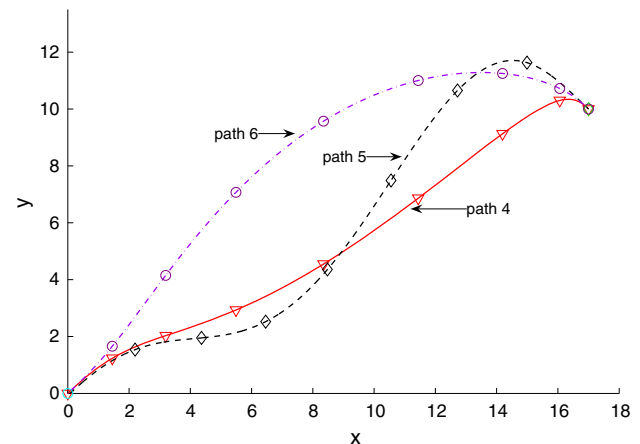
**Example 1 (trivial situation):** The robot task settings are the following:

- Robot radius, distance between front and rear wheel axle, radius of rear wheels:  $r=1, l=0.8, \rho=0.1$ .
- Redefined Robot Boundary conditions:  $q_0^* = (0, 0, \pi/4, 0, 0, 0.4, 0)$  and  $q_f^* = (17, 10, -\pi/4, 0, 0.2, 0)$ .
- Starting time and ending time:  $t_0=0, t_f=40s$

It should be noted that all scales and quantities used conform to a uniform unit system.



**Fig. 5** Energy optimal trajectories under different optimal solutions in obstacles free environment. Path 1, 2 and 3 are generated by the proposed optimal method (energy-optimal), energy optimal solutions in Yang et al. (2010) and ordinary solutions in Qu et al. (2004) respectively



**Fig. 6** Length optimal trajectories under different optimal solutions in obstacles free environment. Path 4, 5 and 6 are generated by the proposed optimal method (length-optimal), length optimal solutions in Yang et al. (2010) and Guo and Tang (2008) respectively

In the case that no obstacles are around, we present robot trajectories under different optimal solutions, including the proposed optimal method and other analytical solutions. In Figs. 5 and 6, the comparison trajectories are divided into two groups of energy-optimal and length-optimal respectively, so that their performance can be better observed. For the purpose of comparison, weights of energy consumption and path length indices in (14) are set to be (1,0) and (0,1), i.e. single optimal solutions of the proposed method are demonstrated. In Yang et al. (2010), different optimal indexes for parametric method have been discussed and two relatively more effective solutions are proposed, which have been used to generate Path 2 and 5. Path 3 is based on basic geometric analytic solution of trajectory generation in Qu et al. (2004), and Path 6 is for Guo and Tang (2008) that uses time-parametric tra-



**Table 1** Path comparison without obstacles in Figs. 5 and 6

	Energy optimal			Length optimal		
	Path	Path	Path	Path	path	path
	1	2	3	4	5	6
Energy	1147.6	1276.4	1477.8	1167.4	1348.0	1348.2
Length	20.27	22.49	24.08	20.20	22.79	22.11

jectory models and optimal solution. Corresponding energy consumption and path length data of the six trajectories in Figs. 5 and 6 are recorded in Table 1. It should be noted that all these trajectories share the same boundary condition  $q = [x, y, \theta, \varphi]^T$  except boundary velocities and accelerations (for path 2, 3 and 5 these boundary conditions cannot be flexibly controlled).

Among the three trajectories of energy-optimal comparison group in Fig. 5, it is obvious that the blue solid path 1 generated by proposed optimal method is the shortest one with no unnecessary swings. Consider the data in Table 1, it is straightforward that when tracking optimal path 1, both energy consumption and path length are much less than those of optimal path 2 (energy consumption saved by 10.1 %, length decreased by 9.9 %) and original path 3 (saved by 22.3 %, length decreased by 15.8 %). For the other three trajectories of length-optimal comparison group in Fig. 6, the red solid optimal path 4 generated by the proposed method also outweighs path 5 and path 6 that planned by other analytic solutions. Likewise, compared with path 5 and 6, the energy consumption and path length of optimal path 4 are largely saved by 13.4 and 11.0 % (to path 5), and 13.4 and 8.3 % (to path 6). These advantages results from facts that the chosen path parameters in path 1 and 4 are closer to optimal points than other four paths, thus obtain better performance. Due to the absence of any constraints in such case, these optimal results prove that the proposed optimal method is in principle more effective to find best optimal solutions than other analytic solutions. Moreover, in parametric methods such as Qu et al. (2004), Yang et al. (2010) the optimality may be further weakened by inducing intermediate waypoints in order to avoid singularity situation that the initial position and ending position for the robot are aligned vertically.

## 5.2 Suboptimal trajectory generation and comparison amidst moving obstacles

This subsection deals with dynamic and cluttered environments. Robots' kinodynamic constraints and obstacles are considered to better demonstrate the validity of proposed method in practical situation and verify the effectiveness of

steering control selection on single or combined optimization problem.

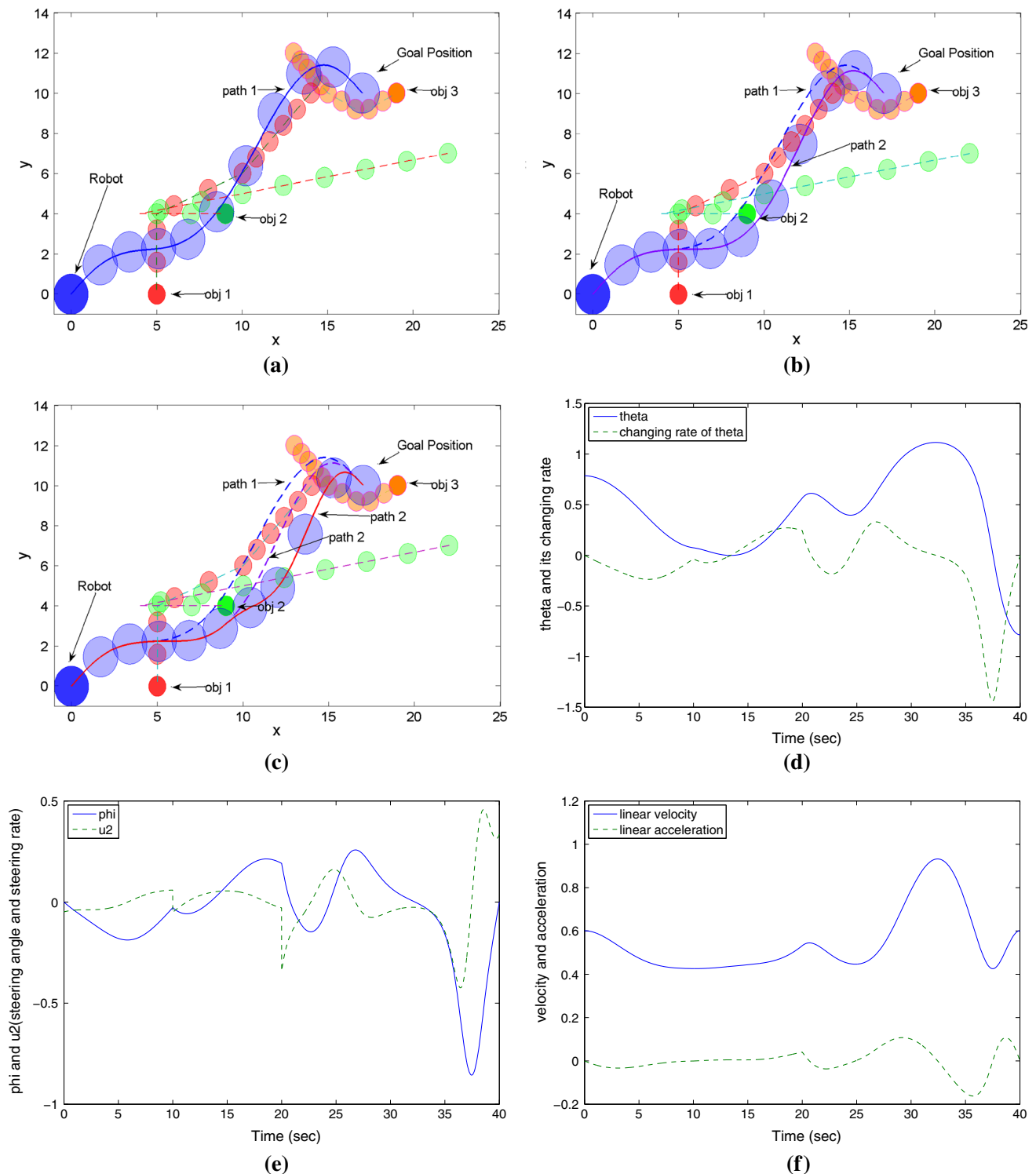
(1) *Example 2 (optimization in dynamic environments)*: This example contains simulations of planning process of the proposed suboptimal method in Fig. 7, and comparisons with other optimal analytic solutions in Figs. 8 and 9. Dynamic environments in these three figures are the same. Default settings are the following (robot structure parameters are the same with those in 5.1):

- Robot kinodynamic constraints:  $v_{max}=1.5$ ,  $a_{max}=0.5$ . Initial coordinates of the three obstacles:  $O_1(t_0) = [5, 0]^T$ ,  $O_2(t_0) = [9, 4]^T$ ,  $O_3(t_0) = [19, 10]^T$ .
- Radius of obstacles:  $r_i=0.5(i=1,2,3)$ .
- Redefined Boundary conditions:  $q_0^* = (0, 0, \pi/4, 0, 0.6, 0)$  and  $q_f^* = (17, 10, -\pi/4, 0, 0.4, 0)$ .
- Starting time and ending time:  $t_0=0$ ,  $t_f=40s$ .
- Velocities of mobile obstacles:

$$\begin{aligned} v_1^0 &= [0, 0.4]^T, v_1^1 = [0.5, 0.2]^T, v_1^2 = v_1^3 = [0.2, 0.2]^T \\ v_2^0 &= [-0.5, 0]^T, v_2^1 = [0.6, 0.1]^T, v_2^2 = v_2^3 = [0.6, 0.1]^T \\ v_3^0 &= [-0.2, -0.1]^T, v_3^1 = [-0.2, 0.1]^T, v_3^2 = v_3^3 = [-0.1, 0.1]^T \end{aligned}$$

Consider a complete planning process of the proposed suboptimal method in Fig. 7. The entire maneuver time is 40s and sampling time-point is chosen to be 0s, 10s and 20s. At each sampling time-point, velocity alteration of each obstacle will be detected, thus trajectory parameters would be updated three times in this specific simulation, which shows the on-line planning property of our algorithm under dynamic environment. During each sampling interval, magnitude and orientation of velocity vector of obstacles remain constant. Simulation results under our suboptimal solutions (take energy-optimal solution for example) are shown in Fig. 7a–f. In Fig. 7a–c, trajectories of robot (blue circles) and moving obstacles 1–3 (red, green and orange circles) are represented by the solid line, the dotted line, the dash-dotted line, and the dashed line, respectively. Note that in order to clearly illustrates the dynamical trajectory of robot and moving obstacles, their locations have been printed once every 4 s during operation time  $T = 40s$ , namely different circles with the same color indicates footprints of those obstacles or robot. Their initial positions are marked with deeper color.

Figure 7a gives a trajectory result (*path 1*) with  $(c_6^1, d_6^1)$  based on obstacles information sensed at 0 s. In this case robot could only be guaranteed collision free for time period  $t \in [0, 10]$  since further velocity alteration of obstacles hasn't been taken into consideration. As between 20 and 25 s, robot will collide with obstacle 1.

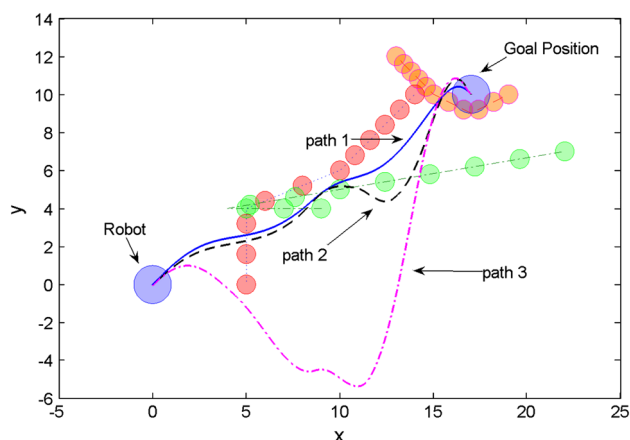


**Fig. 7** Simulation results under energy-optimal solution. **a** Paths of robot and obstacles. **b** Paths of robot and obstacles. **c** Paths of robot and obstacles. **d** Trajectories of robot driving wheels orientation angle and

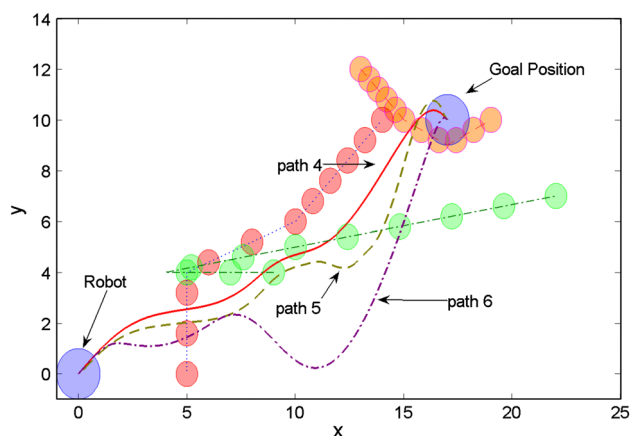
its changing rate. **e** Trajectories of steering angle and steering rate of robot rear wheels. **f** Trajectories of robot driving velocity and acceleration

Figure 7b provides a trajectory result (*path 2*) with two piecewise constant parameters  $(c_6^1, d_6^1)$  and  $(c_6^2, d_6^2)$  based on obstacles information sensed at 0 and 10 s. Also, as

the lack of enough information updating, collision will happen during 24–28 s between robot and obstacle 1. At last, Fig. 7 (c) displays an final trajectory result *path 3* with three



**Fig. 8** Energy optimal trajectories under different optimal solutions amidst mobile obstacles. Path 1, 2 and 3 are generated by the proposed suboptimal method (near minimum energy), energy optimal solutions in Yang et al. (2010) and ordinary solutions in Qu et al. (2004) respectively



**Fig. 9** Length optimal trajectories under different optimal solutions amidst mobile obstacles. Path 4, 5 and 6 are generated by the proposed suboptimal method (near shortest path), length optimal solutions in Yang et al. (2010) and Qu et al. (2004) respectively

piecewise constant parameters  $(c_6^1, d_6^1)$ ,  $(c_6^2, d_6^2)$  and  $(c_6^3, d_6^3)$ , which means at each sampling time-point trajectory parameters would be updated based on the new obstacles moving

information. In such case robot could be ensured safe during the entire maneuver time.

Figure 7d–f shows the corresponding trajectories of robot orientation, steering control inputs as velocity, acceleration, and steering rate of *path 3*. It should be noted that in Fig. 7f the velocity and acceleration are both below their respective limitations as  $v_{max}$  and  $a_{max}$ , which means kinodynamic constraints are strictly obeyed. The corresponding parameters of optimal points  $(c_6^{k*}, d_6^{k*})$  and chosen suboptimal points  $(c_6^k, d_6^k)$  are as follows (obtained at each sampling time-point):

$$\begin{cases} c_6^{1*} = 1 \times 10^{-8}, d_6^{1*} = 5.1 \times 10^{-8} \\ c_6^{2*} = 3.81 \times 10^{-8}, d_6^{2*} = 9.6 \times 10^{-8} \\ c_6^{3*} = 4.74 \times 10^{-7}, d_6^{3*} = 1.84 \times 10^{-6} \end{cases} \quad (53)$$

$$\begin{cases} c_6^1 = 1.5 \times 10^{-8}, d_6^1 = 8 \times 10^{-8} \\ c_6^2 = 3.8 \times 10^{-8}, d_6^2 = 9.65 \times 10^{-8} \\ c_6^3 = 3 \times 10^{-7}, d_6^3 = 2.7 \times 10^{-6} \end{cases} \quad (54)$$

Next, we discuss optimal performance of trajectories under different solutions in the same dynamic environment as the previous one. Also, such comparison is divided into two groups of energy-optimal and length-optimal in Figs. 8 and 9 respectively. In Table 2, corresponding comparative data of energy consumption and path length are provided. It is quite clear that towards mobile obstacles, path 1 and 4 generated by proposed suboptimal method are much shorter (shortened by 9.8 % at least) and more efficient in energy consumption (saving 21.4 % at least) than other solutions, with less unnecessary swings. In those two figures, all the six trajectories can avoid mobile obstacles, while the other four trajectories are more conservative. Path 2, 3 and 5 are based on the geometry-parameterized trajectory model, and the robot's velocity on the x-axis is fixed. Due to that limitation, the robot may have to take bigger swing to avoid obstacles when the robot is very close to them. As what has been analyzed in 5.1, the incompleteness on optimal parameters selection could make path 6 take long detours. It is noted that the length-optimal path 4 seems even longer than energy-optimal path 1 according to the data. Due to the existence of collision constrained areas, in some cases the suboptimal parameters chosen by length

**Table 2** Path comparison with obstacles in Figs. 8 and 9

		Energy	Length	[0,10]		[10,20]		[20,40]	
				$c_6^1 \times 10^{-8}$	$d_6^1 \times 10^{-8}$	$c_6^2 \times 10^{-8}$	$d_6^2 \times 10^{-8}$	$c_6^3 \times 10^{-7}$	$d_6^3 \times 10^{-7}$
Near minimum energy	Path 1	1125.6	20.72	1.5	8	3.8	9.65	3	27
	Path 2	1431.2	22.68						
	Path 3	3710.7	32.54						
Near shortest path	Path 4	1178.2	20.84	1.62	8.1	5.59	15.6	5.4	29.6
	Path 5	1453.8	23.10						
	Path 6	2146.7	24.28	5	8	8	16	10	70

optimal solutions could be closer to energy-optimal points than length-optimal points on the parameter space. Therefore, it is reasonable to find path 4 is a little longer than path 1. Nevertheless, validity of our optimal principles has been proved in *example 1*.

(2) Example 3 (Experiment): In order to better compare the method in this paper with other existing results in [Qu et al. \(2004\)](#), [Yang et al. \(2005\)](#), we have to maintain the same simulation settings in the Example 2, which results in the small numerical range of  $c_6$  and  $d_6$  (in the magnitude of  $10^7$ ). In practice, however, those values may be scaled up according to the given unit system as well as the setting range of robot's driving velocity and angular velocity. A new experiment is provided in this part to show that in real-world settings the algorithm is still effective and efficient to compute.

Compared to the simulation in Example 2, the experiment settings in this part are more practical in the real-world implementation, which are as follows.

- Robot kinodynamic constraints:  $v_{max} = 1.5$ ,  $a_{max} = 0.5$ .
- Initial coordinates of the three dynamic obstacles:  $O_1(t_0) = [750, 2,000]^T$ ,  $O_2(t_0) = [1,000, 4,000]^T$ ,  $O_3(t_0) = [2000, 500]^T$ .
- Initial coordinates of the three static obstacles:  $O_4 = [1,500, 300]^T$ ,  $O_5 = [2,000, 3,500]^T$ ,  $O_6 = [5,000, 3,800]^T$ .
- Radius of dynamic obstacles:  $r_i = 250$  ( $i = 1, 2, 3$ ). Radius of static obstacles:  $r_i = 100$  ( $i = 4, 5, 6$ ).
- Redefined Robot Boundary conditions:  $q_0^* = (500, 500, \pi/16, 0, 0, 0)$  and  $q_f^* = (4,500, 4,500, -\pi/4, 0, 600, 0)$ .
- Starting time and ending time:  $t_0 = 0$ ,  $t_f = 9s$ .
- Velocities of mobile obstacles at 0, 3 and 6 s (within each sampling interval they remain constant):
 
$$\begin{aligned} v_1^0 &= [125, -150]^T, v_1^1 = [780, 850]^T, v_1^2 = [0, 0]^T \\ v_2^0 &= [250, 0]^T, v_2^1 = [250, 0]^T, v_2^2 = [600, -500]^T \\ v_3^0 &= [125, 125]^T, v_3^1 = [125, 125]^T, v_3^2 = [125, 125]^T \end{aligned}$$
- Sampling time-point:  $t_0 = 0$  s,  $t_1 = 3$  s,  $t_2 = 6$  s.
- Control loop: 10 Hz

The time unit is second and the length unit is micrometer.

As shown in Fig. 10, in the experiment the robot labeled as *R* is desired to plan a length-optimal path that navigates through three dynamic obstacles and three static obstacles within the given maneuver time  $T = 9$  s. The straight lines on the floor denote the X and Y-coordinate axes, and the moving tendency of the robot and the three dynamic obstacles are also given in the figure. When any robot obstacles are detected at sampling time-point, the newest information is sent to robot

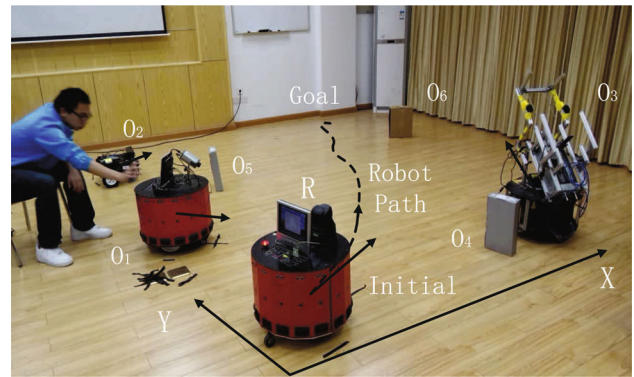


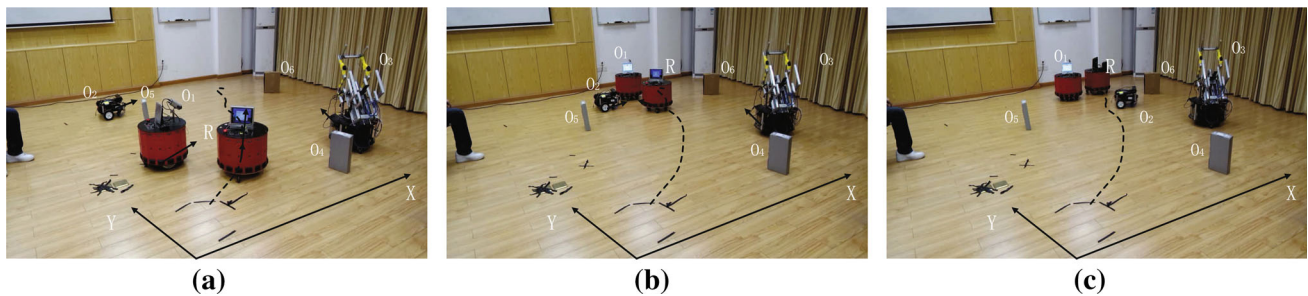
Fig. 10 Experiment scenario

*R* so as to update its path. Fig. 11 illustrates the mobile robot moving in the experiment.

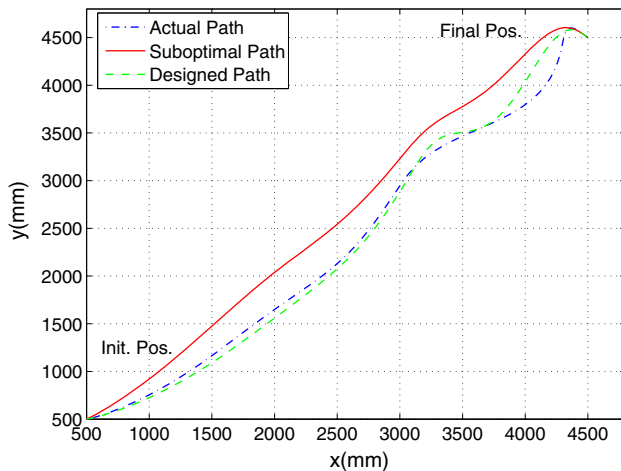
At the beginning, the robot computed an optimal solution such as  $c_6^* = -0.05$  and  $d_6^* = 0.05$  in order to render a shortest path to the goal position. However, on this condition the robot could collide with obstacle  $O_1$  at around 3s according to the current sensed information, and hence it adjusted the solution to an alternative suboptimal one,  $c_6^1 = -0.08$  and  $d_6^1 = 0.08$ , while keeping away from  $O_1$  and the static obstacle  $O_4$  sensed at 0s. When it came to 3s, the robot sensed the environment again and detected the motion change of obstacle  $O_1$  as well as the appearance of static obstacle  $O_5$  and dynamic obstacle  $O_3$ . According to the updated motion information of  $O_1$  and estimation, the robot would run into  $O_1$  at around 4.5s, making the robot change the parameters again from the inherited ones  $c_6^2 = -0.08$  and  $d_6^2 = 0.08$  to  $c_6^2 = 0.3$  and  $d_6^2 = 1.2$ . Likewise, at the final sampling time-point 6s when dynamic obstacle  $O_1$  stopped suddenly and a new dynamic obstacle  $O_2$  rushed to the robot, it had to replan the parameters from the ones reset at the last sampling moment to the new ones,  $c_6^3 = 0.4$  and  $d_6^3 = 67$  for the rest of the path, thus making the robot take a relative sharp turn to avoid  $O_2$  and subsequently run a bigger detour to get away from  $O_1$ , while keeping a long distance from another newly detected static obstacle  $O_6$  and reaching the goal state successfully and safely.

In order to compare the experiment performance with the desired ones obtained from simulation, Fig. 12 plotted three trajectories that are rendered by the corresponding simulation and the experiment. The lengths of the three paths are 6,567 mm (suboptimal), 6,678 mm (designed) and approximately 6,700 mm (experiment). The desired velocity profile and the sampled velocity for the robot in simulation and the experiment are shown in Fig. 13, both of which satisfy the kinodynamic constraints (velocity/acceleration bounds). Furthermore, in this real-world experiment of our method, the maximum running time for trajectory updating at each sampling time-point is about 0.2 s, including the computation time for solving the constrained optimization problem

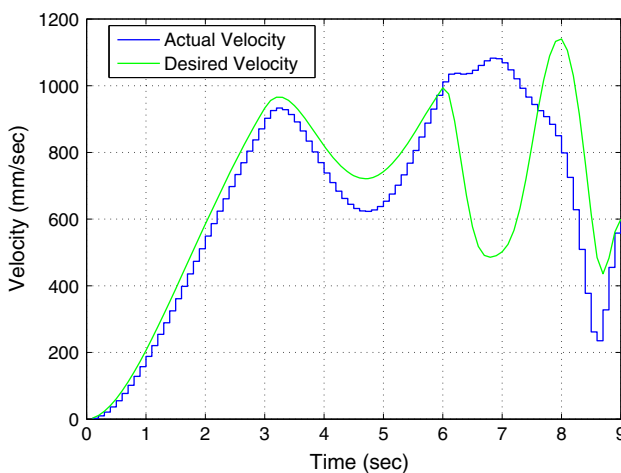




**Fig. 11** Robot in experiment. **a** 3 s **b** 6 s **c** 9 s



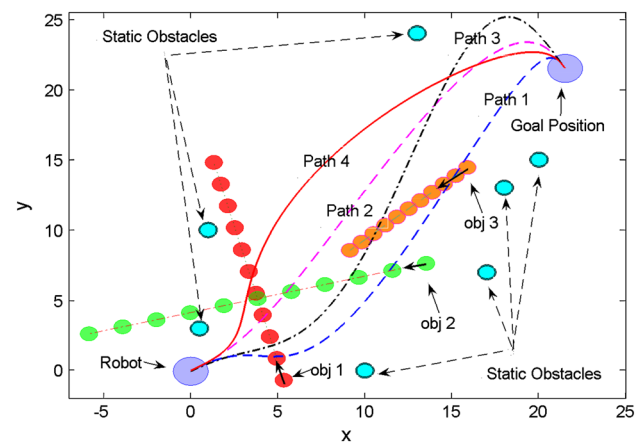
**Fig. 12** Robot actual moving path, designed path, and suboptimal path without consideration of obstacles



**Fig. 13** Velocity profiles of the robot in experiment and in simulation

(50) using SQP or analytical solutions online. It is seen from the figures and data that the proposed method can work well in real-world implementation while maintaining good efficiency for on-line computation.

(3) *Example 4 (suboptimal solutions in highly cluttered environments)*: This part of example addresses the trajectory



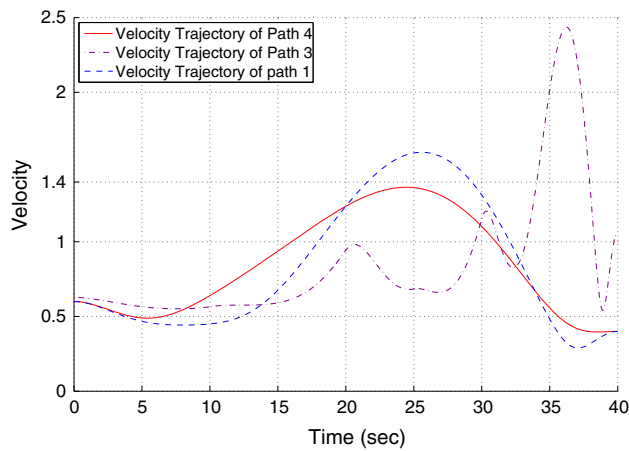
**Fig. 14** Robot trajectories in highly cluttered environment. Path 1 and 2 are generated by length optimal solution and 'weighed optimization' in our method respectively. Path 3 are generated by length optimal solutions in Yang et al. (2010). Path 4 is the adjusted trajectory by our method under new kinodynamic constraints  $v_{max} = 1.4$ ,  $a_{max} = 0.3$

generation with guaranteed combined performance under strict kinodynamic constraints in cluttered environments containing both static and dynamic obstacles. Redefined boundary conditions of the robot are  $q_0^* = (0, 0, 0.53, 0, 0.6, 0)$  and  $q_f^* = (21.5, 21.5, -1.04, 0, 0.4, 0)$  respectively. As shown in Fig. 14, there are seven static disc obstacles of radius 0.5 (denoted by cyan circles with black edge). Obj1, obj2 and obj3 are three moving obstacles of radius 0.5, and they are denoted by red, green and orange circles respectively. The initial kinodynamic constraints are  $v_{max} = 2.5$ ,  $a_{max} = 0.3$ . In Fig. 14 the maneuver time is 40s, and each circles of moving obstacles represent their temporal positions of every 4 s, with arrow at initial positions indicating their respective moving direction.

As shown in Fig. 14, compared to path 3, path 1 and 2 that generated under our method are more aggressive but still safe in the highly cluttered environment, with obvious improvement on path length.

Next, we consider combined optimization problem with weights  $\omega_1 = 0.5$ ,  $\omega_2 = 0.5$ . Path 1 is the initial trajectory created according to length-optimal objective in (12), of which the energy consumption and length are 3,583.0 and



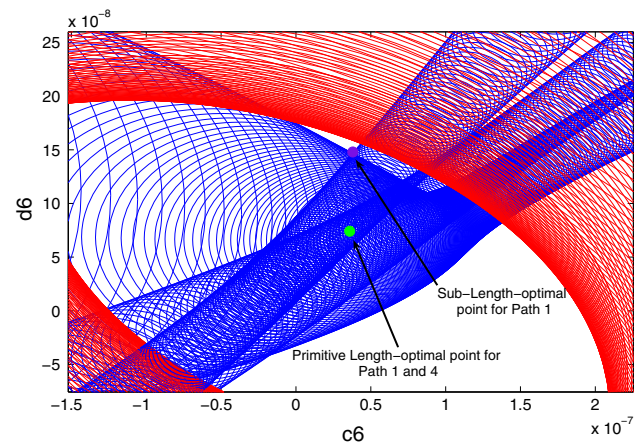


**Fig. 15** Robot velocity trajectories of the paths in Fig. 14. Blue dashed curve and magenta dash-dotted curve represent trajectories under original velocity limitation of 2.5. Red solid curve represents trajectory under new velocity limitation of 1.4 (Color figure online)

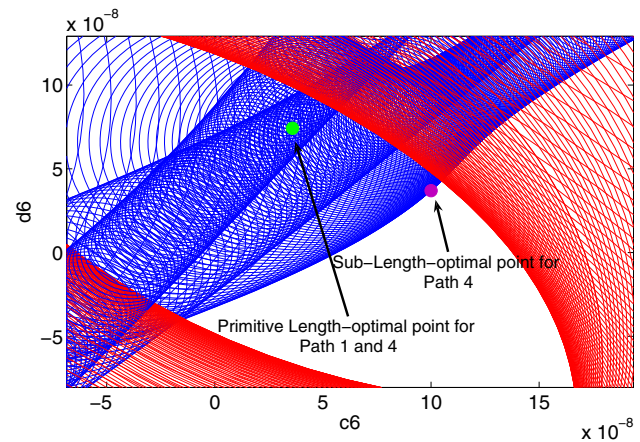
33.51. By specifying  $\omega_1 = 0.5$ ,  $\omega_2 = 0.5$  into the combined performance index (14), path 2 is rendered with updated optimal performance of 3,397.7/33.87 in energy consumption and path length. Since the performance evaluation on energy and length are more balanced, compared to path 1, the length of path 2 is sacrificed a bit for the improvement on energy consumption.

Besides, to highlight flexibility of our method, new kinodynamic constraints are employed in this example (taking new velocity limitation for instance). With the same environment in Fig. 14, velocity constraint of robot is changed from  $v_{max} = 2.5$  to  $v_{max} = 1.4$ , making it impossible for robot to easily outrun the moving obstacles. The initial velocity trajectory of path 1 and 3 in Fig. 14 are shown in Fig. 15. Although robot motions on both of the two trajectories satisfy the original limitation of 2.5, they violate the new one. In order to replan a feasible path under such updated constraints using our method, we investigate the selection of new parameter points on parameter space in Figs. 16 and 17. It is noted that the chosen suboptimal point in Fig. 16 is no longer covered by the solution area (restrained area) in Fig. 17, which means the previous trajectory will not satisfy the present limitation. So, we choose an alternative suboptimal point in Fig. 17 such that the solution space wholly cover the selected point. The updated velocity trajectory of such path is shown by red solid curve in Fig. 15, and updated path is given by red solid curve in Fig. 14. These results show that feasible solutions can also be found by our method to plan a qualified path in front of updated kinodynamic constraints.

**Remark 10** It should be noted that although data of the energy consumption and path length in Table 1 and 2 don't seem to diverge much in absolute value, the percentage of those difference could still reveal the effects. In real world



**Fig. 16** Solution space in the parameter space under initial velocity limitation of 2.5. Red circle sets represent corresponding feasible areas mapped from analytic inequations (4). Blue circle sets represent collision areas with static and dynamic obstacles, which is mapped from (39). Magenta point is the chosen suboptimal solution of trajectory parameters  $c_6$  and  $d_6$  (Color figure online)

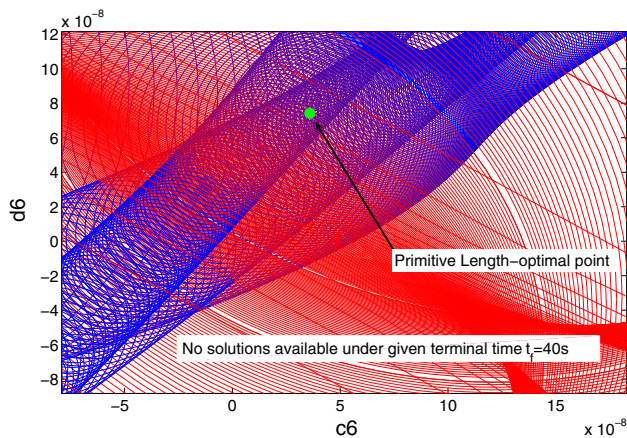


**Fig. 17** Solution space in the parameter space under updated velocity limitation of 1.4. Red circle sets represent corresponding feasible areas mapped from analytic inequations (4). Blue circle sets represent collision areas with static and dynamic obstacles, which is mapped from (39). Magenta point is the re-chosen suboptimal solution of trajectory parameters  $c_6$  and  $d_6$  (Color figure online)

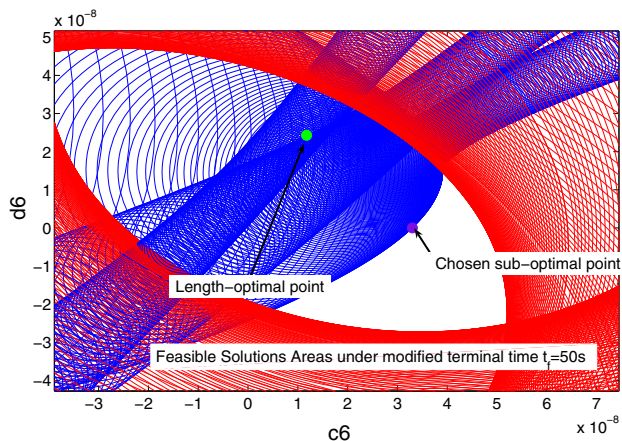
applications, the unit used could always be much larger as kilometer of length, rather than standard meter in this simulation section, which can better prove the excellence of the proposed method.  $\square$

### 5.3 Adjustment towards non-solution situation

This section considers trajectory re-generation towards non-solution situation to improve the completeness of our method. Since solution space in parameter space such as those in Figs. 16 and 17 may not always exist, what happens when the robot cannot physically find a feasible solution? As we have discussed in previous part, proper adjustment of termi-



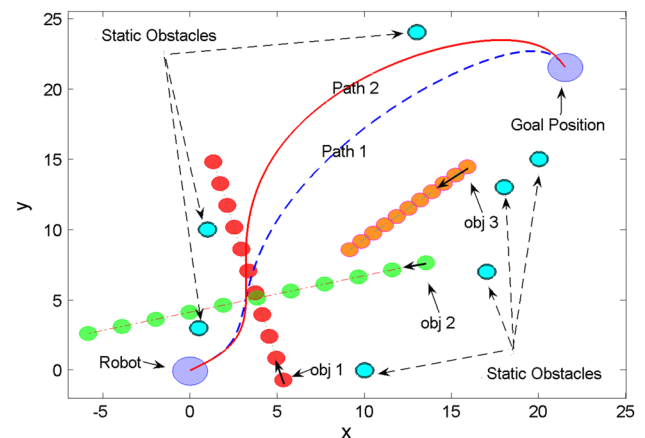
**Fig. 18** Solution space in the parameter space becomes null under strict velocity limitation of 1.2 with original terminal time  $t_f=40$ s



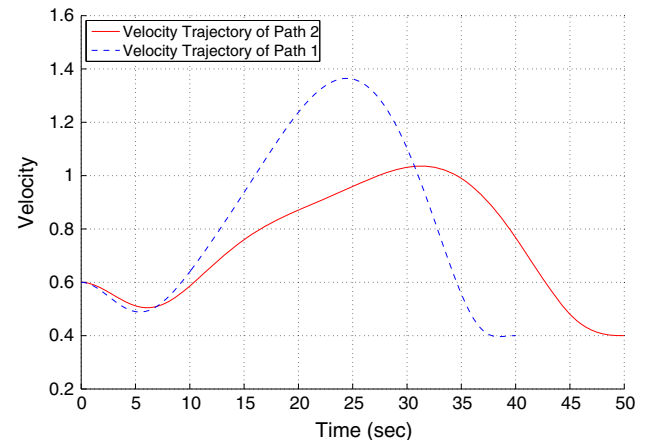
**Fig. 19** Solution space in the parameter space reappears after prolonging the terminal time  $t_f$  from 40 to 50 s

nal time can be employed in our method to find feasible solutions unless all possible passages are blocked by obstacles. This case is investigated by further limiting the velocity constraint in *Example 4* from 1.4 to 1.2, with the same settings of other environment parameters in Fig. 14. Due to such strictly imposed constraint, the initial solution space in Fig. 17 cease to exist, as shown in Fig. 18.

Since the non-solution situation in this case is caused only by strict velocity constraint, we prolong the terminal time  $t_f$  to 50s. It is reasonable in real-world to extend maneuver time for low speed navigation of mobile robots. Then, it is noted that the feasible solution space empty in Fig. 18 reappears in Fig. 19. The following selection of parameter points is simple as ordinary suboptimal process of our method. As the magenta point chosen in Fig. 19, a feasible suboptimal point is found to generate new trajectories. The comparison of paths and velocity trajectories before and after such adjustment are shown in Figs. 20 and 21. After adjustment, robot can still generate a feasible collision-free path to achieve the



**Fig. 20** Path comparison in cluttered environment. Blue dashed path 1 is generated under original terminal time  $t_f=40$ s. Red solid path 2 is generated under adjusted terminal time  $t_f=50$ s (Color figure online)



**Fig. 21** Velocity trajectories comparison of paths in Fig. 20. Blue dashed curve corresponds to path 1 under original terminal time. Red solid curve corresponds to path 2 under adjusted terminal time (Color figure online)

original goal, while well satisfying the given strict kinodynamic constraints. This is an example where non-solution situation appears, our approach could still try to find an alternative way to complete the tasks. It should be noted that the solvable or unsolvable situations in Figs. 19 and 18 are found before executing, and robot can determine whether to adjust terminal time in advance to guarantee the feasibility of trajectories once generated, which is good for on-line planning. Nonetheless, as discussed in the part of *Algorithm Completeness and Stability*, it cannot be ensured that the solutions will always be found under all circumstances.

## 6 Conclusion

This paper presents a novel analytical method to solve combined optimization trajectory generation problem for mobile

robots with kinodynamic constraints in the presence of moving obstacles. Analytical suboptimal solutions that integrate two different weighed optimal performance indices can flexibly satisfy various optimal requirements. By explicitly considering redefined higher dimensional boundary conditions as well as robot kinodynamic constraints and collision avoidance criteria, the constrained optimal trajectory generation problem was recast as a constrained minimum distance problem on a 2D parameter space with straightforward geometric implications that can be well handled by simple on-line computation with analytical results or SQP method. The analytical suboptimal solutions could automatically detect unsolvable situations and make remedies to make the problem solvable. Simulation and experiment results have verified the efficiency and effectiveness of the proposed method in dealing with local dynamic environments. Further works include an extension of the proposed approach to other global search-based algorithm in order to deal with highly dense 2D/3D environment and solve the issues as lack of completeness and stability. Another issue is to further improve the rigorousness of the method by quantify the exact amount of time to be extended in order for the robot to retain feasible paths.

**Acknowledgments** The authors would like to acknowledge that this work was partially supported by the National Natural Science Foundation of China (Grant No. 61379111, 61402538 and 61202342) and Specialized Research Fund for the Doctoral Program of Higher Education (20110162110042).

## References

- Balkcom, D. J., & Mason, M. T. (2002). Time optimal trajectories for bounded velocity differential drive vehicles. *The International Journal of Robotics Research (IJRR)*, 21, 199–217.
- Barraquand, J., & Latombe, J. C. (1991). Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3, 2328–2335.
- Bhattacharya, S., Likhachev, M., & Kumar, V. (2012). Topological constraints in search-based robot path planning. *Autonomous Robots*, 33(3), 273–290.
- Bicchi, A., Casalino, G., & Santilli, C. (1995). Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2, 1349–1354.
- Bonnans, J.-F., et al. (2006). *Numerical optimization: Theoretical and practical aspects*. Heidelberg: Springer.
- Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7, 278–288.
- Cheng, P., Frazzoli, E., & LaValle, S. (2008). Improving the performance of sampling-based motion planning with symmetry-based gap reduction. *IEEE Transactions on Robotics*, 24, 488–494.
- Divelbiss, A. W., & Wen, J. T. (1997). A path space approach to nonholonomic motion planning in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 13, 443–451.
- Dong, W., & Guo, Y. (2005). New trajectory generation methods for nonholonomic mobile robots. *Proceedings of the international symposium on collaborative technologies and systems*, pp. 353–358.
- Duleba, I., & Sasiadek, J. Z. (2003). Nonholonomic motion planning based on Newton algorithm with energy optimization. *IEEE Transactions on Control Systems Technology*, 11, 355–363.
- Fiorini, P., & Shiller, Z. (1993). Motion planning in dynamic environments using the relative velocity paradigm. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1, 560–565.
- Fliess, M., LeVine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: Introductory theory and examples. *International Journal of Control*, 61, 1327–1361.
- Guo, Y., & Tang, T. (2008). Optimal trajectory generation for nonholonomic robots in dynamic environments. *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 2552–2557.
- Guo, Y., Li, Y., & Sheng, W. (2007). Global trajectory generation for nonholonomic robots in dynamic environments. *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 1324–1329.
- Hashim, S., & Tien-Fu, L. (2009). A new strategy in dynamic time-dependent motion planning for nonholonomic mobile robots. *IEEE international conference on robotics and biomimetics (ROBIO)*, pp. 1692–1697.
- Howard, T. M., & Kelly, A. (2007). Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research (IJRR)*, 26(2), 141–166.
- John, S., et al. (2014). Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9), 1251–1270.
- Karaman, S., & Frazzoli, E. (2010). Optimal kinodynamic motion planning using incremental sampling-based methods. In: *49th IEEE conference on decision and control (CDC)*, pp. 7681–7687.
- Kim, J. O., & Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8, 338–349.
- Kindel, R., Hsu, D., Latombe, J. C., & Rock, S. (2000). Kinodynamic motion planning amidst moving obstacles. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1, 537–543.
- Koenig, Sven, & Likhachev, Maxim. (2002). D\* Lite. *Proceedings of the eighteenth national conference on artificial intelligence (AAAI)*, pp. 476–483.
- Kushleyev, A., & Likhachev, M. (2009). Time-bounded lattice for efficient planning in dynamic environments. *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 1662–1668.
- Ladd, A. M., & Kavraki, L. E. (2004). Measure theoretic analysis of probabilistic path planning. *IEEE Transactions on Robotics and Automation*, 20, 229–242.
- LaValle, S. M., & Kuffner, J. J., Jr. (1999). Randomized kinodynamic planning. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1, 473–479.
- LaValle, S. M. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.
- Li, Z. X., & Canny, J. (1992). *Nonholonomic motion planning*. Dordrecht. The Netherlands: Kluwer.
- Likhachev, M., & Ferguson, D. (2009). Planning long dynamically-feasible maneuvers for autonomous vehicles. *International Journal of Robotics Research (IJRR)*, 28(8), 933–945.
- Liu, S., & Sun, D. (2011). Optimal motion planning of a mobile robot with minimum energy consumption. *IEEE/ASME international conference on advanced intelligent mechatronics (AIM)*, pp. 43–48.



- De Luca, A., Oriolo, G., & Samson, C. (1998). Feedback control of a nonholonomic car-like robot. *Robot Motion Planning and Control*, 229, 171–253.
- Lvine, J. (2009). *Analysis and control of nonlinear systems: A flatness-based approach*. Berlin: Springer.
- Morales, J. L., Nocedal, J., Wu, Y. (2011). A sequential quadratic programming algorithm with an additional equality constrained phase. *IMA Journal of Numerical Analysis*, drq037.
- Murray, R. M., & Sastry, S. S. (1993). Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38, 700–716.
- Narayanan, V., Phillips, M., & Likhachev, M. (2012). Anytime safe interval path planning for dynamic environments. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4708–4715.
- Pivtoraiko, M., Knepper, R. A., & Kelly, A. (2009). Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics (JFR)*, 26(3), 308–333.
- Qu, Z., Wang, J., & Plaisted, C. E. (2004). A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, 20, 978–993.
- Reister, D. B., & Pin, F. G. (1994). Time-optimal trajectories for mobile robots with two independently driven wheels. *The International Journal of Robotics Research (IJRR)*, 13, 38–54.
- Shiller, Z., Gal, O., & Raz, A. (2011). Adaptive time horizon for on-line avoidance in dynamic environments. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 3539–3544.
- Shiller, Z., Large, F., & Sekhavat, S. (2001). Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 4, 3716–3721.
- Tilbury, D., & Murray, R. M. (1995). Trajectory generation for the N-trailer problem using Goursat normal form. *IEEE Transactions on Automatic Control*, 40, 802–819.
- Ucan, X. I., & Kavraki, L. E. (2012). A sampling-based tree planner for systems with complex dynamics. *IEEE Transactions on Robotics*, 28, 116–131.
- van den Berg, J., & Overmars, M. (2007). Kinodynamic motion planning on roadmaps in dynamic environments. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4253–4258.
- Wu, A., & How, J. P. (2012). Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles. *Autonomous Robots*, 32(3), 227–242.
- Yang, J., Daoui, A., Qu, Z., Wang, J., & Hull, R. A. (2005). An optimal and real-time solution to parameterized mobile robot trajectories in the presence of moving obstacles. *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 4412–4417.
- Yang, J., Qu, Z., Wang, J., & Conrad, K. (2010). Comparison of optimal solutions to real-time path planning for a mobile vehicle. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40, 721–731.
- Yuan, H., & Shim, T. (2011). Model based real-time collision-free motion planning for mobile robots in unknown dynamic environments. In: *14th international IEEE conference on intelligent transportation systems (ITSC)*, pp. 416–421.



Jun Peng is a Professor in School of Information Science & Engineering at the Central South University, China. Her research interests include cooperative control; Multi-agent systems and wireless communications. She received her B.S. degree from Xiangtan University and the MSc degree from National University of Defense Technology, China, in 1987 and 1990, respectively. She received her PhD degree from Central South University in 2005. In April, 1990, she joined the staff of Central South University. From 2006 to 2007, she was with School of Electrical and Computer Science of University of Central Florida, USA, as a visiting scholar. She is a Member of IEEE.



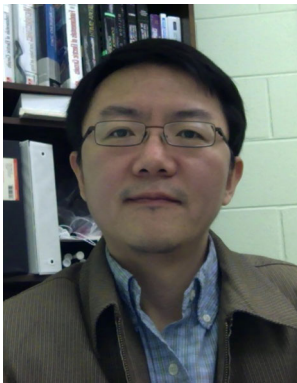
Wenhao Luo received the B.Eng degree in Measurement & Control Technology and Instruments from Central South University, China, in 2012. He is currently pursuing the M. S. Degree in Robotics at Robotics Institute, Carnegie Mellon University, USA where he is also a research assistant. Before that, he was a research assistant and Ph.D. candidate in Control Science and Engineering in the Department of Automation at Shanghai Jiao Tong University, China, from 2012 to 2014. His research interests are in the areas of motion planning, multi-agent systems and control theory.



Weirong Liu received his B.S., M.S degrees in the School of Information Science and Engineering, Central South University, and received his Ph.D. degrees in the Laboratory of Complex Systems & Intelligence Science, Institute of Automation, Chinese Academy of Sciences. Currently, he is an associate professor of the School of Information Science and Engineering, Central South University. His special fields of interest include robot control, cooperative control, networked system control and multi-agent system.



**Wentao Yu** received the B.S degree and M.E degree from Central South University, Changsha, China in 2004 and 2008 respectively. She is currently pursuing the Ph.D. degree in School of Information Science and Engineering, Central South University. Her research focuses on cooperative navigation of autonomous vehicles. Specifically she is investigating cooperative path planning strategies for a team of autonomous vehicles under dynamic constraints.



**Jing Wang** received the Ph.D. in control theory and control engineering from Central South University, China, in 1997. He was a Postdoctoral Research Fellow at the Institute of Computing Technology, Chinese Academy of Sciences, from 1997 to 1999, and at Department of Electrical and Computer Engineering, National University of Singapore, from 1999 to 2002. From 2002 to 2007, He was with School of Electrical Engineering and Computer Science of University of Central Florida as a Research Assistant Professor. From 2007 to 2014, he was with the Department of Computer Engineering, Bethune-Cookman University, first as an Assistant Professor and then Associate Professor. Since August 2014, he joined in the Department of Electrical and Computer Engineering, Bradley University, Peoria, IL, as an Assistant Professor. He is a Senior Member of IEEE.