

Trajectory Optimization for 3D Shape-Changing Robots with Differential Mobile Base

Mengke Zhang^{1,2}, Chao Xu¹, Fei Gao¹, Yanjun Cao¹

Abstract—Service robots have attracted extensive attention due to specially designed functions, such as mobile manipulators or robots with extra structures. For robots that have changing shapes, autonomous navigation in the real world presents new challenges. In this paper, we propose a trajectory optimization method for differential-drive mobile robots with controllable changing shapes in dense 3D environments. We model the whole-body trajectory as a polynomial trajectory that satisfies the nonholonomic dynamics of the base and dynamics of the extra joints. These constraints are converted into soft constraints, and an activation function for dense sampling is applied to avoid nonlinear mutations. In addition, we guarantee the safety of full shape by limiting the system’s distance from obstacles. To comprehensively simulate a large extent of height and width changes, we designed a novel Shape-Changing Robot with a Differential Base (SCR-DB). Our global trajectory optimization gives a smooth and collision-free trajectory for SCR-DB at a low computational cost. We present enormous simulations and real-world experiments to validate our performance, including coupled whole-body and independent differential-driven vehicle motion planning.

I. INTRODUCTION

Service robots are expected to help and assist humans in various fields. Recently, the application of robots in factories or households has increased rapidly thanks to the advanced development of mobile platforms as well as robot manipulators or specialized functional structures, such as Stretch [1] and Picker [2]. The functional structures usually change the shapes of robots while performing tasks. These changing shapes can cause extra constraints for the navigation of mobile base, especially in tight and cluttered 3D environments.

Normally, robots are programmed to move in a minimal setup with the smallest shapes while moving to keep the safest solution and to reduce the complexity of control. However, this usually limits the usage and lacks efficiency in many tasks, such as moving only through absolute safe pathways after the obstacle inflation. In this paper, we target to solve the trajectory planning for general robots with changing shapes in a complex 3D environment, especially for systems using the differential mobile base, which is most widely used for service robots. An optimal trajectory for the systems allows the simultaneous control of the mobile base

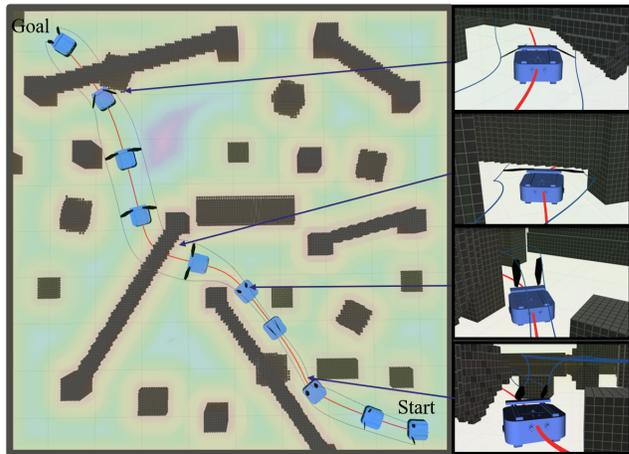


Fig. 1. Simulation experiments in the environment containing bridges and square obstacles (left). Both the base and the SC-DOF have smooth trajectories and can achieve obstacle avoidance (right).

and the shape-changing mechanism actuator to move safely and smoothly.

Taking a Shape-Changing Robot with Differential Base (SCR-DB), for example, there are two main challenges for its trajectory planning. Firstly, the safety constraint imposed by the changing shapes while moving requires the algorithm to plan all the actuators simultaneously as a whole system. Separating the mobile base from the Shape-Changing Degree-Of-Freedom (SC-DOF) and controlling them sequentially is time inefficiency. Furthermore, taking static inflation for the robot or environment can be spacial inefficiency and falsely ignore some possible paths. Secondly, the robot’s kinodynamic feasibility should be fully considered. As the maximum angular velocity is coupled with the linear velocity for differential wheeled control, the driving torque should be considered for the dynamics to get an easy-to-execute trajectory. What’s more, when solving the above difficulties, the computing time should also be short of supporting real-time deployment, which places additional challenges on the algorithm.

In this paper, we propose a novel optimization-based real-time spatial-temporal planning solution to get a smooth trajectory that satisfies the above requirements. For the obstacle avoidance constraint, a euclidean signed distance field (ESDF) map is formulated as inequality constraints to ensure the distance between the robot and obstacles to ensure safety. We apply the differential flatness property of the mobile base to represent the robot’s states and dynamical feasibility in a continuously differentiable model. The kinematic properties constraints are formulated as inequalities. The SC-DOF is

*This work was supported by National Natural Science Foundation of China under Grant 62103368 and funding from Yunjing Intelligent (Shenzhen) Co. Ltd.

¹The State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and Huzhou Institute, Zhejiang University, Huzhou 313000, China.

²Yunjing Intelligent (Shenzhen) Co. Ltd, Shenzhen, 518000, China.

Email:{mkzhang233, cxu, fgaoaa, yanjunhi}@zju.edu.cn
Corresponding author: Yanjun Cao, Fei Gao.

designed as an optimization variable coupled in the optimization considering the collision between the whole-body shape and the environment. Then we use the penalty function to transform the constrained problem into an unconstrained problem and solve it by the quasi-Newton method. For constraints with strong nonlinearity, a novel dense sampling controlled by the activation function is designed to avoid pathological trajectory.

The main contributions of this paper are:

1. We propose a optimization-based real-time spatial-temporal planning method considering changing shapes for SCR-DB in the 3D environment. We formulate a full-state model based on differential flatness and the joint angle.
2. We transform the optimization model for SCR-DB with constraints into an unconstrained problem and retain the differential base's motion performance.
3. We combine the shape-changing motion with dynamics and 3D environment collision constraints to ensure the feasibility and safety of the whole system.
4. We validate the effectiveness and robustness of our method both in simulation and real experiments.

II. RELATED WORK

Motion planning is one of the most active topics in robotics. Various methods have been proposed for mobile robots or robot arms, with two main categories sampling-based [3]–[5] and optimization-based [6], [7]. Although sampling-based methods are widely used as they are effective and probabilistically complete, the computation and trajectory quality limit their performance. Optimization-based methods can flexibly include various constraints into a general nonlinear optimization. CHOMP [6] used functional gradient techniques to improve the trajectory quality interactively. In contrast, STOMP [7] improves the trajectory through stochastic gradient methods. These algorithms have been evaluated in high-dimensional robot arms but are still hard to be online planning.

The scenario relative to our situation is the motion planning for the mobile manipulator. However, we focus more on the planning of the mobility instead of end effector tasks such as pick-and-place [5], [8]–[10]. Namely, the whole body planning of the system to move in a challenging environment can happen for mobile manipulator motion in a large and clustered environment. These planning algorithms [5], [8]–[10] decouple the planning for the arm and mobile base, which is less efficient. Recent studies [11], [12] design optimization-based planners to plan the motion of the manipulator and the base simultaneously. As their models have not considered the higher order of dynamics of the system, trajectory quality for actuators can not be guaranteed.

Although we target high-quality motion planning for SCR-DB in the 3D dense environment, some studies about planning for vehicles in 2D cases are worth mentioning. Works [13], [14] propose path optimization by interpolating and optimizing key waypoints, but without taking temporal information into consideration. Rosmann et al. [15], [16] took time into account and presented "timed elastic band"

(TEB) based hyper-graph and gave the implementation with G2o. However, it may take a long time to generate an optimal trajectory. Moreover, trajectory optimization by optimal control is also a feasible approach. [17] proposed a real-time minimum-time trajectory planning that considered actuator limits. Recently, Han et al. [18] parameterized the Ackerman model with differential flatness to acquire high efficiency to constrain nonholonomic dynamics.

In summary, our novelty lies in coupled motion planning for the differential-driven mobile base and controllable structure using differential flatness output in a dense 3D environment, considering higher order dynamics. At the same time, the proposed method is efficient in meeting the requirements of real-time.

III. MODELING OF THE SCR-DB

This section will introduce our approach to modeling the SCR-DB and optimization variables.

A. SCR-DB

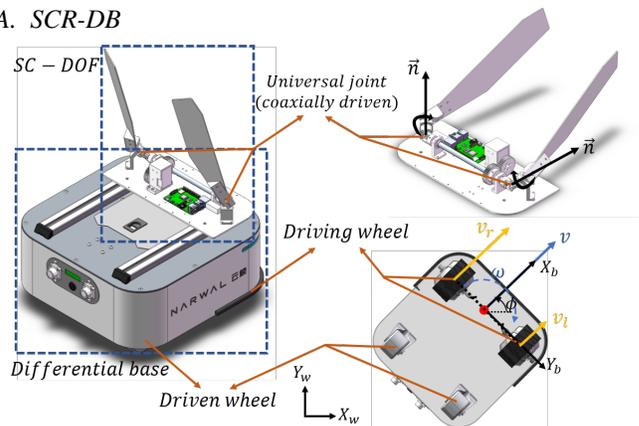


Fig. 2. Model of our SCR-DB.

To comprehensively simulate a large extent of height and shape changes, we designed a novel SCR-DB shown in Fig. 2. SCR-DB has a differential drive mobile base, and two front wheels are driving wheels. A pair of blades are placed on the base and controlled through coaxial-driven universal joints, with a tilted rotation axis to achieve large-degree shape changes. We show three special states in Fig. 3 to illustrate the collision constraint through corresponding obstacles with different joint angles. This SCR-DB can simulate a dual-arm robot by simplifying its arm to a single controllable joint.

We assume that the origin of the base coordinate frame is the projection point of the center of the driving wheel on the ground, and the direction of the X-axis is the forward direction. The robot coordinate frame coincides with it. We use the left superscript to indicate the coordinate system where vectors locate in. For example, ${}^w p$, ${}^b p$, ${}^j p$ represents coordinates of p in the world, base and joint coordinate frames respectively.

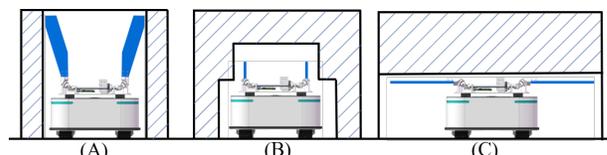


Fig. 3. Three special cases of SCR-DB for different obstacles.

B. Differential base

Let the state variables be the state of the center of the driving wheels: ${}^w\mathbf{p} = [p_x, p_y, \phi]^T \in \mathbb{R}^3$. The base is controlled by the wheel speed $\mathbf{u} = [v_r, v_l]^T \in \mathbb{R}^2$, where v_r and v_l are the speed of the right wheel and the left wheel. Thus the kinematics of the base can be expressed as:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{\cos \phi}{2} & \frac{\cos \phi}{2} \\ \frac{\sin \phi}{2} & \frac{\sin \phi}{2} \\ \frac{1}{d_{wb}} & -\frac{1}{d_{wb}} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}, \quad (1)$$

$$\phi = \arctan 2(\dot{p}_y, \dot{p}_x), \quad (2)$$

where d_{wb} is the wheelbase of the driving wheels. Eq.(2) implies nonholonomic dynamics. We choose the flat output $\boldsymbol{\sigma} := [\sigma_x, \sigma_y]^T \in \mathbb{R}^2$, where $\boldsymbol{\sigma}$ is the position of the center of the driving wheel, i.e. $\boldsymbol{\sigma} = [p_x, p_y]^T$. Then we can derive other dynamical states from Eq.(1) as:

$$v = \sqrt{\dot{\sigma}_x^2 + \dot{\sigma}_y^2}, \quad (3a)$$

$$a_l = (\dot{\sigma}_x \ddot{\sigma}_x + \dot{\sigma}_y \ddot{\sigma}_y) / \sqrt{\dot{\sigma}_x^2 + \dot{\sigma}_y^2}, \quad (3b)$$

$$\omega = (\dot{\sigma}_x \ddot{\sigma}_y - \dot{\sigma}_y \ddot{\sigma}_x) / (\dot{\sigma}_x^2 + \dot{\sigma}_y^2), \quad (3c)$$

$$\alpha = \frac{\dot{\sigma}_x \sigma_y^{(3)} - \dot{\sigma}_y \sigma_x^{(3)}}{\dot{\sigma}_x^2 + \dot{\sigma}_y^2} - \frac{2(\dot{\sigma}_x \ddot{\sigma}_y - \dot{\sigma}_y \ddot{\sigma}_x)(\dot{\sigma}_x \ddot{\sigma}_y + \dot{\sigma}_y \ddot{\sigma}_x)}{(\dot{\sigma}_x^2 + \dot{\sigma}_y^2)^2}, \quad (3d)$$

where v is base's velocity, a_l is the linear acceleration, ω is the angular velocity, and α is the angular acceleration.

C. Shape-changing mechanism

The conversion between coordinate frames can be achieved by forward kinematics with the joint angle. Therefore we use $\theta \in R$ to represent the joint angle. The rotation matrix ${}^b\mathbf{R}_j(\theta)$ can be calculated by the rotation axis and θ . Thus for any point ${}^j\mathbf{h}_m \in \mathbb{R}^3$ of the manipulator, we can find its corresponding coordinate ${}^b\mathbf{p}_m$ as:

$${}^b\mathbf{p}_m = {}^b\mathbf{R}_j(\theta) {}^j\mathbf{h}_m + {}^b\mathbf{p}_{m0}, \quad (4)$$

where ${}^b\mathbf{p}_{m0}$ is the coordinate of the joint in the base coordinate frame.

In summary, we choose the optimization variables to be the flat output of the base and the joint angle, i.e.

$$\mathbf{z} = [\boldsymbol{\sigma}^T, \theta]^T \in \mathbb{R}^3. \quad (5)$$

Therefore, all the control and state variables of the system can be obtained by \mathbf{z} and their derivatives.

IV. SPATIAL-TEMPORAL OPTIMIZATION FOR SCR-DB

In this section, we give a spatial-temporal optimization formulation for trajectory planning and transform the problem into an unconstrained optimization problem.

A. Optimization problem construction

We divide the trajectory into M segments. Each segment is expressed as a 3-dimensional polynomial with degree $2s-1$. With the coefficient matrix $\mathbf{c} = [\mathbf{c}_1^T, \dots, \mathbf{c}_M^T]^T \in \mathbb{R}^{2Ms \times 3}$, the i -th segment of the trajectory can be expressed as:

$$\begin{aligned} \mathbf{z}_i(t) &= \mathbf{c}_i^T \boldsymbol{\beta}(t), \quad \boldsymbol{\beta}(t) = [1, t, \dots, t^{2s-1}]^T, \\ \forall t \in [0, T_i], \forall i \in \{1, 2, \dots, M\}. \end{aligned} \quad (6)$$

The minimal control effort problem with time regularization can be expressed as a nonlinear constrained optimization:

$$\min_{\mathbf{c}(t), \mathbf{T}} J_0 = \int_0^{T_s} \boldsymbol{\psi}(t)^T \mathbf{W} \boldsymbol{\psi}(t) dt + \omega_T T_s, \quad (7a)$$

$$s.t. \boldsymbol{\psi}(t) = \mathbf{z}^{(s)}(t), \forall t \in [0, T_s], \quad (7b)$$

$$\mathbf{z}_1^{[s-1]}(0) = \bar{\mathbf{z}}_0, \mathbf{z}_M^{[s-1]}(T_s) = \bar{\mathbf{z}}_f, \quad (7c)$$

$$\mathbf{z}_i^{[s-1]}(T_i) = \mathbf{z}_{i+1}^{[s-1]}(0), \quad (7d)$$

$$T_s = \sum_{i=1}^M T_i, T_i > 0, \quad (7e)$$

$$\mathcal{C}_{d_b}(\boldsymbol{\sigma}(t), \dots, \boldsymbol{\sigma}^{(s)}(t)) \leq 0, \forall d_b \in \mathcal{D}_b, \forall t \in [0, T_s], \quad (7f)$$

$$\mathcal{C}_{d_m}(\mathbf{z}(t), \dots, \mathbf{z}^{(s)}(t)) \leq 0, \forall d_m \in \mathcal{D}_m, \forall t \in [0, T_s], \quad (7g)$$

where $\mathbf{W} \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix to penalize control efforts, $\omega_T T_s$ is the time regularization term. The constraints can be imposed on two parts, where $\mathcal{D}_b = \{d_b : v, \bar{v}, a_l, \omega, \alpha, s\}$ is the constraint of the differential base and $\mathcal{D}_m = \{d_m : \omega_m, \alpha_m, s\}$ is the constraint of the joint, both contain kinematic constraints and safety constraint (represented by s).

Inspired by trajectory class Σ_{MINCO} [19], polynomial coefficients are uniquely determined by waypoints $\mathbf{z}'_w = [\mathbf{z}'_1, \dots, \mathbf{z}'_{M-1}] \in \mathbb{R}^{3 \times (M-1)}$ and time of each piece T_i . Hence we can transform the optimization variables from \mathbf{c} to \mathbf{z}'_w and \mathbf{T} . The trajectory naturally satisfies the initial and final states Eq.(7c) and higher order continuity Eq.(7d). In the subsequent optimization, we choose $s = 4$ to obtain a smoother trajectory and allow the joint's torque to be smooth.

B. Smoothing of inequality constraint functions

Considering the strict positive constraint of time Eq.(7e), we can change optimization variables from real time $T_i \in R^+$ to unconstrained virtual time $\tau_i \in R$ with a smooth bijection [18].

Penalty methods [19] can be used for other inequality constraints Eq.(7f) and Eq.(7g) with a first-order smooth penalty function $L_1(x)$. Since we cannot directly solve the integral of the penalty function in a closed form, we can approximate it by discretizing the constraint function and calculating its sum:

$$\mathcal{C}_d(\mathbf{c}_i, T_i, \hat{t}) = \mathcal{C}_d(\mathbf{c}_i^T \boldsymbol{\beta}(T_i \cdot \hat{t}), \dots, \mathbf{c}_i^T \boldsymbol{\beta}^{(s)}(T_i \cdot \hat{t})), \quad (8)$$

$$I_d(\mathbf{c}, \mathbf{T}) = \sum_{i=0}^M \sum_{j=0}^{\kappa} \frac{T_i}{\kappa} \bar{\epsilon}_j L_1(\mathcal{C}_d(\mathbf{c}_i, T_i, \frac{j}{\kappa})), \quad (9)$$

where \hat{t} is the normalized timestamp, $\kappa \in Z^+$ is the number of samples, $(\bar{\epsilon}_0, \bar{\epsilon}_1, \dots, \bar{\epsilon}_{\kappa-1}, \bar{\epsilon}_{\kappa}) = (1/2, 1, \dots, 1, 1/2)$ are the coefficients from the trapezoidal rule [20].

C. Dense sampling for nonlinear constraints

When modeling, we find that both angular velocity ω and angular acceleration α of the base are related to the reciprocal of velocity from Eq.(3c) and Eq.(3d), which leads to strongly nonlinear when the velocity is relatively small as shown in Fig. 4. It is inappropriate to sample it with the same density as other constraints, for its nonlinearity could lead to

mutations in an unsampled interval. Though denser sampling can constrain it better, it is also inappropriate to densely sample the entire trajectory, which will cause unnecessary computation. Therefore, we use a smooth activation function $L_a(x)$ to activate intervals of greater sampling density:

$$L_a(x) = \begin{cases} 0 & x < -\epsilon_a \\ \frac{1}{2\epsilon_a^4}(x + \epsilon_a)^3(\epsilon_a - x) & -\epsilon_a < x \leq 0 \\ \frac{1}{2\epsilon_a^4}(x - \epsilon_a)^3(\epsilon_a + x) + 1 & 0 < x \leq \epsilon_a \\ 1 & \epsilon_a < x \end{cases}, \quad (10)$$

where $\epsilon_a \in R^+$. We set a velocity v_{la} and set the activation value as the quadratic difference between the current velocity and v_{la} , i.e., $L_a(v_{la}^2 - v^2)$, so that the expression of dense sampling is:

$$I_a(\mathbf{c}, \mathbf{T}) = \sum_{i=1}^K \sum_{j=0}^{\kappa} L_a(v_{la}^2 - v_{i,j}^2) F(\mathbf{c}_i, T_i, j), \quad (11)$$

$$F(\mathbf{c}_i, T_i, j) = \sum_{k=0}^{\kappa'} \frac{T_i}{\kappa \kappa'} \bar{c}'_k L_1(\mathcal{C}_{a,i,j,k}).$$

$(\bar{c}'_0, \bar{c}'_1, \dots, \bar{c}'_{\kappa'-1}, \bar{c}'_{\kappa'}) = (1/2, 1, \dots, 1, 1/2)$ has the same meaning as \bar{c} , $\kappa' \in Z^+$ is the number of dense sampling, $v_{i,j} = \|\dot{\sigma}_i(\frac{j}{\kappa} T_i)\|_2$ is the velocity got from sparse sampling, $\mathcal{C}_{a,i,j,k} = \mathcal{C}_a(\mathbf{c}_i, T_i, \frac{j-1/2}{\kappa} + \frac{k}{\kappa \kappa'})$ has the same definition as Eq.(8). In particular, when $j = 0$ or κ , only $\kappa'/2 + 1$ times are sampled backward or forward. If $v_{i,j}$ is smaller than v_{la} , dense sampling is performed, by which the sampling density in the interval is increased to $\kappa \kappa'$. Otherwise, we can directly skip the subsequent dense sampling. Thus, we can densely sample only where nonlinear mutations are likely to occur, which balances the accuracy and efficiency.

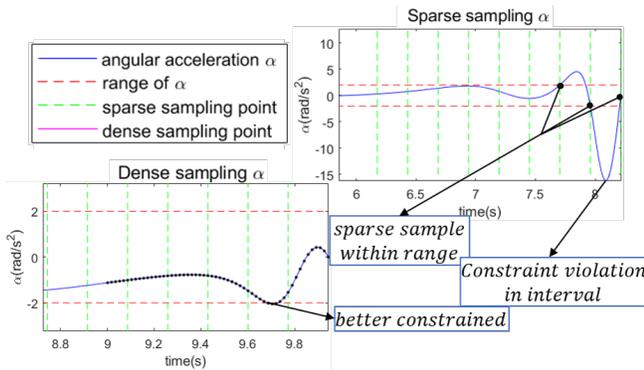


Fig. 4. Optimization of different sampling methods. Although α of sampling points is within the range, it may mutate (upper right) in sampling intervals. However, dense sampling can better limit α (lower left). Dense sampling is activated only for part of the time, and is not computed in larger speed intervals to save computing time.

As mentioned above, we can re-represent the optimization problem as an unconstrained problem by Eq.(9)(11):

$$\min_{z'_w, \tau} J = J_0(\mathbf{c}(z'_w, \tau), \mathbf{T}(\tau)) + I_\Sigma(\mathbf{c}(z'_w, \tau), \mathbf{T}(\tau)), \quad (12)$$

$$I_\Sigma = \omega_d I_d(\mathbf{c}(z'_w, \tau), \mathbf{T}(\tau)) + \omega_a I_a(\mathbf{c}(z'_w, \tau), \mathbf{T}(\tau)),$$

where ω_d, ω_a are weights of each constraint of the sparse and dense sampling. We sample base's constraints $\mathcal{D}_b =$

$\{d_b : v, a_l, \omega, \alpha, s\}$ and joint's constraints $\mathcal{D}_m = \{d_m : \omega_m, \alpha_m, s\}$ sparsely. While minimum velocity, angular velocity and angular acceleration of the base $\mathcal{A}_{base} = \{a_{base} : \bar{v}, \omega, \alpha\}$ are densely sampled additionally.

So far, we have transformed the problem into an unconstrained optimization problem as Eq.(12). If we have the expression of inequality constraints, we can use the chain rule to get the gradient and solve the problem by L-BFGS algorithm [21].

V. DYNAMIC FEASIBILITY AND SAFETY CONSTRAINTS

In this section, we focus on the specific forms of inequality constraints.

A. Linear velocity and angular velocity of the base

From Eq.(1) we can find the maximum angular velocity ω_{max} and velocity v are linear correlations in kinematics because of the limited maximum wheel speeds. However, the torque required for rotation is different from linear motion usually. We simplified this by still assuming that ω_{max} is linearly related to v but scaled with a factor $\eta \in R^+$: $\omega_{max}(v) = 2\eta(v_{max} - v)/d_{wb}$, where v_{max} is the maximum speed of the base and also the driving wheels. Then we can give constraint functions of v and ω :

$$\mathcal{C}_v(\dot{\sigma}) = \dot{\sigma}^T \dot{\sigma} - v_{max}^2, \quad (13)$$

$$\mathcal{C}_{\omega_r}(\dot{\sigma}, \ddot{\sigma}) = \frac{\ddot{\sigma}^T \mathbf{B} \dot{\sigma}}{\|\dot{\sigma}\|_2^2} - \omega_{max}(\|\dot{\sigma}\|_2), \quad (14)$$

$$\mathcal{C}_{\omega_l}(\dot{\sigma}, \ddot{\sigma}) = -\frac{\ddot{\sigma}^T \mathbf{B} \dot{\sigma}}{\|\dot{\sigma}\|_2^2} - \omega_{max}(\|\dot{\sigma}\|_2),$$

where $\mathbf{B} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is a constant matrix, and $\|\cdot\|$ is 2-norm of vectors. We use constraints on both ends of ω to reduce the nonlinearity from adversely affecting the optimization.

B. Linear acceleration and angular acceleration of the base

Considering the torque limitation of the wheels, we also set constraints of linear acceleration a_l and angular acceleration α for the differential base:

$$\mathcal{C}_{a_l}(\dot{\sigma}, \ddot{\sigma}) = \frac{(\ddot{\sigma}^T \dot{\sigma})^2}{\|\dot{\sigma}\|_2^2} - a_{l,max}^2, \quad (15)$$

$$\mathcal{C}_{\alpha_r}(\dot{\sigma}, \ddot{\sigma}, \sigma^{(3)}) = \frac{\sigma^{(3)T} \mathbf{B} \dot{\sigma}}{\|\dot{\sigma}\|_2^2} - \frac{2\ddot{\sigma}^T \mathbf{B} \dot{\sigma} \ddot{\sigma}^T \dot{\sigma}}{\|\dot{\sigma}\|_2^4} - \alpha_{max}, \quad (16)$$

$$\mathcal{C}_{\alpha_l}(\dot{\sigma}, \ddot{\sigma}, \sigma^{(3)}) = -\frac{\sigma^{(3)T} \mathbf{B} \dot{\sigma}}{\|\dot{\sigma}\|_2^2} + \frac{2\ddot{\sigma}^T \mathbf{B} \dot{\sigma} \ddot{\sigma}^T \dot{\sigma}}{\|\dot{\sigma}\|_2^4} - \alpha_{max},$$

where $a_{l,max}, \alpha_{max}$ are the maximum linear acceleration and the maximum angular acceleration.

C. Minimum velocity of the base

Since both ω and α take velocity's higher orders as the denominator and have no limit when v approaches 0, which means that there will be a singularity when v is equal to zero, as in Eq.(3c) and Eq.(3d). To avoid the singularity, we set a minimum velocity v_{min} . If the velocity v_d is less than v_{min} at a given initial or final state, we will set the velocity to $v'_d = [v_{min} \cos \theta_d, v_{min} \sin \theta_d]^T$, where $\theta_d = \arctan 2(v_{dy}, v_{dx})$.

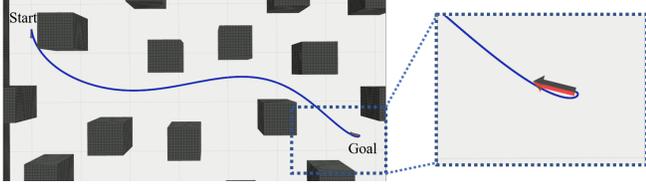


Fig. 5. Trajectory optimization in dense environments. The proposed method allows rotations with a very small radius, the minimum radius of the trajectory in the figure is 9.3 mm, which can be equivalent to spinning in place.

In the subsequent application, we choose $v_{min} = 10^{-2}m/s$, whose corresponding minimum rotation radius is only several millimeters, as shown in Fig. 5, which can be approximated as spin in place. We also give the minimum speed constraint in the optimization:

$$\mathcal{C}_v(\dot{\sigma}) = v_{min}^2 - \dot{\sigma}^T \dot{\sigma}. \quad (17)$$

Therefore we avoid the singularity of the flat output while preserving the motion performance.

D. Velocity and angular acceleration of the joint

Considering the limitation of joint's torque, it is necessary to constrain its velocity and acceleration:

$$\mathcal{C}_{\omega_m}(\dot{\theta}) = \dot{\theta}^2 - \omega_{m,max}^2, \quad (18)$$

$$\mathcal{C}_{\alpha_m}(\ddot{\theta}) = \ddot{\theta}^2 - \alpha_{m,max}^2, \quad (19)$$

where $\omega_{m,max}^2, \alpha_{m,max}^2$ is the maximum velocity and the maximum acceleration of the joint.

E. Safety

Considering the shape of our system, we should consider the safety of the whole body instead of reducing it to a point. Therefore we use ESDF to obtain the signed distance of any point relative to obstacles. After obtaining the gridmap, we can get ESDF by computing an efficient $O(n)$ algorithm [22], where n is the number of updated grids. Because of the errors of ESDF introduced by the discretization of the voxel grid map, trilinear interpolation is used to improve the accuracy of the distance and gradient information [23]. We can ensure safety by constraining ESDF values of the whole system greater than the safety distance d_s .

We take γ_b points ${}^b\mathbf{h}_{b,l} \in \mathbb{R}^3$ along base's contour. Then their corresponding positions ${}^w\mathbf{p}_{b,l}$ can be calculated as:

$${}^w\mathbf{p}_{b,l} = \begin{bmatrix} \sigma \\ 0 \end{bmatrix} + {}^w\mathbf{R}_b {}^b\mathbf{h}_{b,l}, \quad {}^w\mathbf{R}_b = \begin{bmatrix} [\dot{\sigma}, B\dot{\sigma}] & \mathbf{0} \\ \|\dot{\sigma}\|_2 & 1 \end{bmatrix}, \quad (20)$$

where ${}^w\mathbf{R}_b$ is the rotation matrix of the base. Then the safety constraint of the base can be expressed as:

$$\mathcal{C}_{s,b,l}(\sigma, \dot{\sigma}) = d_s - E({}^w\mathbf{p}_{b,l}), \quad (21)$$

where $E(\cdot)$ is the ESDF value obtained by trilinear interpolation. Similarly, we can get the position of each point on the manipulator with Eq.(4). We choose γ_m points ${}^j\mathbf{h}_{m,l} \in$

$\mathbb{R}^3, l \in \{1, 2, \dots, \gamma_m\}$ along manipulator's contour. Then their corresponding positions ${}^w\mathbf{p}_{m,l}$ can be calculated as:

$${}^b\mathbf{T}_j = \begin{bmatrix} {}^b\mathbf{R}_j & {}^b\mathbf{p}_{m,0} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad {}^w\mathbf{T}_b = \begin{bmatrix} {}^w\mathbf{R}_b & [\sigma^T \ 0]^T \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (22)$$

$${}^j\bar{\mathbf{h}}_{m,l} = [{}^j\mathbf{h}_{m,l}^T, 1]^T, \quad [{}^w\mathbf{p}_{m,l}^T, 1]^T = {}^w\mathbf{T}_b {}^b\mathbf{T}_j {}^j\bar{\mathbf{h}}_{m,j},$$

Therefore we can get the safety constraint of the joint as:

$$\mathcal{C}_{s,m,l}(\sigma, \dot{\sigma}, \theta) = d_s - E({}^w\mathbf{p}_{m,l}). \quad (23)$$

Until now, we give all constraints of SCR-DB.

VI. EXPERIMENT

Three sets of experiments were conducted to validate our proposed system. Firstly, we test in complex 3D simulated environments and show the optimized smooth trajectories relative to the initial path. Then, we compare our method with other vehicle planning solutions to verify the effectiveness of our method for the differential base. Lastly, we verify our method in the real-world setup to prove our system. We use the hybridA* [24] to give the initial path.

A. Planning for the shape-changing robot

This section aims to verify that our proposed method can work well in the simulation environment. We test our algorithm in two scenarios with obstacles, including cubes and arch bridges. All simulation experiments are conducted on a desktop computer running Ubuntu 18.04 with an Intel Core i7-10700K CPU.

In the first scene, there are two kinds of arch bridges, and we give the initial path to pass through arch bridges to test in the specific 3D environment as shown in Fig. 6.

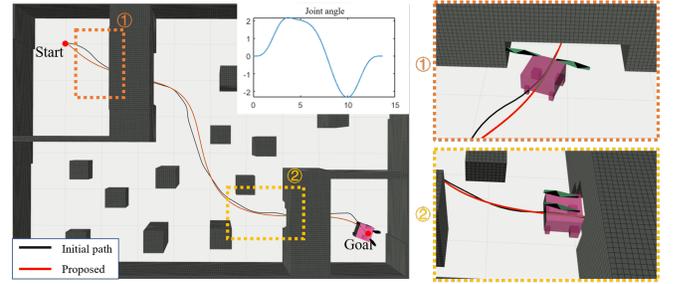


Fig. 6. The robot crosses arch bridge obstacles of different shapes, the green is the trajectory of the proposed method. The curve is the change of the joint angle relative to time.

In the second scene, we randomly choose the start and end states in the environment of size $10m \times 10m \times 1m$ like fig. 1 for 500 experiments. We build ESDF in advance with a time of 170ms. Average line acceleration (Ma), average angular acceleration ($M\alpha$), and average jerk (Mj) are used to measure the smoothness of the trajectory [18]. Average joint jerk (Mmj) is used to measure the joint's smoothness for jerk minimization, which is able to reduce tracking error and realize small actuator stresses. The trajectory length of the base ($length$) and trajectory time ($time$) are for measuring trajectory quality. The data are shown in Tab.I. The success rate is 93.0% with an average computation time (CT) of 104.2ms.

TABLE I. Comparison before and after optimization

	M_j (m/s^3)	Ma (m/s^2)	$M\alpha$ (rad/s^2)	length (m)	time (s)	Mmj (rad/s^3)
initial	0.614	0.428	2.683	8.870	9.516	2.000
optimized	0.193	0.370	0.724	8.508	10.196	0.294

B. Planning for the differential base

This section aims to verify our proposed real-time planning for the differential base. The size of scenarios is set to $10m \times 10m \times 0.4m$, the size of the differential vehicle is $40cm \times 40cm \times 30cm$, and the obstacles are all rectangles of $60cm \times 60cm \times 40cm$. We randomly select feasible initial and final points and build ESDF in advance with time spent 71.2ms.

To verify the effect of dense sampling, we experiment 200 times in the above scenarios with 20 obstacles. Then we count the number of conditions that the maximum α of the trajectory exceeds constraint with different thresholds (TH) and the proportion of time when α exceeds the constraint (PT). The result is shown in Tab.II. It can be found that dense sampling can reduce the level of violation, and appropriately dense sampling can speed up convergence.

TABLE II. Comparison with and without Dense Sampling (DS)

	TH	TH	TH	PT	CT
	< 10%	< 25%	> 25%	%	(ms)
without DS	10	72	128	4.36	131.8
with DS	152	165	35	2.24	64.8

What's more, we design 3 scenarios with different obstacle densities. We choose Timed Elastic Bands (TEB) [16] for comparison. We perform 200 tests with randomly given starting and ending states with distances greater than 5m. The data are shown in Tab.III.

TABLE III. Comparison in different cases for differential-driven base

Env	Method	M_j (m/s^3)	$M\alpha$ (rad/s^2)	CT (s)	length (m)	time (s)
sparse (10boxes)	Proposed	0.554	0.503	0.051	7.822	9.355
	TEB	7.794	1.684	6.287	7.945	5.184
dense (20boxes)	Proposed	0.550	0.548	0.053	7.720	9.585
	TEB	5.899	1.753	7.578	7.794	5.412
denser (35boxes)	Proposed	0.574	0.598	0.058	7.656	10.332
	TEB	6.987	1.845	20.208	7.778	5.855

The results show that our planner takes less planning time and produces smaller jerk, linear and angular accelerations, which makes trajectories smoother and easier to be tracked. In addition, the proposed method takes the limitation of torque into account instead of simply giving the maximum v and ω . The trajectory can better satisfy the dynamics constraints of the differential vehicle and be better executed. The results of different methods are shown in Fig. 7.

C. Real-World Experiments

We also test in a real environment to verify that our proposed method can work on real robots. The experiment site is a dense $5m \times 6m$ space with square obstacles and tapes for height limitation. The experiment uses the above-mentioned designed SCR-DB in Fig. 2, which is developed

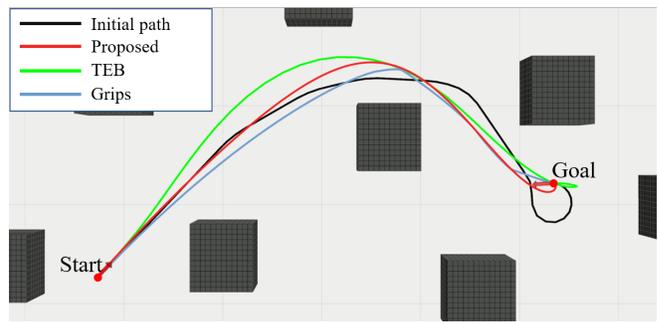


Fig. 7. Visual comparison of different methods. Since our algorithm runs in 3D space, but most of the planning runs in 2D space, the edges of the obstacles are set parallel or perpendicular to the horizontal plane to avoid the effect of extra dimension. Grips [13] is a path optimization method. It can be seen that the trajectory of the proposed method is smoother.

based on a differential-drive agilex TRACER MINI¹ platform with a maximum speed of $0.5m/s$, a scaling factor $\eta = 0.1$, and a pair of controllable blades. The processing unit of the robot is Intel NUC11TNHi7. We use the motion capture system for positioning and map the environment in advance. The robot's movement is shown in Fig. 8, which shows that the robot can avoid obstacles following the planned trajectory.

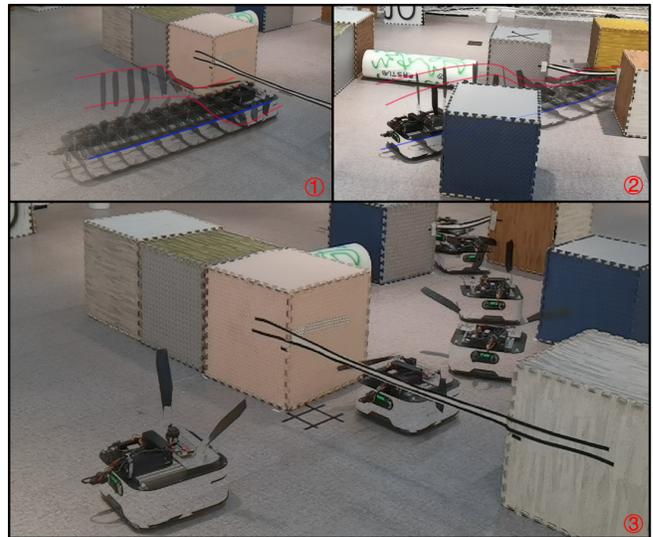


Fig. 8. Real-world experiment.

VII. CONCLUSION

In this paper, we propose a trajectory optimization method for shape-changing robots with differential-driven base. By modeling the optimization problem, we complete the planning of the proposed SCR-DB in a 3D environment and preserve the motion ability of each part as much as possible. In simulation and real experiments, we test the effectiveness and robustness of our method. Moreover, our algorithm can be applied to robots consisting of a differential base and additional DOFs. In the future, we will try our method on more robots, such as mobile manipulators.

¹<https://www.agilex.ai/product/16>

REFERENCES

- [1] “Hello robot: Simply useful robots,” <https://hello-robot.com/>, [Accessed 12-Sep-2022].
- [2] “Picker Robot: inVia Robotics,” <https://inviarobotics.com/>, [Accessed 12-Sep-2022].
- [3] S. M. LaValle *et al.*, “Rapidly-exploring random trees: A new tool for path planning,” 1998.
- [4] J. Kuffner and S. LaValle, “Rrt-connect: An efficient approach to single-query path planning,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.
- [5] F. Burget, M. Bennewitz, and W. Burgard, “Bi 2 rrt: An efficient sampling-based path planning framework for task-constrained mobile manipulation,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3714–3721.
- [6] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, “Chomp: Gradient optimization techniques for efficient motion planning,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 489–494.
- [7] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, “Stomp: Stochastic trajectory optimization for motion planning,” in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4569–4574.
- [8] J. Xu, K. Harada, W. Wan, T. Ueshiba, and Y. Domae, “Planning an efficient and robust base sequence for a mobile manipulator performing multiple pick-and-place tasks,” *CoRR*, vol. abs/2001.08042, 2020. [Online]. Available: <https://arxiv.org/abs/2001.08042>
- [9] S. Thakar, L. Fang, B. Shah, and S. Gupta, “Towards time-optimal trajectory planning for pick-and-transport operation with a mobile manipulator,” in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 981–987.
- [10] S. Thakar, P. Rajendran, A. M. Kabir, and S. K. Gupta, “Manipulator motion planning for part pickup and transport operations from a moving base,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 191–206, 2022.
- [11] M. Spahn, B. Brito, and J. Alonso-Mora, “Coupled mobile manipulation via trajectory optimization with free space decomposition,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 759–12 765.
- [12] J. Xu, Y. Domae, W. Wan, and K. Harada, “An optimization-based motion planner for a mobile manipulator to perform tasks during the motion,” in *2022 IEEE/SICE International Symposium on System Integration (SII)*, 2022, pp. 519–524.
- [13] E. Heiden, L. Palmieri, S. Koenig, K. O. Arras, and G. S. Sukhatme, “Gradient-informed path smoothing for wheeled mobile robots,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1710–1717.
- [14] Z. Jian, S. Zhang, J. Zhang, S. Chen, and N. Zheng, “Parametric path optimization for wheeled robots navigation,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 883–10 889.
- [15] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, “Trajectory modification considering dynamic constraints of autonomous robots,” in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.
- [16] C. Rosmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, “Efficient trajectory optimization using a sparse model,” in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 138–143.
- [17] I. Okuyama, M. R. Maximo, and R. J. Afonso, “Minimum-time trajectory planning for a differential drive mobile robot considering non-slipping constraints,” *Journal of Control, Automation and Electrical Systems*, vol. 32, no. 1, pp. 120–131, 2021.
- [18] Z. Han, Y. Wu, T. Li, L. Zhang, L. Pei, L. Xu, C. Li, C. Ma, C. Xu, S. Shen, and F. Gao, “Differential flatness-based trajectory planning for autonomous vehicles,” 2022. [Online]. Available: <https://arxiv.org/abs/2208.13160>
- [19] Z. Wang, X. Zhou, C. Xu, and F. Gao, “Geometrically constrained trajectory optimization for multicopters,” *IEEE Transactions on Robotics*, 2022.
- [20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [21] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.
- [22] P. F. Felzenszwalb and D. P. Huttenlocher, “Distance transforms of sampled functions,” *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.
- [23] V. Usenko, L. Von Stumberg, A. Pangercic, and D. Cremers, “Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 215–222.
- [24] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path planning for autonomous vehicles in unknown semi-structured environments,” *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.