



A geometric motion planning for a spin-rolling sphere on a plane

Seyed Amir Tafrishi^{a,*}, Mikhail Svinin^b, Motoji Yamamoto^c, Yasuhisa Hirata^d

^a Engineering School, Cardiff University, 5 The Parade, Newport Road, Queen's Buildings, Cardiff, CF24 3AA, Wales, UK

^b Department of Info. Science and Eng., Ritsumeikan University, 1-1-1 Noji-Higashi, Shiga-ken, 525-8577, Japan

^c Department of Mechanical Engineering, Kyushu University, 744 Motoooka, Nishi-ku, Fukuoka, 810-0395, Japan

^d Department of Robotics, Tohoku University, Aoba 6-6-01, Aramaki, Sendai, 980-8579, Japan

ARTICLE INFO

Article history:

Received 11 January 2023

Revised 25 April 2023

Accepted 4 May 2023

Available online 7 May 2023

Keywords:

Motion planning

Spin-rolling

Darboux frame

Shortest path

Smooth trajectory

Geometric mechanics

ABSTRACT

The paper deals with motion planning for a spin-rolling sphere when the sphere follows an optimal straight path on a plane. Since the straight line constrains the sphere's motion, controlling the sphere's spin motion is essential to converge to a desired full configuration of the sphere. In this paper, we show a new geometric-based planning approach that is based on a full-state description of this nonlinear system. First, the problem statement of the motion planning is posed. Next, we develop a geometric controller implemented as a virtual surface by using the Darboux frame kinematics. This virtual surface generates arc-length-based inputs for controlling the trajectories of the sphere. Then, an iterative algorithm is designed to tune these inputs for the desired configurations. Finally, the feasibility of the proposed approach is verified by simulations.

© 2023 The Authors. Published by Elsevier Inc.
This is an open access article under the CC BY license
(<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Manipulating a spin-rolling sphere to the desired configuration is a significant research problem that can find many applications. Different mechanisms can realize a spin-rolling motion of the sphere on a plane (see Fig. 1). In the field of dexterous manipulation, rolling fingertips with convex surfaces can be utilized to grasp and manipulate different objects [1–3]. For example, Yuan et al. developed rolling spherical fingertips for grasping mechanisms [3]. Propulsion systems can realize spin-rolling motions with multiple-degrees-of-freedom (DoFs) actuators. For instance, spherical mobile robots with mass-imbalance propulsion [4,5], with cart-based actuators inside their spherical shell [6,7] and internal motor drivers [8–10] can have a similar feature with simple increase in number of actuators. Additionally, the ballbot robots with drivers on top of the rolling ball can be included in the kinematic model of spin-rolling motion [11,12]. Furthermore, nano/micro spherical particle manipulation on plane surfaces is finding important applications [13–15]. Thus, the problem of path planning for a spin-rolling sphere is applicable to different mechanisms.

The spin-rolling motion analysis can be done using the conventional ball-plate system kinematics [16,17] but we cannot spin the ball directly by the rotation of the upper sandwiched plane [18]. Different kinematic parametrization of rolling contact has been considered in [18–23]. To conduct motion planning, Kiss et al. [1] used three independent planes to manipulate the sphere at the kinematic level and controlled the relative angles of the sphere without considering desired plane

* Corresponding author.

E-mail address: Tafrishisa@cardiff.ac.uk (S.A. Tafrishi).

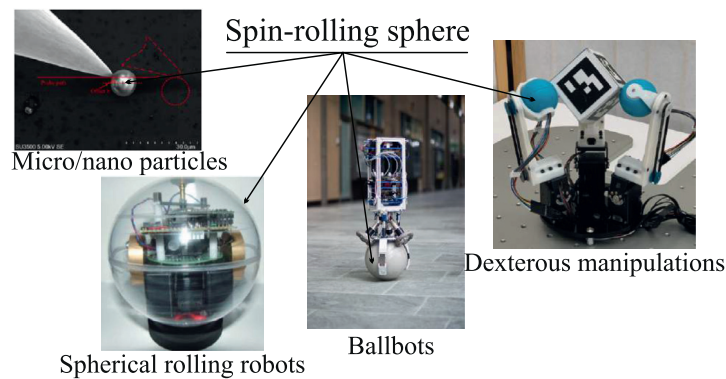


Fig. 1. Realization of the kinematics for spin-rolling motion of the sphere in different topics: the micro/nano particle manipulation [13], the spherical rolling robots [6], ballbots [12] and the Dexterous manipulation [3].

configuration. Date et al. used the advantage of spinning indirectly to control the ball-plate system [24]. Their motion planning algorithm was based on shifting the coordinate of the actuating plane in iterations with respect to different reference frames. This change can be looked like a rotational virtual centre. However, the time scaling with the included coordinate transformation of the kinematic model can bring the system to an uncontrollable state [25].

In the literature on the pure rolling, different feed-forward [10,16,26,27] and feedback control [24,25,28,29] methods were developed. Some works [30,31] also used a geometric phase control where an optimal problem was solved to follow certain trajectories. Since the spin angle change affects the sphere's orientation, the conventional planning approaches, such as the ones based on the geometric phase shifting [32–36], cannot be adjusted naturally for the spin-rolling motion planning. Arthurs and Walsh [37] and Jurdjevic [16] proposed planning approaches based on the optimal control theory. This research direction was developed later on by Sachkov [38], Mashtakov and Sachkov [39]. As an alternative to feedforward planning, to stabilize the non-differentially flat ball-plate system, iterative steering was formulated by Oriolo and Vendittelli [25]. However, the solutions had serious fluctuations as trajectories converged to the desired states. Thus, their created trajectories were not easy to be realized by dynamical systems [40] and there were singularities in different regions of the spherical manifold due to the locality of the solutions. Beschastnyi [41] studied the optimality problem of the spin-rolling sphere. Extremal trajectories were parameterized, and their cut times were estimated for optimality. It was proposed that Maxwell time be determined while the sphere follows the straight path for the optimal solution. However, this research did not cover the control problem for arbitrary desired states.

In this paper, we approach the motion planning problem with two main motivations: considering a spin-rolling sphere on the plane and developing optimal smooth and non-trivial trajectory for the arbitrary desired states. Our method defines the optimal path as the shortest trajectory on the plane surface that can be travelled by a rolling sphere from its starting point to its desired configuration. The length of this optimal path is determined by the Euclidean distance between the start and end points, which is always the minimum. This path is considered optimal since the length of the trajectory on the sphere is equivalent to the distance travelled along a straight line on the plane due to the no-sliding constraint. Consequently, our approach results in the shortest path between two points on the plane surface and minimizes the amount of rolling required for the sphere to reach its desired configuration with a single trajectory. This creates a smooth and non-trivial trajectory in the sphere, without requiring convergence with discrete sequences of motions [8,24,25,32,34]. Our proposed method finds the best path by solving our new geometric model while all the states converge at once (single sequence), thus eliminating the need for arbitrary sine-based angular motions (with projected similar curvature of trajectories on both sphere and plane) that are predominant in geometric phase shifting approaches [8,25,34]. Our approach is based on parameterizing the spin-rolling sphere motion by using the Darboux-frame-based kinematics of the ball-plate system initially proposed by Cui et al. [2], Cui and Dai [21,42]. In our previous work [18], we transformed the underactuated ball-plate system to a fully-actuated one with arc-length-based control inputs. Also, a new assistive geometric-based controller is developed using the same kinematic transformation for mobile robots [43]. In this paper, after explaining the motion planning problem, we introduce a virtual surface to manipulate the control inputs of the Darboux-frame-based kinematic model (derived in [18]) to the desired configuration. To the best of our knowledge, this is the first proposed geometric arc-length-based control strategy for the motion planning of a spin-rolling sphere. Furthermore, this control strategy separates the time scale from the kinematic equations, which allows for constructing motion with different convergence rates in a given time. Finally, we propose a tuning algorithm to solve the differential kinematic model in iterations until achieving a successful convergence to the final full configuration. It is important to note that the designed Darboux-frame parametrization increases the number of planning parameters that simplify the developed algorithm. Also, the optimal solution that the algorithm tunes can result in multiple solutions for the same configuration based on changing the geometric parameters in the planner.

This paper is organized as follows. Section 2 explains the motion planning problem and a distance constraint due to the no-sliding constraint. Next, Section 3 includes the design of the geometric controller for a Darboux frame kinematics. In Section 4, the iterative algorithm with included tuning variables is described. Simulations (the video of the simulation

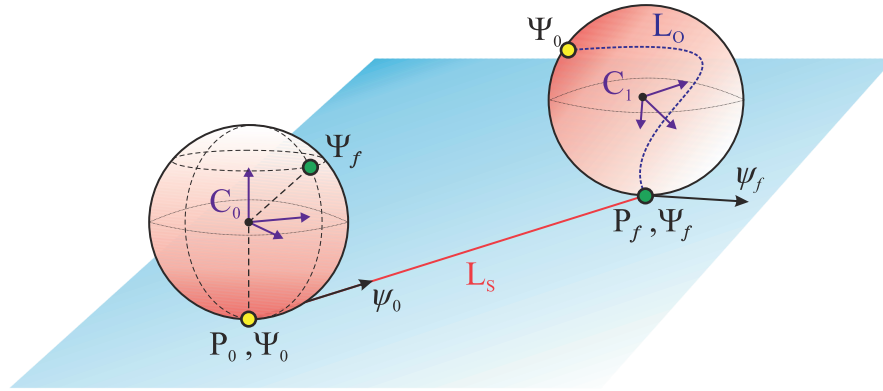


Fig. 2. Statement of a planning problem while the sphere with contact path of L_o follows a straight path L_s on the plane.

is available at Youtube [44]) of this new approach are demonstrated and discussed in Section 5. Finally, we conclude our findings in Section 6.

2. Problem statement

It is assumed that the sphere rolls and spins along a straight path on the plane, and the path on the sphere, connecting the initial and final configurations, must be established. Because of the no-sliding constraint, the length of the curve on the sphere L_o is equal to that on the plane L_s , the curve L_o may not be long enough to reach certain configurations on the sphere. This motion happens under the assumption that the plane trajectory L_o is not allowed to be travelled by the sphere in reverse (with the aim to extend the travel length of the trajectory L_o). This limitation is mainly due to our considered approach with a restricted path on the plane. To deal with this issue, we find a distance constraint that determines the minimum length of the curve for reaching the sphere to its desired states. Note that this limitation only appears when the length between the initial and final states of the plane is less than the sphere’s circumference. We calculate this distance constraint for the plane by applying the Gauss-Bonnet theorem [45] to the created circular path on the desired angular states of the sphere.

Configuration of a sphere on a plane (see Fig. 2) is described with the sphere position on plane $\mathbf{P} = (u_s, v_s)$ and its local orientation $\Psi = (u_o, v_o)$ which ψ is the spin angle between the sphere and plane [34]. We consider our initial and final states with following notations $\{\mathbf{P}_0, \Psi_0, \psi_0\} = \{u_{s,0}, v_{s,0}, u_{o,0}, v_{o,0}, \psi_0\}$ and $\{\mathbf{P}_f, \Psi_f, \psi_f\} = \{u_{s,f}, v_{s,f}, u_{o,f}, v_{o,f}, \psi_f\}$, respectively. The coordinate system and preliminary information of this ball-plate system are described in Appendix A.

We utilize the model of the Darboux frame at the contact point of the spin-rolling sphere and plane presented in [18]. The kinematics readily suits both contact trajectories and arbitrary parameters of the surfaces [2,21,46]. Also, this transformation provides two significant benefits through our planning approach. First, the spin-rolling angular rotations explicitly appear on the relative curvature and torsion [21] making it easier for manipulation. Second, the Darboux frame separates the time variable from the planning due to its time- and coordinate-invariance. As shown in [18], the transformed fully-actuated Darboux-frame-based kinematics model of a spin-rolling sphere on a plane in the time domain is presented as

$$\begin{aligned}
 \begin{bmatrix} \dot{u}_s(t) \\ \dot{v}_s(t) \\ \dot{u}_o(t) \\ \dot{v}_o(t) \\ \dot{\psi}(t) \end{bmatrix} &= \delta \left(\begin{bmatrix} \frac{\sin(\theta + \varphi)}{\sin(\theta + \varphi)} \\ \frac{\sin(\theta + \varphi)[\sin \psi - \cos \psi]}{R_o \cos v_o} \\ \frac{\sin(\theta + \varphi)[\cos \psi + \sin \psi]}{R_o} \\ \frac{\tan v_o[\sin(\theta + \varphi)(\sin \psi - \cos \psi) + \cos \varphi]}{R_o} \end{bmatrix} + \begin{bmatrix} -R_o \sin(\theta + \varphi) \\ -R_o \sin(\theta + \varphi) \\ \frac{\sin(\theta + \varphi)[\cos \psi - \sin \psi]}{\cos v_o} \\ -\sin(\theta + \varphi)[\sin \psi + \cos \psi] \\ \tan v_o[\sin(\theta + \varphi)(\cos \psi - \sin \psi)] \end{bmatrix} \gamma_s \right. \\
 &\left. + \begin{bmatrix} R_o \sin(\theta + \varphi) \\ -R_o \cos(\theta + \varphi) \\ -\sin(\psi + \theta + \varphi) \\ \frac{\cos v_o}{-\cos(\psi + \theta + \varphi)} \\ -\tan v_o \sin(\psi + \theta + \varphi) \end{bmatrix} \beta_s + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \alpha_s \right). \tag{1}
 \end{aligned}$$

where five-state system $\{\mathbf{P}(t), \Psi(t), \psi(t)\}$ has $\{\alpha_s, \beta_s, \gamma_s, G_f, \delta\}$ inputs. Also, the angles $\{\varphi(G_f), \theta(\beta_s, \gamma_s, G_f)\}$ as a function of extended angular input $G_f \in [-\pi, \pi]$ in (1) are for changing the direction of the sphere on the plane, they are obtained from our Darboux-frame-based transformation in previous work [18], that are given

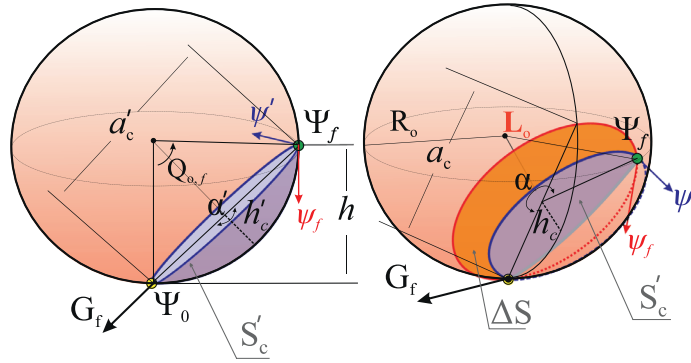


Fig. 3. Calculation of path length d that is required to reach the desired configuration $\{\Psi_f, \psi_f\}$. Note that G_f shows the desired goal angle on the plane.

$$\theta(\beta_s, \gamma_s, G_f) = \cot^{-1} \left[\frac{1}{\beta_s} \left(\frac{1}{R_o} (1 - \tan G_f) + \gamma_s (-1 + \tan G_f) - \beta_s \tan G_f \right) \right] - \psi_q,$$

$$\varphi(G_f) = \psi_q + \begin{cases} \pi, & -\frac{3\pi}{4} < G_f < 0 \quad \& \quad 0 \leq G_f < \frac{\pi}{4} \\ 0, & -\pi < G_f < -\frac{3\pi}{4} \quad \& \quad \frac{\pi}{4} \leq G_f < \pi \end{cases} \quad (2)$$

where ψ_q is the spin angle deviation. Please note that there is a shift $\frac{\pi}{4}$ between the φ and G_f , which appears due to the nature of $\theta + \varphi$ in (2) [18]. The three arc-length-based control inputs $\{\alpha_s, \beta_s, \gamma_s\}$ correspond to the angular velocities of the sphere in the time domain. The remaining angular G_f and rolling rate δ inputs are for directing sphere on fixed surface (plane) while they are constraining virtual surface inputs $\{\alpha_s, \beta_s, \gamma_s\}$. The controllability of this geometric model (1) is in [18]. This geometric-based controller converges the rotating object (sphere) trajectory to the desired angular states of the sphere while G_f keeps the sphere in a suitable direction with a rolling rate δ . In here, we define the goal angle by $G_f = \tan^{-1} [(v_{s,f} - v_{s,0}) / (u_{s,f} - u_{s,0})]$; hence, θ and φ will change relative to any values of arc-length inputs, for keeping the sphere always along G_f angle to reach \mathbf{P}_f . Therefore, the sphere stays in the prescribed straight direction by substituting constant value to G_f from the desired position on the plane \mathbf{P}_f .

In the proposed planning method, the new state Eqs. (1)-(2) are solved in k iterations till finding an admissible path. The travelled paths on the sphere and on the plane are denoted as \mathbf{L}_o and \mathbf{L}_s . We plan the motion of the sphere reaching its final configuration $C_0 \rightarrow C_1$ with a straight trajectory on \mathbf{L}_s in given time t_f (see Fig. 2 as an illustration). It is assumed that the initial states $\{\mathbf{P}_0, \Psi_0, \psi_0\}$ and the final desired states of the sphere $\{\Psi_f, \psi_f\}$ are given. However, the final desired position on the plane \mathbf{P}_f has to be chosen under the consideration of length limitation between initial and final position $\|\mathbf{P}_f - \mathbf{P}_0\| = [(u_{s,f} - u_{s,0})^2 + (v_{s,f} - v_{s,0})^2]^{\frac{1}{2}}$ because the total path length of \mathbf{L}_o on the sphere is always the same (no-sliding constraint) as straight trajectory on the plane \mathbf{L}_s . The optimal solution in this planning problem is considered as the Euclidean distance between initial and desired states on plane, $\min_{\mathbf{P}(t)} \|\mathbf{P}_f - \mathbf{P}(t)\| \leq \|\mathbf{P}_f - \mathbf{P}_0\|$. Because the length of path of plane \mathbf{L}_s is same as the sphere \mathbf{L}_o due to no-sliding constraint, the non-trivial single optimal trajectory will be the solution of the motion planning. This means local coordinate of sphere $\Psi(t)$ during convergence will be constrained under optimality. However, the spin angle $\psi \in [-\infty, \infty]$ has arbitrary orientation since it has not effect on the path of plane \mathbf{L}_s but only \mathbf{L}_o .

Let d be the length of a path that is constructed as a circular segment on the sphere that reaches Ψ_f (red line illustrated in Fig. 3), and then the cap area is changed (from blue to the orange area as shown in Fig. 3) toward true desired ψ_f by Gauss-Bonnet theorem [45]. Therefore, by knowing the sphere desired states $\{\Psi_f, \psi_f\}$, plane's desired states \mathbf{P}_f are chosen in any desired direction $\{u_{s,f}, v_{s,f}\}$ with the length $\|\mathbf{L}_s\| = \|\mathbf{P}_f - \mathbf{P}_0\|$ larger than the path length d . The path length d is determined by the circumference of a circular path that the sphere travels between Ψ_0 and Ψ_f (red line \mathbf{L}_o on the right part of Fig. 3) as

$$d = 2\pi a_c \alpha < \|\mathbf{P}_f - \mathbf{P}_0\|, \quad (3)$$

where a_c and α are the base diameter of this circumference (circle with orange cap-area in Fig. 3) and the angle of sector from Ψ_0 to Ψ_f on the sphere U_c , respectively.

Note that the sphere has to arrive at desired local coordinate Ψ_f with different approaching angles, as the desired spin ψ_f , while it follows an optimal straight line rather than manoeuvring freely through the plane U_s . This constraint is not important and the sphere can reach all possible configurations if the length of the desired position $\|\mathbf{P}_f - \mathbf{P}_0\|$ is set larger than sphere circumference $2\pi R_o$. Also, we assume that the rolling sphere never moves in the reverse direction of the traversed plane's trajectory \mathbf{L}_s to extend the sphere's surface U_c trajectory length on \mathbf{L}_o . Also, if the condition (3) is violated, there will be no solution for the planning strategy under the given assumption.

In order to calculate d , we find the base diameter a_c of the cap with the following formulation

$$a_c = 2\sqrt{(S_t/\pi) - h_c^2}, \quad h_c = S_t/(2\pi R_o), \quad (4)$$

where S_t and h_c are the total areas of the cap-shaped region [orange and blue regions in Fig. 3] and the height of the cap area. Notice that the rotation of the sphere along the red path L_o doesn't cover the whole cap-shaped region. Thus, we find the total area as $S_t = 2S'_c + \Delta S$ where S'_c is the constructed cap area of the circular path by Ψ_f [see the blue cap-shaped region in Fig. 3] and ΔS is the area change [see the orange region in Fig. 3] for reaching from the spin angle ψ' with cap area of S'_c to desired spin angle ψ_f . Here, S_t has two parts, combining them and giving the full segment of the circular cap (dashed and solid red line L_o). First, the area of the cap-shaped region of S'_c is determined from the simple closed circle that passes Ψ_f

$$S'_c = (\alpha'/2\pi) \left[(a'_c/2)^2 + h'^2_c \right], \tag{5}$$

where a'_c , h'_c and α' are the diameters of the cap's base, the height of the cap and the sector angle from sphere initial to final configuration Ψ_f on the cap's base, α' equals to π . Following parameters in (5) are calculated with the help of (A.1) and trigonometric relations, shown in Fig. 3,

$$h = R_o [1 - \cos u_{o,f} \cos v_{o,f}],$$

$$a'_c = [h^2 + R_o^2 (\sin^2 v_{o,f} + \sin^2 u_{o,f} \cos^2 v_{o,f})]^{1/2},$$

$$h'_c = \begin{cases} R_o [1 - \cos (Q_{o,f}/2)], & a'_c \leq 2R_o \\ R_o [1 - a'_c \cos (Q_{o,f}/2)/2R_o], & a'_c > 2R_o \end{cases}$$

where $Q_{o,f}$ is the angle from initial contact point $\Psi_0 = \{0, 0\}$ to desired local coordinates Ψ_f as

$$Q_{o,f} = \begin{cases} \pi - 2 \cos^{-1} (h/a'_c), & h \leq R_o \\ \frac{\pi}{2} - \sin^{-1} ((h - R_o)/R_o), & h > R_o \end{cases}$$

Also, by knowing the a_c and a'_c , the new sector angle from Ψ_0 to $\{\Psi_f, \psi_f\}$ (L_o as the red line in Fig. 3) becomes

$$\alpha = \frac{1}{2\pi} \cdot \begin{cases} 1 - 2 \sin^{-1} (a'_c/a_c), & \Delta S \geq 0 \\ 2 \sin^{-1} (a'_c/a_c), & \Delta S < 0 \end{cases} \tag{6}$$

To find the achieved spin angle ψ' by the contacted S'_c cap-area, we use the Gauss-Bonnet theorem [45] while the sphere touches the circular closed blue path

$$\Delta\psi = \psi' - \psi_0 = \iint_{S'_c} \kappa_o dS = S'_c/R_o^2, \tag{7}$$

where $\kappa_o = 1/R_o^2$ is the Gaussian curvature and the initial spin angle is assumed $\psi_0 = 0$. Finding the spin angle ψ' also lets us calculate ΔS in S_t . The required area-change ΔS for achieving ψ_f is calculated by the same Gauss-Bonnet theorem as follows

$$\Delta S = (\psi_f - \psi')/\kappa_o = R_o^2(\psi_f - \psi'). \tag{8}$$

After obtaining the total area S_t for reaching ψ_f , a_c in (4) and α in (6) give us the value of path length d . Fig. 4 depicts the normalized length for different desired angles, which shows how the small distance $\|\mathbf{P}_f - \mathbf{P}_0\|$ can be for the example

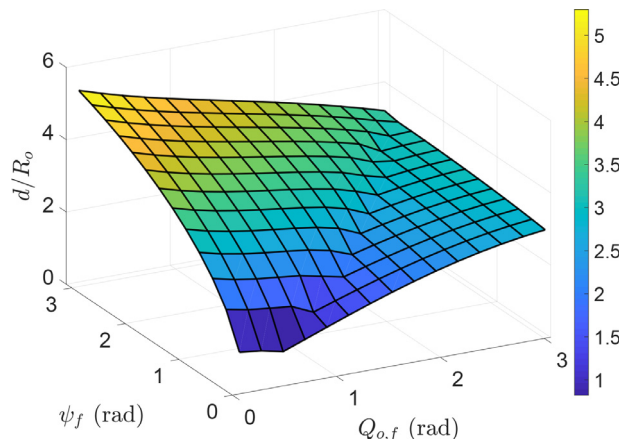


Fig. 4. Normalized length d/R_o for different desired angles where $u_{o,f} \in [0, \pi]$ and $v_{o,f} = 0.01$.

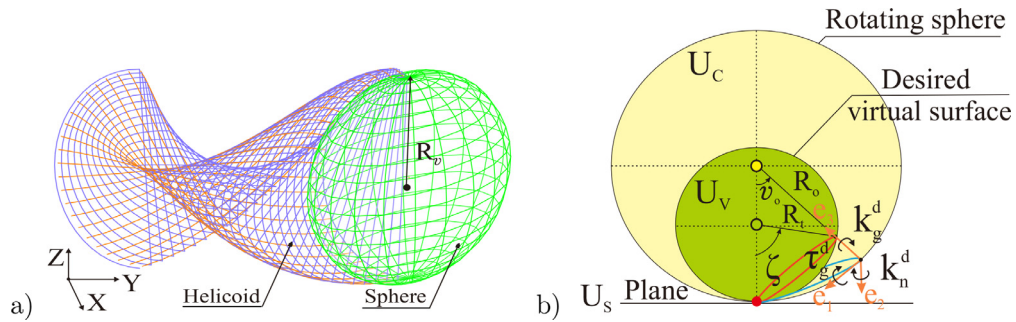


Fig. 5. a) The geometric shape of the used virtual surface U_v . Note: Green and blue surfaces are for $R_t = 0$ as the sphere and $R_t = R_v$ as the helicoid, b) Design of the arc-length-based inputs by using the desired virtual surface. Note that schematic is drawn for $u_o = 0$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

final configurations. As the desired local orientation Ψ_f moves to the upper-hemisphere, \mathbf{P}_f has to set for the longer length from \mathbf{P}_o . Also, the same property is true for raising the spin angle ψ_f . Moreover, the largest distance requirement happens at lower points of Ψ_f ($Q_{o,f} \leq \pi/2$) with larger desired spin angles ψ_f . Note that this constraint (3), with the assumption of a simple circular curve, is looked at as the shortest length under the isoperimetric inequality, $4\pi S_t \leq d^2$.

3. Geometric controller for Darboux-frame-based kinematics

The Darboux-frame-based kinematics (1)-(2) with the inputs in the arc-length domain requires a geometric control with tuning algorithm for converging spin-rolling sphere to desired angular configuration Ψ_f . In the bigger picture, we will tune the developed geometric controller in our path planning algorithm to converge toward the desired configuration. We introduce a virtual surface U_v to define functions of these arc-length-based inputs $\{\alpha_s, \beta_s, \gamma_s\}$ in (1). Also, the virtual surface is a surface sandwiched between the sphere and plane at the contact frame. From a physical point of view, this virtual surface's deformation (curvature changes) is projected onto both sphere and plane trajectories [18]. These changes in the curvature manipulate the curve on the sphere L_o like a flexible rope by tuning the geometric controller. To bend this rope-like curve toward the desired full configuration, we propose a tuning algorithm to update the variables of this virtual surface (geometric controller) in the incoming section.

Here, we design the virtual surface for controlling the curve L_o on the sphere U_c when L_s is specified as a straight path.

Definition 1. In order to develop a geometric controller based on the virtual surface, we use the induced curvature (differential characteristics), with the geodesic curvature k_g , the normal curvature k_n and the geodesic torsion τ_g , that consists of the rolling convex object (sphere) on a plane with sandwiched virtual surface as follows [18]

$$k_g^o = k_g^s - k_g^o - \alpha_s, \quad \tau_g^o = \tau_g^s - \tau_g^o - \beta_s, \quad k_n = k_n^o - k_n^s - \gamma_s, \tag{9}$$

where $\{k_g^o, \tau_g^o, k_n^o\}$, $\{k_g^s, \tau_g^s, k_n^s\}$ and $\{\alpha_s, \beta_s, \gamma_s\}$ are the geodesic curvature, geodesic torsion and normal curvature for, respectively, the rolling object (sphere), plane and virtual surface. In the geometric controller, the curvature properties are designed with similarity to rotating object (sphere) manifold U_c (A.2) for planning on the virtual surface. However, the moving object is a sphere with $\tau_g^o = 0$. This cause the model (1) to be unstable [18] because $\beta_s = 0$. As a solution, geodesic torsion of a helicoid shape is applied where the geometric shape is shown in Fig. 5a (see Appendix B for details of derivation)

$$\tau_g^v = \frac{1}{R_v^2} |R_v^2 \cos^2 v_v(t) - R_t^2|^{\frac{1}{2}}, \tag{10}$$

where R_v , R_t and $v_v(t)$ are defined by the main spherical radius, the sum of spherical and torsion radii and v -curve angle of the helicoid surface, respectively. As a keynote, the geometric surface shown in Fig. 5a is able to transform from the sphere to the helicoid surface by changing R_t .

Next, we utilize (9) to construct the desired virtual surface (see Fig. 5b) with the normal and geodesic curvature of the spherical surface $\{k_n^d(s), k_g^d(s)\}$ in (A.2)-(A.3) and the geodesic torsion of the helicoid surface $\tau_g^d(s)$ (10) as follows

$$\begin{aligned} k_g^d(s) &= \tan \zeta / R_t = \tan v_{o,f} / R_o - \alpha_s, \\ \tau_g^d(s) &= \frac{1}{R_o^2} |R_o^2 \cos^2 v'_o - R_t^2|^{\frac{1}{2}} = \beta_s, \\ k_n^d(s) &= 1/R_n = 1/R_o - \gamma_s. \end{aligned} \tag{11}$$

where $R_t = R_n + R_g$ and $v'_o = v_{o,f} - v_o(t)$ are the total radius and the angle feed of v -curve in which R_n and R_g are the desired normal curvature and geodesic torsion radii. Also, we assume ζ is the desired stereographic projection angle for

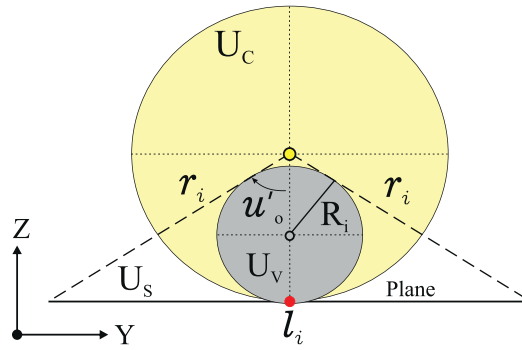


Fig. 6. R_i determination by incircle of u'_o .

the spherical object along $k_g^d e_3$ vector. By considering that the diameter of projected curve is same (red and blue curves in Fig. 5b) and $R_t \tan \zeta = R_o \tan \nu_o$, ζ is determined

$$\zeta = \tan^{-1} [R_o \tan (\nu_{o,f} + \zeta') / R_t], \tag{12}$$

where ζ' is the constant angle shift used during planning updates. Also, R_n and R_g are defined as

$$R_n(t) = R_g(t) = [R_i(t) + R_a] / 2, \tag{13}$$

where R_i and R_a are the built-in circle radius of u -curve and tuning constants that happens by algorithm, respectively. By using the schematic of Fig. 6, built-in radius R_i is computed with the change of the designed incircle radius as follows

$$R_i(t) = \begin{cases} hnt \left[\frac{(S_i - r_i)^2 (S_i - l_i)}{S_i} \right]^{\frac{1}{2}}, & 0 \leq u'_o < \frac{\pi}{2} \\ \frac{R_o}{\mu_r} + \left[\frac{(S_i - r_i)^2 (S_i - l_i)}{S_i} \right]^{\frac{1}{2}}, & \frac{\pi}{2} \leq u'_o \leq \pi \end{cases} \tag{14}$$

where $S_i = (2r_i + l_i) / 2$ is the area of encompassed triangle of the incircle, $u'_o = u_{o,f} - u_o(t)$ is the convergence of u -curve angle, μ_r is the scaler to limit the maximum built-in radius R_i , and also $r_i = R_o / \cos u'_o$ and $l_i = 2R_o \tan u'_o$ are adjacent and hypotenuse sides of isosceles triangle (see Fig. 6).

Remark 1. The sphere curvature in (11) is included to cancel out the existing sphere U_C properties that drift terms is presenting in the kinematic model (1). Thus, this design gives a direct manipulation of virtual surface curvatures (s-domain) on L_o trajectory for Ψ_f .

Now, we can find the geometric control inputs from (11) as

$$\alpha_s = \tan \nu_{o,f} / R_o - \tan \zeta / R_t, \beta_s = \frac{1}{R_o^2} |R_o^2 \cos^2 \nu'_o - R_t^2|^{\frac{1}{2}}, \gamma_s = \frac{R_n - R_o}{R_n R_o}. \tag{15}$$

By substituting the desired values $(u_{o,f}, \nu_{o,f})$, the curves of L_o are constrained on the virtual surface with the projection on the rotating object, as the example of Fig. 5. This constrained curve is created from a conservative vector field in (15) where independent inputs act as force fields $\mathbf{F} = \nabla E_{in}(u_o(t), \nu_o(t), u_{o,f}, \nu_{o,f}) = \{\alpha_s, \beta_s, \gamma_s\}$ to manipulate at contact frame on U_C manifold [47]. By using kinematics (1)-(2), it is interpreted that the desired virtual surface bends this vector field towards the desired local coordinate Ψ_f . Note that our curve manipulation on the sphere manifold U_C can be imagined as a flexible rope with length of $\|\mathbf{P}_f - \mathbf{P}_o\|$ that $\{R_n, R_g\}$ and ζ terms change the curvature radius and angle of ν -curve for L_o trajectory on the spherical surface U_C .

It should also be noted that the proposed virtual surface is designed in a way that eases our planning problem with spherical curvature properties k_n^v and k_g^v . However, more complicated virtual surfaces with existing geodesic torsion τ_g^v can be applied to this planning approach for creating different contact paths, but it can complicate planning problems, e.g., algorithm tuning, which requires a separate study.

4. Motion planning algorithm

4.1. Iterative tuning algorithm

We plan the motion of the sphere spinning on the plane along the straight line L_s with reaching the desired orientation. The flowchart in Fig. 7a shows our algorithm that uses the control inputs (15). Geometrically, it can be considered as manipulating curvature radii $\{R_n, R_g\}$ and angular location ζ' to bend the created rope-like curve L_o from (15) on the sphere. For example, constant raise in $\{R_n, R_g\}$ expand the curve L_o like enlarging the loops of the rope while its length is same

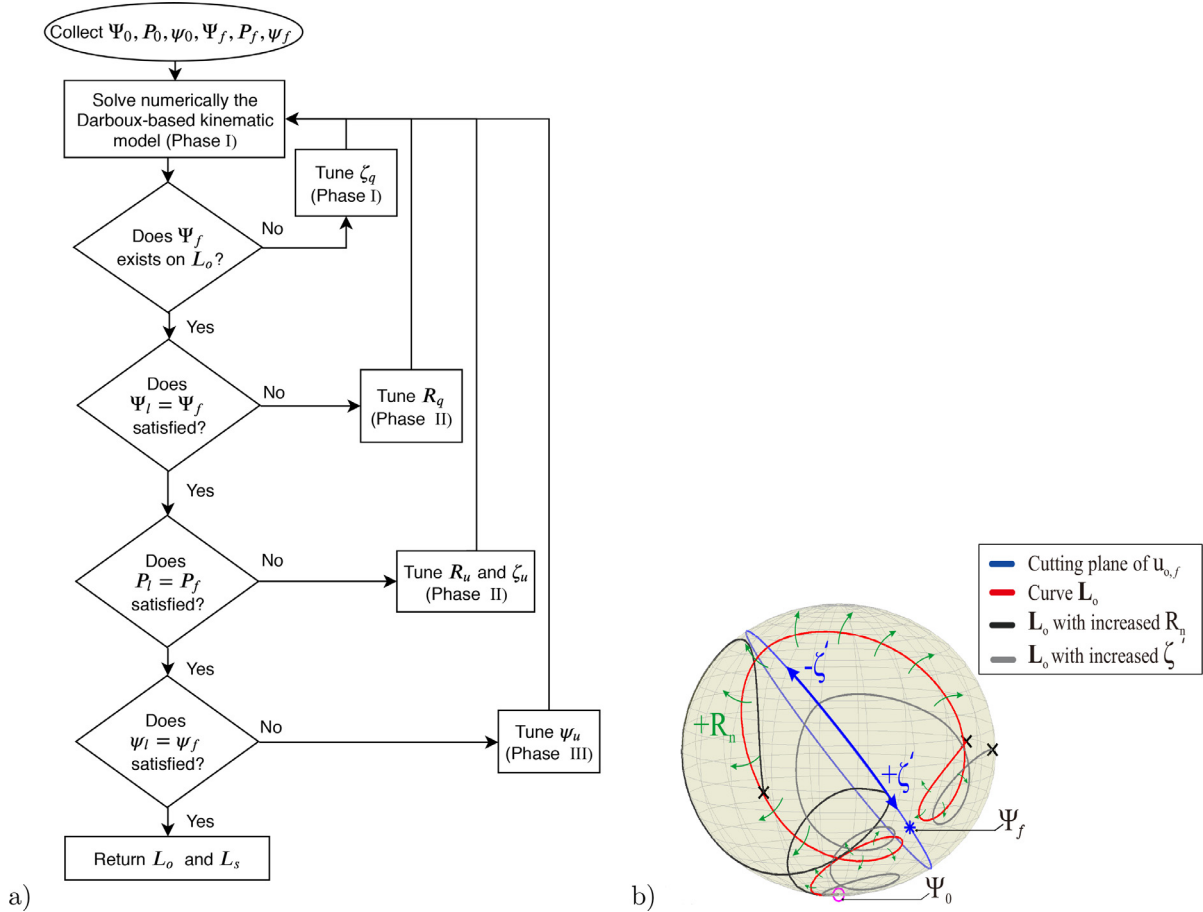


Fig. 7. a) The flowchart of planning algorithm (Algorithm A), b) Manipulation of the rope-like L_0 curve with the designed virtual surface. Note that following curves are obtained by using formulas in Phase I for ψ_q and δ in (16) and (17).

[see Fig. 7b]. Also, constant changes on ζ' moves the curve L_0 on the cutting plane of $u_{0,f}$. To do the convergence of this rope-like model, the proposed algorithm will solve (1)-(2) and (15) in iterations with the re-tuned constants in $\{R_n, R_g\}$ and ζ' .

In general, to shift the curve of the sphere L_0 by $\zeta'(k) = \zeta_q(k) + \zeta_u(k)$ on the cutting plane $u_{0,f}$, updates happen in Phase I and II by ζ_q and ζ_u , respectively. Also, constant change in the desired radius R_a is defined by the inclusion of all phases as $R_a(k) = R_q(k) + R_u(k)$, where R_q and R_u are tuning constants of radii in the steps of the Phase II. Our tuning algorithm consists of three primary steps:

The first step, Phase I, is the main part of the algorithm where the kinematic model (1)-(2) with designed arc-length-based inputs (15) is numerically solved in the time domain. After obtaining L_0 and L_s curves, Phase I checks whether there is a point on L_0 that passes the desired local coordinates on the sphere Ψ_f . If L_0 fails to reach Ψ_f , ζ_q is re-tuned and kinematic model is resolved numerically. After succeeding Phase I, the curve is checked whether the final point of local coordinate Ψ_l is at desired values of Ψ_f as Fig. 8. If the curve L_0 fails from $\Psi_l = \Psi_f$, the radius of desired virtual surface R_q is increased by obtained error. Also, Phase II has a second extra step to reach the sphere final distance P_l to the exact desired plane configuration $P_l = P_f$ by tuning R_u and ζ_u parameters. The final state as the desired spin angle ψ_f is achieved in the final step. Note that this algorithm requires numerically solving differential equations in k iterations until full convergence of the chosen configuration.

4.1.1. Phase I

This phase solves the kinematic model (1)-(2) first. Then, it finds the curve L_0 on the rotating sphere that passes Ψ_f while the sphere moves toward desired position P_f .

Before defining the tuning functions, we have to develop a converging state for the spin angle. The spin motion can make tracking unstable when the sphere passes the same desired spin angle many times during its rotation. Thus, we develop an indirect convergence by using spinning angle deviation ψ_q to converge the desired spin angle ψ_f . During calculations of d at (3), we assumed that under-cap area S_t of traversed distance L_0 is constant while $C_0 \rightarrow C_1$. Thus, the desired under-cap area

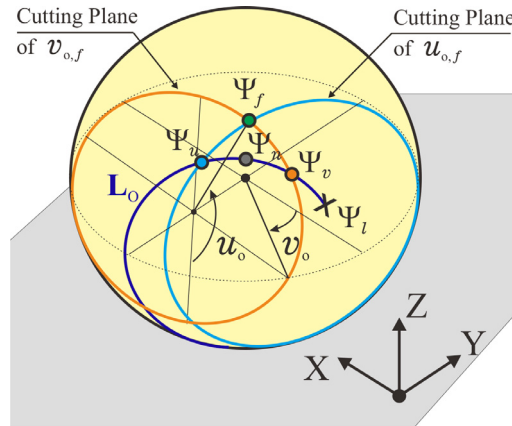


Fig. 8. Parametrization of the curve L_o by cutting planes of $u_{o,f}$ and $v_{o,f}$.

is defined by $S_t = S'_c + R_o^2(\psi_f + \psi_u(k) - \psi')$ (only orange and blue cap areas excluding dashed-line part as Fig. 3), where $\psi_u(k)$ is the constant re-tuning spin angle for Phase III. Next, we extend the Gauss-Bonnet theorem between two points of $\{u_o(t), v_o(t), \psi(t)\}$ and $\{u_{o,f}, v_{o,f}, \psi_f\}$ in time t , as

$$\psi_q(t) = \frac{1}{R_o^2}[S_t - S_i(t)] = \frac{1}{R_o^2}[S_t - R_o^2(\psi(t) - \psi_o)], \tag{16}$$

where $S_i(t)$ and $\psi(t)$ are the changing cap area from the initial state till the current time t and the current spin angle.

As the last parameter in the kinematic model, the rolling rate (arc-length derivation relative to the time $\delta(t)$) is the parameter that is multiplied by all three inputs and drifts term. This variable is related to the time (ds/dt) and it varies the arc-length step of curve L_o and L_s in the given time t . Therefore, δ is defined to have a rest-to-rest motion by

$$\delta(t, u_o, u_s, v_s) = \|\mathbf{P}_f - \mathbf{P}(t)\| \cdot \left| \frac{v_{o,f} \cdot u'_o}{T} \right| = [(u_{s,f} - u_s(t))^2 + (v_{s,f} - v_s(t))^2]^{\frac{1}{2}} \cdot \left| \frac{v_{o,f} \cdot u'_o}{T} \right|, \tag{17}$$

where T is the time scaling.

Algorithm 1 shows the computations of Phase I. By solving the Darboux-based kinematic model, the geometric parameters (see Fig. 8) are determined from obtained L_o and L_s trajectories. Ψ_l and \mathbf{P}_l are the final arrived configuration by the numerical solution of the found trajectories. Also, Ψ_n , Ψ_u and Ψ_v are the nearest point on L_o to Ψ_f and existing nearest points on L_o that are obtained by cutting planes of $u_{o,f}$ and $v_{o,f}$ on the sphere. Note that all points are collected as the final existing values since the curve can pass the nearest point or cutting planes more than once.

Algorithm 1 Phase I.

- 1: **procedure** CONFIGSOLVE($\Psi_f, \mathbf{P}_f, \psi_f, \Psi_0, \mathbf{P}_0, \psi_0$)
 - 2: **while** $e_n \geq \epsilon_n$ **do**
 - 3: Solve the kinematic model (1) numerically
 - 4: Collect Ψ_n, Ψ_l, Ψ_v and Ψ_u
 - 5: Compute e_n and e'_n by (18) ▷ Error of nearest point to Ψ_f
 - 6: **if** $e_n(k) > \epsilon_n$ **then**
 - 7: Apply the directional comparison by Ψ_n, Ψ_v and Ψ_u ▷ Details in Appendix C
 - 8: Calculate $\zeta_q(k)$ by (19)
 - 9: **end if**
 - 10: **end while**
 - 11: **end procedure**
-

In Algorithm 1, ϵ_n is a small value for breaking while loop when required accuracy is achieved for Phase I. Also, the error e_n is the Euclidean distance of nearest Ψ_n and desired Ψ_f points in \mathbb{R}^3

$$e_n = \|\Psi_n - \Psi_f\| = R_o \left[(\sin v_{o,f} - \sin v_{o,n})^2 + (\cos u_{o,f} \cos v_{o,f} - \cos u_{o,n} \cos v_{o,n})^2 + (\sin u_{o,f} \cos v_{o,f} - \sin u_{o,n} \cos v_{o,n})^2 \right]^{\frac{1}{2}}. \tag{18}$$

Next, the absolute angle difference between Ψ_f and Ψ_n are utilized to update the $\zeta_q(k)$ angle in k th iteration of Phase I as follows

$$\zeta_q(k) = \zeta_q(k - 1) \pm \begin{cases} e'_n \cdot |Q_f^{zx} - Q_n^{zx}|, & Q_f^{zx} = Q_n^{zx} \\ e'_n \cdot |Q_f^{zy} - Q_n^{zy}|, & Q_f^{zx} \neq Q_n^{zx} \end{cases} \tag{19}$$

where $e'_n = \min\{e_n\}$ is smallest error of e_n till iteration k and the angle differences with respect to $Z - X$ plane Q^{zx} and $Z - Y$ plane Q^{zy} for Ψ_f/Ψ_n are obtained by (A.1) as

$$Q^{zx} = \left| \tan^{-1} \left(\frac{\sin u_o \cos v_o}{\cos u_o \cos v_o} \right) \right| = |u_o|, \quad Q^{zy} = \left| \tan^{-1} \left(\frac{\sin u_o \cos v_o}{\sin v_o} \right) \right|. \tag{20}$$

The sign of each update in (19) is chosen from the directional updates in Appendix C. The goal of the directional update is always to move the curve L_o towards Ψ_f on the spherical surface U_C .

4.1.2. Phase II and III

After finding the suitable curve L_o from Phase I that passes Ψ_f , Phase II tunes variable for converging L_o and L_s final arrived points (Ψ_l and P_l) to the desired configurations on Ψ_f and P_f . Algorithm 2 depicts the overall computations, including all the phases using the flowchart of Fig. 7a.

Algorithm 2 Complete Computation for Algorithm A.

```

1: procedure CONFIGCOM( $\Psi_f, P_f, \psi_f, \Psi_0, P_0, \psi_0$ )
2:   while  $e_s \geq \epsilon_s$  do ▷ Phase III
3:     while  $(e_n \geq \epsilon_n) \ \& \ (e_r \geq \epsilon_r) \ \& \ P_l \neq P_f$  do ▷ Phase II
4:       while  $e_r \geq \epsilon_r$  do ▷ Phase II
5:         CONFIGSOLVE(.) ▷ Phase I
6:         Compute  $e_r, e'_r, n_s$  and  $d_s$ 
7:         if  $\Psi_l \neq \Psi_n$  then ▷ Updates  $R_q$ 
8:            $R_q(k) \leftarrow +R_q(k - 1)$  according to (22)
9:         else
10:           $R_q(k) \leftarrow -R_q(k - 1)$  by  $n_s \leq 1$  case according to(22)
11:        end if
12:      end while
13:      if  $\|P_f - P_l\| > \epsilon_p$  then ▷ Phase II
14:        Update  $R_u$  and  $\zeta_u$ 
15:      end if
16:    end while
17:    if  $\psi_l \neq \psi_f$  then ▷ Phase III
18:      Update  $\psi_u$ 
19:    end if
20:    Reset  $e_n, e'_n, e_r$  and  $e'_r$ 
21:  end while
22: return  $L_o$  and  $L_s$ 
23: end procedure

```

In Phase II, the error e_r of final arrival point Ψ_l with respect to desire values of Ψ_f is first determined by

$$e_r = \|\Psi_l - \Psi_f\| = R_o \left[(\sin v_{o,f} - \sin v_{o,l})^2 + (\cos u_{o,f} \cos v_{o,f} - \cos u_{o,l} \cos v_{o,l})^2 + (\sin u_{o,f} \cos v_{o,f} - \sin u_{o,l} \cos v_{o,l})^2 \right]^{\frac{1}{2}}. \tag{21}$$

Next, the update of R_q is to enlarge the loops of L_o for shorting the distance between nearest Ψ_n and final arrived Ψ_l points; hence, the algorithm computes the R_q update for k th iteration by using the error e_r as

$$R_q(k) = R_q(k - 1) + R_o \cdot \begin{cases} d_s \cdot e'_r, & n_s \leq 1 \\ n_s \cdot d_s \cdot e'_r, & 1 < n_s \leq 2 \\ (d_s \cdot e'_r)/n_s, & n_s > 2 \end{cases} \tag{22}$$

where $e'_r = \min\{e_r\}$, d_s and n_s are the smallest error of e_r till iteration k , distance ratio of L_s for the loop on the curve L_o and the numbers of created full loops by spinning ψ , respectively. The distance ratio d_s and the number of created loops $n_s \in \mathbb{R}$ are calculated as follows

$$d_s = \frac{\|P_n - P_f\|}{\|P_0 - P_f\|} = \left[\frac{(u_{s,n} - u_{s,f})^2 + (v_{s,n} - v_{s,f})^2}{(u_{s,0} - u_{s,f})^2 + (v_{s,0} - v_{s,f})^2} \right]^{\frac{1}{2}},$$

$$n_s = |\psi_l / \psi_n|. \tag{23}$$

Note that n_s is computed with the assumption that ψ value can be greater than 2π which every 2π orientation represents a loop on spin angle [47].

As the final step of Phase II, because our arc-length-based inputs create curve \mathbf{L}_o with multiple loops (especially for $d \ll \|\mathbf{P}_f - \mathbf{P}_0\|$), we use R_u and ζ_u variable for shifting \mathbf{L}_o away from Ψ_0 on the surface of the sphere U_C (can be looked as decreasing the size of the enlarged R_q) with same expressed flexible rope model. This computation step with updating ζ_u and R_u tries to reach $\Psi_l \rightarrow \mathbf{P}_f$ as previous loops try to keep $\Psi_l = \Psi_f$ condition true,

$$R_u(k) = R_u(k-1) - R_q(k) \cdot \frac{\|\mathbf{P}_l - \mathbf{P}_f\|}{\|\mathbf{P}_l - \mathbf{P}_0\|},$$

$$\zeta_u(k) = \zeta_u(k-1) + \begin{cases} -\tan^{-1}\left(\frac{R_u(k)\tan\zeta_q(k-1)}{R_o}\right), & v_{o,f} \geq 0 \\ \tan^{-1}\left(\frac{R_u(k)\tan\zeta_q(k-1)}{R_o}\right), & v_{o,f} < 0 \end{cases} \tag{24}$$

Final phase is to update the tuning variable of the spinning angle ψ_u in (16) with

$$\psi_u(k) = \psi_u(k-1) - e'_s \operatorname{sgn}(\psi_f - \psi_l), \tag{25}$$

where $e'_s = \min\{\psi_f - \psi_l\}$ is the smallest error till iteration k . Also, ϵ_s , ϵ_r and ϵ_p are the accuracies of each designated step in Algorithm 2.

5. Global convergence of the planning algorithm

To prove the convergence of the proposed iterative Algorithm 2, one can utilize Zangwill's convergence theorem [48,49].

Theorem 1. Let the iterative algorithm $\mathbf{A} : \mathbb{R}^5 \rightarrow \mathbb{R}^3$, shown in the flowchart at Fig. 7, be on $\mathbf{X} = \mathbf{X}_I \circ \mathbf{X}_{II} \circ \mathbf{X}_{III}$ compact set where \mathbf{X}_I , \mathbf{X}_{II} and \mathbf{X}_{III} are sets of phase I, II and phase III computation steps, respectively. These sets contain the converged final states with re-tuned geometric controller (15) after solving the (1)–(2) in each particular phase in k th iteration. Given $\mathbf{x}_0 \in \mathbf{X}$, the created sequence $\{\mathbf{x}_k\}_{k=1}^\infty$ satisfies $\mathbf{x}_{k+1} \in \mathbf{A}(\mathbf{x}_k)$. Then, \mathbf{A} is globally convergent to a solution $\Gamma \subset \mathbf{X}$ with following conditional statements [48]

- \exists a descent function \mathbf{z} for Γ and \mathbf{A} ,
- The sequence $\{\mathbf{x}_k\}_{k=0}^\infty \subset S$ for $S \subset \mathbf{X}$ is a compact set,
- The mapping \mathbf{A} is closed at all points of \mathbf{X}/Γ .

Regarding the first condition in Theorem 1, the designed line search algorithm \mathbf{A} has descent function $\mathbf{z} = \{R_q, \zeta_q, R_u, \zeta_u, \psi_u\}$ which roughly can be presented as

$$\mathbf{z}(\mathbf{x}) = \mathbf{x}_k + \mathbf{h}_k \mathbf{d}_k \tag{26}$$

where \mathbf{d}_k and $\mathbf{h}_k = \operatorname{diag}\{h(1), h(2), h(3), h(4), h(5)\}$ are the direction of iteration vector and the step size diagonal matrix of iterations. Note that ζ_u is updated in the same way with others, only because R_u changes the curve \mathbf{L}_o , the new angular location shift happens by ζ_u . Next, the given descent function $\mathbf{z}(\mathbf{x})$ has the following properties:

$$\begin{cases} \text{If } \mathbf{x} \notin \Gamma \text{ and } \mathbf{y} \in \mathbf{X}, \mathbf{z}(\mathbf{y}) < \mathbf{z}(\mathbf{x}), \\ \text{If } \mathbf{x} \in \Gamma \text{ and } \mathbf{y} \in \mathbf{X}, \mathbf{z}(\mathbf{y}) \leq \mathbf{z}(\mathbf{x}), \end{cases}$$

hence, the first condition of Theorem 1 is satisfied. The algorithm \mathbf{A} is on the compact manifold of the sphere U_C and the plane U_S ; hence, \mathbf{h}_k norms as the error, e.g., $e_r, \mathbb{R}^n \rightarrow \mathbb{R}$ become a compact set as well. This can be clearly extended for the second condition about all produced sequences $\{\mathbf{x}_k\}_{k=0}^\infty$ on \mathbf{X} . Now, we provide a proposition to satisfy the final condition as

Proposition 1. Let f be a real continues function of \mathbf{X} . Then, Algorithm \mathbf{A} with the solution set

$$\Gamma(\mathbf{x}, \mathbf{d}) = \left\{ \mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = \mathbf{x} + \mathbf{h}\mathbf{d}, \mathbf{h} > 0, \text{ and } f(\mathbf{y}) = \min_{0 < \mathbf{h} < \infty} \{ f(\mathbf{x} + \mathbf{h}\mathbf{d}) \} \right\}, \tag{27}$$

is closed at any point (\mathbf{x}, \mathbf{d}) at which always vector $\mathbf{d} \neq \mathbf{0}$ and $\mathbf{h} > 0$ means all the diagonal elements in \mathbf{h} are positive.

Appendix D presents the proof of the given proposition which was explained in detail at [48,49]. By satisfying all the three conditions in Theorem 1, the proposed line search algorithm is always convergent to a solution for the desired configuration.

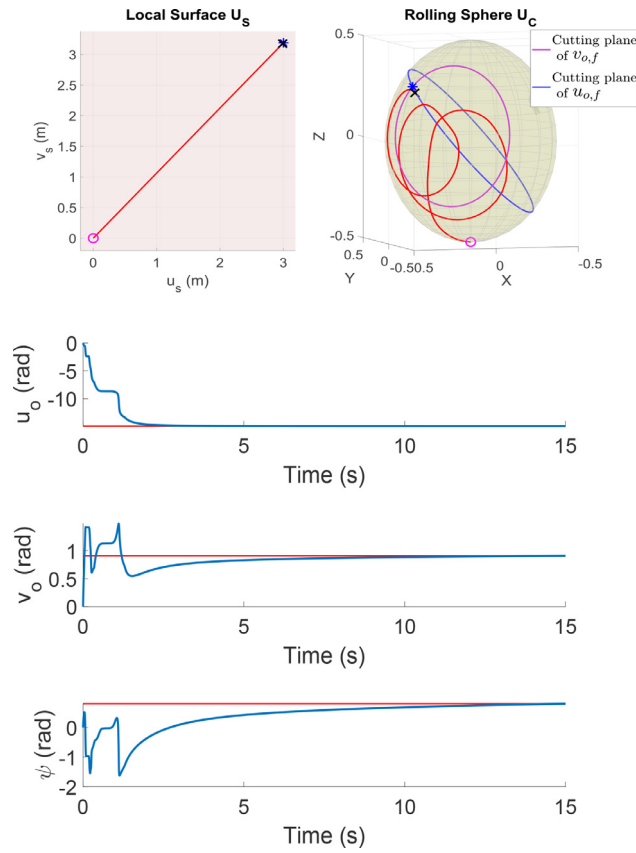


Fig. 9. Simulation results for a case study with the final configuration $\{u_{s,f}, v_{s,f}, u_{o,f}, v_{o,f}, \psi_f\} = \{3, 3.2, -\frac{\pi}{2} - 0.8, 0.8, 0.8\}$.

6. Results and discussion

The operation and performance of the proposed planning approach are tested under simulations. First, we check how phases of the algorithm work to achieve a successful convergence. Next, we analyze the simulation results for different desired spin angles while the rest of the configuration is the same.

In the simulations, the kinematic model (1)-(2) is solved together with (15) and (17) by using Matlab's ODE45 routine. We set the initial and final configurations as $\{u_{s,0}, v_{s,0}, u_{o,0}, v_{o,0}, \psi_0\} = \{0, 0, 0, 0, 0\}$ and $\{u_{s,f}, v_{s,f}, u_{o,f}, v_{o,f}, \psi_f\} = \{3, 3.2, -\frac{\pi}{2} - 0.8, 0.8, 0.8\}$. Note that our plane's final configuration satisfies distance constraint (3) where d for our case is $2.15 \leq \|\mathbf{P}_f - \mathbf{P}_0\|$, and to see curve with multiple loops, we have $\|\mathbf{P}_f - \mathbf{P}_0\| = 4.38$ m. For this case, the simulation time is set to $t_f = 15$ s and time constant T in (17) is 1. The accuracy of phases in Algorithm 2 are $\epsilon_n = \epsilon_r = 0.07$, $\epsilon_p = 0.12$ and $\epsilon_s = 0.05$. As expressed in the controllability analysis in Ref. [18], there is an uncontrollable point at $\pi/4$ on the plane. For this case, we chose it away from the point with $G_f = 0.754$ rad to see our approach abilities. The initial value of R_q is 0.005 to prevent any computation singularities. Also, we set the sphere radius and maximum divider as $R_0 = 0.5$ m and $\mu_r = 4$.

After running the simulation, the final results are shown as Fig. 9. The video of the simulation results for this section is available on Youtube [44]. The calculation steps k for this case are achieved with 48 iterations. The computation process takes about 7.2 s. By the achieved successful final iteration, we can interpret that while the sphere is rolling along the given desired final position \mathbf{P}_f , it spins with smooth trajectories toward its final configuration for $\{\Psi_f, \psi_f\}$.

We plot four collected iteration steps to understand how the algorithm works by utilizing the derived virtual surface [see Fig. 10]. From the initial iteration ($k = 0$), it is clear that \mathbf{L}_0 is created by the designed virtual surface toward the desired states Ψ_f but because \mathbf{P}_f is far, \mathbf{L}_0 curve moves along the $u_{o,f}$ cutting plane. A rope on a sphere explains the way that the achieved curve \mathbf{L}_0 on U_C can be imagined. So, the algorithm tunes the virtual surface and this bends the obtained curve \mathbf{L}_0 in loops, like a rope, toward the desired Ψ_f . Here, Phase I rises the \mathbf{L}_0 to move Ψ_n near to Ψ_f as the result of succeeded iteration $k = 11$. To move Ψ_1 on Ψ_f , Phase II enlarges the radius of the rope-like curve \mathbf{L}_0 , as the example iteration $k = 14$ shows. The iteration $k = 41$ in Fig. 10 satisfies $\Psi_1 = \Psi_f$ condition while the next step is tuning the plain configuration to achieve $\mathbf{P}_1 = \mathbf{P}_f$ using (24). Finally, the Phase III condition is satisfied for the desired ψ_f value shown in Fig. 9.

The angular velocities of the sphere collected by the search algorithm are presented in Fig. 11, but there are certain fast accelerations in the angular velocities. Note that the problem is considered at the kinematics level without any slipping;

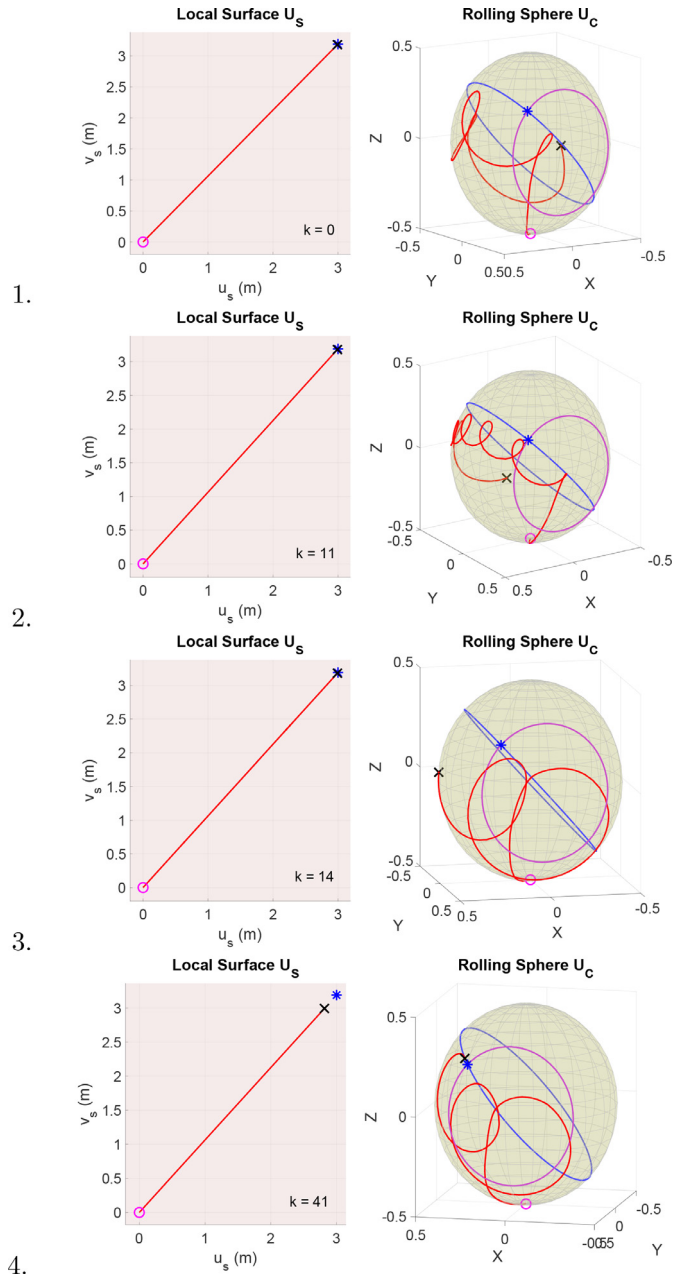


Fig. 10. Collected iterations of the given final configuration: 1. The first solution of system ($k = 0$), 2. The first successful results of Phase I ($k = 11$), 3. Converging Phase II ($k = 14$), 4. The first results before applying Phase III ($k = 41$).

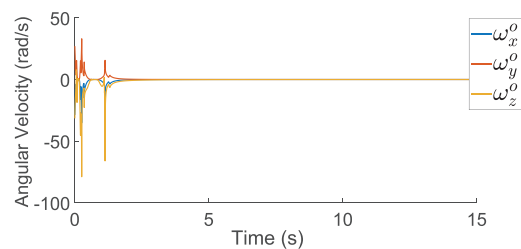


Fig. 11. Angular velocities of the sphere with the designed arc-length-based based controller.

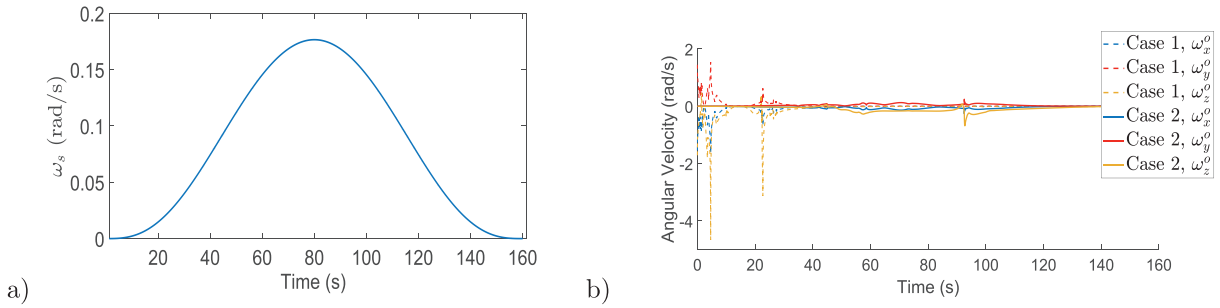


Fig. 12. a) The designed function ω_s in time, b) Smoothed angular velocities of the sphere in two different cases of time scaling T .

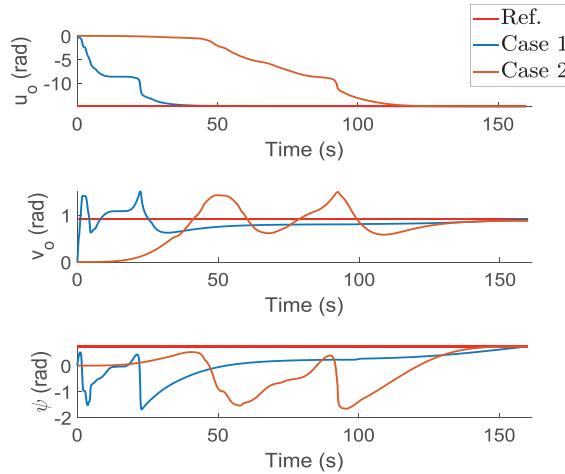


Fig. 13. Comparing angular convergence of Ψ_f at two cases.

Hence, arriving longer distances in a short amount of time results in faster velocities. However, we can use the aforementioned advantage in the derived kinematic model (1) where the time domain is separated from the kinematics by the rolling rate δ . Thus, We alter the sphere angular velocities in two cases:

In the initial case, the time span can be enlarged to decrease the overall velocity without affecting the achieved paths \mathbf{L}_o and \mathbf{L}_s . For the considered example, we increase the time scaling constant T (it is at δ in (17)) and simulation time t_f to 160 s while the kinematic model is solved with the succeeded parameters in ζ' and $R_q + R_u$ from the algorithm. Clearly, the overall velocities decreased by expanding them in time as Fig. 12b (dashed line velocities). However, fast accelerations (around 3 s and 21 s) make it harder to apply practically. To solve this issue, we keep the simulation time t_f the same as 160 s, but a smooth function is chosen for T as the second case study. With the known relation of $\omega_z^o = \delta k_g^* \triangleq \delta \alpha_s$ [18] and (17), we define T as

$$T \triangleq (c \cdot \omega_s) / \alpha_s, \tag{28}$$

where $c = |v_{o,f} \cdot u_o'| \cdot \|\mathbf{P}_f - \mathbf{P}(t)\|$ and ω_s is the desired smooth angular velocity. The desired smooth velocity ω_s is created with a symmetric time function as

$$\omega_s = a \left(-\frac{140}{T_s^7} t^7 + \frac{420}{T_s^6} t^6 - \frac{420}{T_s^5} t^5 + \frac{140}{T_s^4} t^4 \right), \tag{29}$$

where $a = 12.91$ and $T_s = 160$ are the time constant and the amplitude. Fig. 12a shows the rest-to-rest function behavior of (29). After running the same simulation with our defined T , we get the results in Fig. 12b (solid lines velocities). It is clear that the second case has smooth angular velocities, which are practical without large accelerations at the same time span of t_f . The angular displacement in Fig. 13 also demonstrate how the desired angles of the sphere get expanded with the same convergence trend (\mathbf{L}_o trajectory is the same as Fig. 9). Note that more complex functions for T to converge with different angular velocities without affecting the trajectories of \mathbf{L}_o and \mathbf{L}_s . Next, we analyze the patterns of the created curves \mathbf{L}_o for different desired spin angles with the same final configuration. In here, the final configuration is considered $\{u_{s,f}, v_{s,f}, u_{o,f}, v_{o,f}\} = \{3, 3.2, 0.6, 0.7\}$ besides different desired spin angles $\psi_f = \{-1.7, 1.3, 2.3\}$. To have these different curves approximately on same configuration, the ϵ_n and ϵ_r accuracies are chosen $\{0.045, 0.05\}$, $\{0.048, 0.034\}$ and

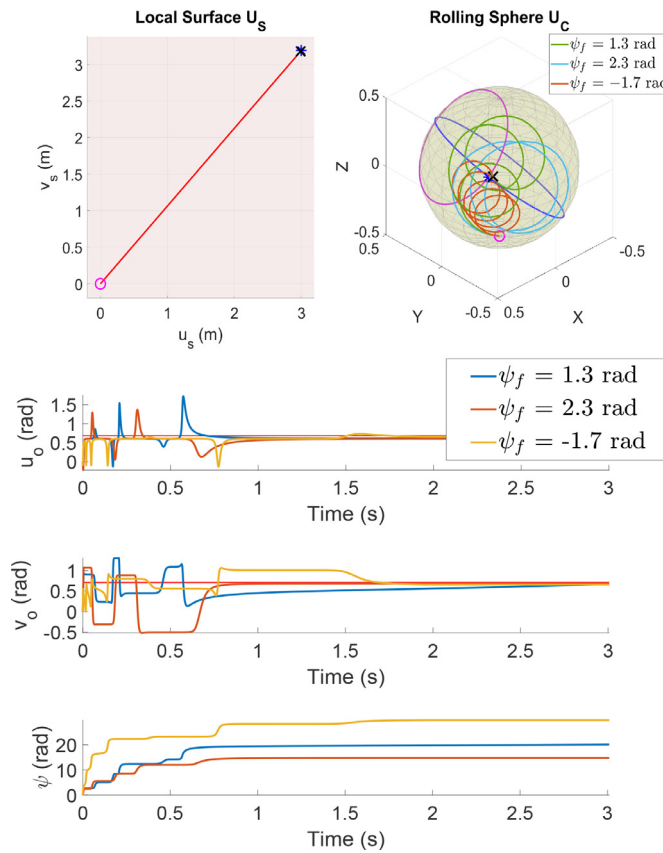


Fig. 14. Simulation results for different final spin angles $\psi_f = \{-1.7, 1.3, 2.3\}$.

$\{0.048, 0.05\}$ for $\psi_f = \{-1.7, 1.3, 2.3\}$ cases, in the given order. Also, simulation time is set for $t_f = 20$ s with $T = 1$. Note that the rest of the initial conditions are the same as in the previous case study.

Simulation results for different final spin angles are presented in Fig. 14. As can be seen, the algorithm converges successfully to desired states with different trajectories. The average process time for these simulations was about 40 s. Note that the average process time rise is due to exceptional computation for $\psi_f = -1.7$ rad. This occurs depending on the accuracy and how near the desired Ψ_f angles are to Ψ_0 which causes larger iterations (110 s process time).

For the case of $\psi_f = -1.7$ rad, the curve creates multiple loops to reach the desired goal. As mentioned before, the reason is that some spin angles exhibit slower convergence by the designed approach. To achieve faster convergence, we have to change the region of attraction for the desired configuration. This can be done by adding $+\frac{\pi}{2}$ to v -curve for the $\alpha_s(v_{o,f} + \frac{\pi}{2})$ arc-length-based input for this configuration. Since we have already guaranteed the convergence of the desired goal by the Theorem 1, we can have regional shifts in the $v_{o,f}$ and $u_{o,f}$. Note that this is an interesting property of the proposed algorithm since multiple solutions would be achievable for the same configuration by changing the time constant T , resulting to different regions of attraction on u - and v -curves.

7. Conclusion

In this paper, we presented a new approach for path planning of the spin-rolling sphere on a straight plane trajectory. The planning was mainly constructed by the introduced geometric control. In the beginning, the path planning problem is explained. Next, a virtual surface was designed to produce arc-length-based control inputs for the fully-actuated transformed model in [18]. The problem is imagined as manipulating a flexible rope-like curve on the sphere with a constant length. Thus, we proposed an iterative algorithm to tune the traversed curve toward desired configurations. The achieved simulations clarified the performance of the proposed planning approach.

The proposed algorithm produces smooth trajectories of the spin-rolling sphere. In this connection, it should be noted that the feedback-based planning algorithms [24–26,28] results in piece-wise smooth trajectories. Also, our approach does not require decomposition of the planning strategy into motion steps as featured in geometric phase shift methods [24,28,32,40]. Another advantage is that the resulting angular velocities of the spin-rolling sphere can shape to different desired convergence rates.

The current algorithm does the motion planning offline. Therefore, for applications that include direct/in-direct actuation of the sphere, a decoupled trajectory tracking control strategy should be utilized for dealing with perturbations that consider the dynamics of mechanical drivers to keep the sphere on the reference trajectory produced by the proposed motion planner. Alternatively, if the deviation becomes unrecoverable by sphere orientation, the path planning approach can be re-executed with the latest state of the system. In the future, we plan to generalize the problem to arbitrary convex surfaces and include perturbations using our geometric transformation and nilpotent approximation method together, which will enable online motion planning.

Availability of Data and Material

The data generated will be made available online.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by JSPS KAKENHI grant number JP21K20391 and partially by Japan Science and Technology Agency (JST) [Moonshot R & D Program] under Grant JPMJMS2034.

Appendix A. Ball-plate System Preliminaries

Fig. A.1 illustrates the rotating object and surface coordinates in the ball-plate system [20,32]. Here, we have Σ_o and Σ_s as the fixed frames on the rolling object (red sphere) and the plane. The frame Σ_s is fixed relative to other coordinate frames. It is assumed that the sphere with radius R_o is rotating with no sliding constraint. The local coordinate systems for the sphere and plane are considered

$$\begin{aligned} f_o : U_C \rightarrow \mathbb{R}^3 : c(u_o, v_o) &\mapsto (-R_o \sin u_o \cos v_o, R_o \sin v_o, -R_o \cos u_o \cos v_o), \\ f_s : U_S \rightarrow \mathbb{R}^3 : c(u_s, v_s) &\mapsto (u_s, v_s, 0), \end{aligned} \tag{A.1}$$

where $c(u_o, v_o) \in [-\pi, \pi]$ and $c(u_s, v_s)$ are contact parameters of the sphere and plane. For the sphere, we have [32]:

$$k_{nu}^o = k_{nv}^o = 1/R_o, \quad \tau_{gu}^o = \tau_{gv}^o = 0, \quad k_{gu}^o = \tan(v_o)/R_o, \quad k_{gv}^o = 0, \tag{A.2}$$

where $k_{nu}^o, k_{nv}^o, \tau_{gu}^o, \tau_{gv}^o, k_{gu}^o, k_{gv}^o$ are the normal curvature, geodesic torsion and geodesic curvature of the rotating body respect to u_o and v_o principle angle. The curvature terms of the plane surface U_s are

$$k_{nu}^s = k_{nv}^s = \tau_{gu}^s = \tau_{gv}^s = k_{gu}^s = k_{gv}^s = 0, \tag{A.3}$$

where $k_{nu}^s, k_{nv}^s, \tau_{gu}^s, \tau_{gv}^s, k_{gu}^s, k_{gv}^s$ are the normal curvature, geodesic torsion and the geodesic curvature of the fixed surface (plane) respect to u_s and v_s principle angles. Also, the ψ is the angle between the sphere and plane surfaces.

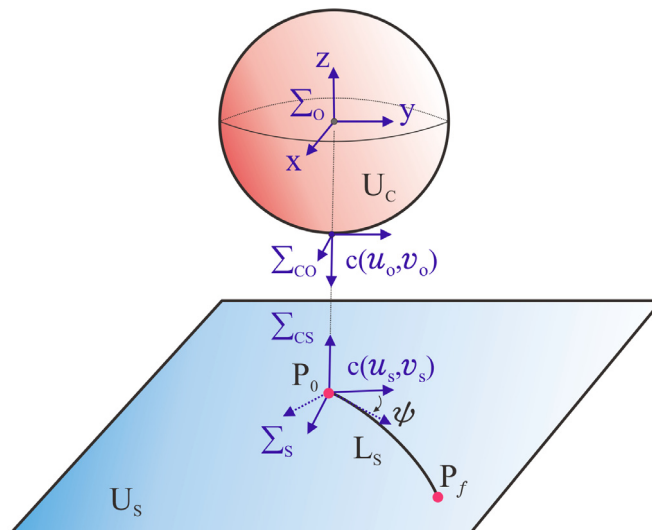


Fig. A.1. Kinematic model of rotating sphere. Note: ψ is the spin angle between sphere and plane surfaces.

Appendix B. Geodesic Torsion Design of the Virtual Surface

The rotating object and plane do not have the geodesic torsion, $\tau_g^* = -\beta_s$. It makes the kinematic model (1) uncontrollable [18]. Thus, a helicoid virtual surface with similar curvature properties to the rotating object (see Fig. 5) is proposed as

$$f_v : U_V \rightarrow R^3 : c(u_v, v_v) \mapsto (-R_v \sin u_v \cos v_v, R_v \sin v_v + R_t u_v, -R_v \cos u_v \cos v_v), \tag{B.1}$$

where R_v and R_t are defined by main spherical and sum of spherical and torsion radii, respectively. The curvature properties can be obtained as

$$k_n^v = \frac{1}{R_v}, \quad k_g^v = \frac{R_v \cos v_v \sin v_v}{R_v^2 \cos^2 v_v + R_t^2}, \quad \tau_g^v = \frac{1}{R_v^2} (R_v^2 \cos^2 v_v + R_t^2)^{\frac{1}{2}} \tag{B.2}$$

where k_n^v, k_g^v, τ_g^v are normal curvature, geodesic curvature and geodesic torsion. We here mainly care about $\tau_g^v(R_v, R_t, v_v)$ and $k_n^v(R_v)$ to understand relation of geodesic torsion design. That's why we related them with separate corresponding radii. To prove that this surface let us manipulate τ_g by using R_t , we consider $R_t = nR_v$ where $n > 1$, which results

$$\tau_g^v = \frac{1}{R_v} (\cos^2 v_v + n^2)^{\frac{1}{2}}. \tag{B.3}$$

Theorem 2. At an arbitrary point P on a surface, geodesic torsion relation with normal curvature is defined as [46]

$$\tau_g = \frac{1}{2} (k_{nu} - k_{nv}) \sin 2\Omega \tag{B.4}$$

where k_{nu} and k_{nv} are the principle curvatures. Also, Ω is the counterclockwise angle from the direction of minimum curvature k_{nu} on the tangent plane.

By relying on the Theorem 2, it is clear that sphere tangent plane angle Ω is $\pi/4$. Now, we can do some algebraic operations on (B.4) with considering $k_{nu} = 1/R_v$ and (B.3),

$$k_{nv} = 1/R_v \left[1 + (\cos^2 v_v + n^2)^{\frac{1}{2}} \right] \tag{B.5}$$

where k_{nv} is the second principle curvature of the virtual surface. This supports our assumption that designed geodesic torsion is inverse of sphere radius with similar unity of normal curvature $R_t = nR_v$. We do this for sake of separating the radius of geodesic torsion and normal curvature in (11) as we use two separate surfaces (sphere and helicoid) in arc-length-based inputs.

Appendix C. Phase I Directional Update

We can separate the sphere regions to four by $c(u_{o,f}, v_{o,f})$ cutting planes (see example in Fig. C.1) for the desired configuration where ζ_q is calculated for its directional updates. Because we need to find directional updates in different G_f angles on the plane, we apply a spherical rotational transformation to ease our regional computation. To find corresponding local points' e.g., Ψ_n , right location for the directional update from plane angle G_f with respect to v_s , we rotate the local contact coordinate $c(u_o, v_o)$ of the sphere around z_o axis ($G'_f = G_f - \pi/4$ as shown in Fig. C.1) as follows

$$\begin{bmatrix} x_o \\ y_o \end{bmatrix} = \begin{bmatrix} \sin u_o^r \cos v_o^r \\ \sin v_o^r \end{bmatrix} = \begin{bmatrix} \cos G'_f & -\sin G'_f \\ \sin G'_f & \cos G'_f \end{bmatrix} \begin{bmatrix} \sin u_o \cos v_o \\ \sin v_o \end{bmatrix} \tag{C.1}$$

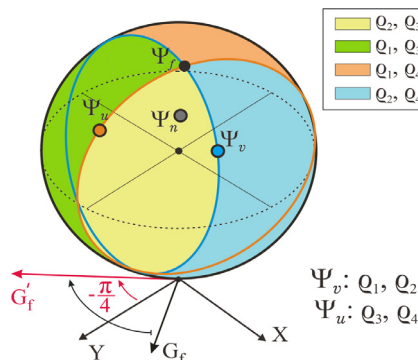


Fig. C.1. The sectioned sphere based on the Ψ_v and Ψ_u cutting planes for directional updates of ζ_q . Note that Ψ_v and Ψ_u are on Q_2 and Q_3 as an example.

where $c(u_o^r, v_o^r)$ is the new angles and sphere's contact coordinate is $[x_o, y_o, z_o] = [\sin u_o \cos v_o, \sin v_o, \cos u_o \cos v_o]$ (details are given in [Appendix A](#)). After re-ordering the (C.1), we have algebraic solution by

$$\begin{aligned} v_o^r &= \sin^{-1} \left[-\sin G'_f \sin u_o \cos v_o + \cos G'_f \sin v_o \right], \\ u_o^r &= \sin^{-1} \left[(\cos G'_f \sin u_o \cos v_o + \sin G'_f \sin v_o) / \cos v_o^r \right]. \end{aligned} \tag{C.2}$$

Next, to avoid the numerical solution of (C.2), we present it in algebraic form depending on $c(u_o, v_o)$ location by

$$\left\{ \begin{aligned} & \left\{ \begin{aligned} u_o^r &\leftarrow -u_o^r, \\ v_o^r &\leftarrow \pi - v_o^r, \end{aligned} \text{ for } \left[(0 \leq |v_o| \leq \frac{\pi}{2} \ \& \ \frac{\pi}{2} \leq |u_o| \leq \pi) \mid \left(\frac{\pi}{2} < |v_o| \leq \pi \ \& \ 0 \leq |u_o| < \frac{\pi}{2} \right) \right] \\ & \left\{ \begin{aligned} u_o^r &\leftarrow u_o^r, \\ v_o^r &\leftarrow v_o^r, \end{aligned} \text{ for } \left[(0 < |v_o| < \frac{\pi}{2} \ \& \ 0 < |u_o| < \frac{\pi}{2}) \mid \left(\frac{\pi}{2} < |v_o| < \pi \ \& \ \frac{\pi}{2} < |u_o| < \pi \right) \right] \end{aligned} \right. \end{aligned} \tag{C.3}$$

The operations (C.2)-(C.3) are applied to rotate Ψ_f, Ψ_u, Ψ_v and Ψ_n relative to angle G_f to find them always in one G'_f direction. Now, the sectioned sphere as [Fig. C.1](#) is obtained by cutting planes Ψ_v and Ψ_u where they create $\{\varrho_1, \varrho_2\}$ and $\{\varrho_3, \varrho_4\}$ regions, respectively. Note that for nearest point Ψ_n on \mathbf{L}_o , the corresponding ϱ are shown as $\varrho_{n1}, \varrho_{n2}, \varrho_{n3}, \varrho_{n4}$. We design our directional updates by comparing Ψ_n with Ψ_u, Ψ_v as following computation

```

Calculate  $G'_f$  depending on the  $\{u_{s,f}, v_{s,f}\}$ 
Calculate rotated coordinates of  $\Psi_f, \Psi_u, \Psi_v$  and  $\Psi_n$  by (C.3)
Calculate  $Q_f^{zy}, Q_f^{zx}, Q_n^{zy}$  and  $Q_n^{zx}$  according to (20)
if  $\varrho_2 = 1 \ \& \ \varrho_4 = 1$  then                                     > Exceptional regional updates
  if only  $\varrho_{n2} = 1$  then
     $\zeta_q(k) \leftarrow +\zeta_q(k-1)$ 
  else if  $(\varrho_{n2} = 1 \ \& \ \varrho_{n4} = 1) \mid \mid$  (only  $\varrho_{n4} = 1$ ) then
    if  $v_{o,f}^r \geq 0$  then
      if  $u_{o,f}^r \geq 0$  then
         $\zeta_q(k) \leftarrow +\zeta_q(k-1)$ 
      else
         $\zeta_q(k) \leftarrow -\zeta_q(k-1)$ 
      end if
    else
       $\zeta_q(k) \leftarrow -\zeta_q(k-1)$ 
    end if
  end if
else if  $(\varrho_{n2} = 1 \ \& \ \varrho_{n3} = 1) \mid \mid$   $(\varrho_3 = 1 \ \& \ \varrho_{n2} = 1 \ \& \ \varrho_{n4} = 1)$  then
  if  $u_{o,f}^r \geq 0$  then
     $\zeta_q(k) \leftarrow +\zeta_q(k-1)$ 
  else
     $\zeta_q(k) \leftarrow -\zeta_q(k-1)$ 
  end if
else if  $\varrho_1 = \varrho_2 = \varrho_3 = \varrho_4 = 0$  then
  if  $v_{o,f}^r \geq 0$  then
     $\zeta_q(k) \leftarrow -\zeta_q(k-1)$ 
  else
     $\zeta_q(k) \leftarrow +\zeta_q(k-1)$ 
  end if
else
  if only  $(\varrho_{n1} = 1) \mid \mid$   $(\varrho_{n4} = 1)$  then                                     > Normal regional updates
    if  $v_{o,f}^r \geq 0$  then
       $\zeta_q(k) \leftarrow +\zeta_q(k-1)$ 
    else
       $\zeta_q(k) \leftarrow -\zeta_q(k-1)$ 
    end if
  else if only  $(\varrho_{n2} = 1) \mid \mid$   $(\varrho_{n3} = 1)$  then
    if  $v_{o,f}^r \geq 0$  then
       $\zeta_q(k) \leftarrow -\zeta_q(k-1)$ 
    else
       $\zeta_q(k) \leftarrow +\zeta_q(k-1)$ 
    end if
  end if
end if

```

where Q_{ni} presents the nearest point Ψ_n location on i -th section of the spherical surface ($i \in [1, 4]$) with respect to Ψ_f . Also, $Q_{ni} = 1$ means the following argument is true. Note that this complexity in sectioned spherical surface, as shown in Fig. C.16, is due to rolling surface that creates different signs of updates of $\zeta_q(k)$ in (19) based on $\{\Psi_0, \Psi_f, \Psi_n\}$. In additions, the normal updates in the algorithm find Ψ_n location on the spherical surface and compare it to the location of desired goal Ψ_f to give correct sign update for $\zeta_q(k)$. However, exceptional updates are designed for exiting from trajectory loops of L_0 for Ψ_n and bringing the L_0 curve toward the location of Ψ_0 .

Appendix D. Proof of Closed Set

The following proof for the closed set is inspired by similar studies of general convergence for the search algorithms [48,49]. First, let $\mathbf{x}_k \rightarrow \mathbf{x}_0$ and $\mathbf{d}_k \rightarrow \mathbf{d}_0$ be the convergence to desired values as $k \rightarrow \infty$ via sequences of $\{\mathbf{x}_k\}_{k=1}^{\infty}$ and $\{\mathbf{d}_k\}_{k=1}^{\infty}$. Also, suppose $\{\mathbf{y}_k\}_{k=1}^{\infty}$ is a sequence where $\mathbf{y} \in \Gamma(\mathbf{x}_k, \mathbf{d}_k)$ for all k while it has convergence of $\mathbf{y}_k \rightarrow \mathbf{y}_0$ as $k \rightarrow \infty$. We want to illustrate that $\mathbf{y}_0 \in \Gamma(\mathbf{x}_0, \mathbf{d}_0)$ for having a closed set.

By taking the norm operation from both sides of $\mathbf{y}_k = \mathbf{x}_k + \mathbf{h}_k \mathbf{d}_k$ in k th iteration, we have $\|\mathbf{d}_k\| = 1$ and

$$\mathbf{h}_k = \begin{bmatrix} \|\mathbf{y}_k(1) - \mathbf{x}_k(1)\| & \dots \\ \dots & \|\mathbf{y}_k(n) - \mathbf{x}_k(n)\| \end{bmatrix} \rightarrow \mathbf{h}^* = \begin{bmatrix} \|\mathbf{y}_0(1) - \mathbf{x}_0(1)\| & \dots \\ \dots & \|\mathbf{y}_0(n) - \mathbf{x}_0(n)\| \end{bmatrix}.$$

Hence, it implies that $\mathbf{y}_0 = \mathbf{x}_0 + \mathbf{h}^* \mathbf{d}_0$.

Now, to show \mathbf{y}_0 minimizes f along the $\mathbf{x}_0 + \mathbf{h} \mathbf{d}_0$, we use the set valued function (27) as follows

$$f(\mathbf{y}_k) \leq f(\mathbf{x}_k + \mathbf{h} \mathbf{d}_k).$$

Therefore, by the continuity of f , $k \rightarrow \infty$ leads to $f(\mathbf{y}_0) \leq f(\mathbf{x}_0 + \mathbf{h} \mathbf{d}_0)$ for all \mathbf{h} which shows

$$f(\mathbf{y}_0) \leq \min_{0 < \mathbf{h} < \infty} \{f(\mathbf{x}_0 + \mathbf{h} \mathbf{d}_0)\}.$$

This proves the definition of $\mathbf{y}_0 \in \Gamma(\mathbf{x}_0, \mathbf{d}_0)$ for having a closed set.

References

- [1] B. Kiss, J. Lévine, B. Lantos, On motion planning for robotic manipulation with permanent rolling contacts, *Int. J. Robot. Res.* 21 (5-6) (2002) 443–461.
- [2] L. Cui, J. Sun, J. Dai, In-hand forward and inverse kinematics with rolling contact, *Robotica* 35 (12) (2017) 2381–2399.
- [3] S. Yuan, L. Shao, C.L. Yako, A. Gruebele, J.K. Salisbury, Design and control of roller grasper V2 for in-hand manipulation, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9151–9158.
- [4] S.A. Tafirishi, “RollRoller” novel spherical mobile robot basic dynamical analysis and motion simulations, University of Sheffield, Sheffield, UK, 2014 Master’s thesis.
- [5] S.A. Tafirishi, M. Svinin, E. Esmaeilzadeh, M. Yamamoto, Design, modeling, and motion analysis of a novel fluid actuated spherical rolling robot, *ASME J. Mech. Robot.* 11 (4) (2019) 041010.
- [6] M. Ishikawa, R. Kitayoshi, T. Sugie, Volvot: a spherical mobile robot with eccentric twin rotors, in: *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2011, pp. 1462–1467.
- [7] Y. Karavaev, I. Mamaev, A. Kilin, E. Pivovarova, Spherical rolling robots: different designs and control algorithms, in: *Robots in Human Life: Proc. of the 23rd Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2020, Moscow, Aug 24–26 2020)*, 2020, pp. 195–202.
- [8] A. Borisov, A. Kilin, I. Mamaev, How to control Chaplygin sphere using rotors, *Regul. Chaotic Dyn.* 17 (3) (2012) 258–272.
- [9] A. Borisov, A. Kilin, I. Mamaev, How to control the Chaplygin ball using rotors. II, *Regul. Chaotic Dyn.* 18 (1) (2013) 144–158.
- [10] M. Svinin, A. Morinaga, M. Yamamoto, On the dynamic model and motion planning for a spherical rolling robot actuated by orthogonal internal rotors, *Regul. Chaotic Dyn.* 18 (1) (2013) 126–143.
- [11] P. Fankhauser, C. Gwerder, Modeling and control of a ballbot, Eidgenössische Technische Hochschule Zürich, 2010 B.S. thesis.
- [12] J. Johnson, R. Senthilnathan, M. Negi, R.G. Patel, A. Bhattacherjee, A fuzzy logic-in-loop control for a novel reduced height ballbot prototype, *Procedia Comput. Sci.* 133 (2018) 960–967.
- [13] B. Sümer, M. Sitti, Rolling and spinning friction characterization of fine particles using lateral force microscopy based contact pushing, *J. Adhes. Sci. Technol.* 22 (5-6) (2008) 481–506.
- [14] E. Diller, M. Sitti, et al., Micro-scale mobile robotics, *Found. Trends Robot.* 2 (3) (2013) 143–259.
- [15] A. Fernández-Pacheco, R. Streubel, O. Fruchart, R. Hertel, P. Fischer, R.P. Cowburn, Three-dimensional nanomagnetism, *Nat. Commun.* 8 (2017) 15756.
- [16] V. Jurdjevic, The geometry of the plate-ball problem, *Arch. Ratio. Mech. Anal.* 124 (4) (1993) 305–328.
- [17] A. Marigo, A. Bicchi, Rolling bodies with regular surface: controllability theory and applications, *IEEE Trans. Autom. Control* 45 (9) (2000) 1586–1599.
- [18] S.A. Tafirishi, M. Svinin, M. Yamamoto, Darboux-frame-based parametrization for a spin-rolling sphere on a plane: a nonlinear transformation of underactuated system to fully-actuated model, *Mech. Mach. Theory* 164 (2021) 104415.
- [19] N. Sankar, V. Kumar, X. Yun, Velocity and acceleration analysis of contact between three-dimensional rigid bodies (1996).
- [20] D.J. Montana, The kinematics of contact and grasp, *Int. J. Robot. Res.* 7 (3) (1988) 17–32.
- [21] L. Cui, J.S. Dai, A Darboux-frame-based formulation of spin-rolling motion of rigid objects with point contact, *IEEE Trans. Robot.* 26 (2) (2010) 383–388.
- [22] I. Bizyaev, A. Borisov, I. Mamaev, Different models of rolling for a robot ball on a plane as a generalization of the Chaplygin ball problem, *Regul. Chaotic Dyn.* 24 (5) (2019) 560–582.
- [23] Z. Woodruff, K. Lynch, Second-order contact kinematics between three-dimensional rigid bodies, *J. Appl. Mech.* 86 (8) (2019).
- [24] H. Date, M. Sampei, M. Ishikawa, M. Koga, Simultaneous control of position and orientation for ball-plate manipulation problem based on time-state control form, *IEEE Trans. Robot.* 20 (3) (2004) 465–480.
- [25] G. Oriolo, M. Vendittelli, A framework for the stabilization of general nonholonomic systems with an application to the plate-ball mechanism, *IEEE Trans. Robot.* 21 (2) (2005) 162–175.
- [26] T. Das, R. Mukherjee, Exponential stabilization of the rolling sphere, *Automatica* 40 (11) (2004) 1877–1889.
- [27] F. Alouges, Y. Chitour, R. Long, A motion-planning algorithm for the rolling-body problem, *IEEE Trans. Robot.* 5 (26) (2010) 827–836.
- [28] R. Mukherjee, T. Das, Feedback stabilization of a spherical mobile robot, in: *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Sys.*, Vol. 3, 2002, pp. 2154–2162.
- [29] Z. Woodruff, S. Ren, K. Lynch, Motion planning and feedback control of rolling bodies, *IEEE Access* 8 (2020) 31780–31791.
- [30] T. Ohsawa, Geometric kinematic control of a spherical rolling robot, *J. Nonlinear Sci.* 30 (1) (2020) 67–91.

- [31] V. Putkaradze, S. Rogers, On the optimal control of a rolling ball robot actuated by internal point masses, *J. Dyn. Syst. Meas. Control* 142 (5) (2020) 051002.
- [32] Z. Li, J. Canny, Motion of two rigid bodies with rolling constraint, *IEEE Trans. Robot. Autom.* 6 (1) (1990) 62–72.
- [33] R. Mukherjee, M.A. Minor, J.T. Pukrushpan, Motion planning for a spherical mobile robot: revisiting the classical ball-plate problem, *ASME J. Dyn. Syst. Meas. Control* 124 (4) (2002) 502–511.
- [34] M. Svinin, S. Hosoe, Motion planning algorithms for a rolling sphere with limited contact area, *IEEE Trans. Robot.* 24 (3) (2008) 612–625.
- [35] A. Kilin, E. Pivovarova, T. Ivanova, Spherical robot of combined type: dynamics and control, *Regul. Chaotic Dyn.* 20 (6) (2015) 716–728.
- [36] Y. Bai, M. Svinin, M. Yamamoto, Dynamics-based motion planning for a pendulum-actuated spherical rolling robot, *Regul. Chaotic Dyn.* 23 (4) (2018) 243–259.
- [37] A. Arthurs, G. Walsh, On Hammersley's minimum problem for a rolling sphere, in: *Math. Proc. Cambridge Philos. Soc.*, Vol. 99, Cambridge University Press, 1986, pp. 529–534.
- [38] Y. Sachkov, Maxwell strata and symmetries in the problem of optimal rolling of a sphere over a plane, *Sb. Math* 201 (7) (2010) 1029–1051.
- [39] A. Mashtakov, Y. Sachkov, Extremal trajectories and the asymptotics of the Maxwell time in the problem of the optimal rolling of a sphere on a plane, *Sb. Math* 202 (9) (2011) 1347–1371.
- [40] A. Morinaga, M. Svinin, M. Yamamoto, A motion planning strategy for a spherical rolling robot driven by two internal rotors, *IEEE Trans. Robot.* 30 (4) (2014) 993–1002.
- [41] I. Beschatnyi, The optimal rolling of a sphere, with twisting but without slipping, *Mat. Sb.* 205 (2) (2014) 157.
- [42] L. Cui, J. Dai, *Sliding-Rolling Contact and In-hand Manipulation*, World Scientific, 2020.
- [43] S.A. Tafrishi, A.A. Ravankar, J.V.S. Luces, Y. Hirata, A novel assistive controller based on differential geometry for users of the differential-drive wheeled mobile robots, in: *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 5755–5761.
- [44] A. Tafrishi, M. Svinin, M. Yamamoto, Y. Hirata, Path Planning of Spin-Rolling Sphere on a Plane, *YouTube*, 2021. <https://www.youtube.com/watch?v=7JK3VtT0aqs>
- [45] M.P.d. Carmo, *Differential Geometry of Curves and Surfaces*, second ed., Prentice-Hall, 1976.
- [46] E. Cartan, *Riemannian Geometry in an Orthogonal Frame*, first ed., World Scientific Pub Co Inc, 2002.
- [47] K. Tapp, *Differential Geometry of Curves and Surfaces*, Springer, New York, 2016.
- [48] W. Zangwill, *Nonlinear Programming: A Unified Approach*, Vol. 196, Prentice-Hall Englewood Cliffs, NJ, 1969.
- [49] H.H. J. Ye, Z. Zhu, *Lecture Notes on Computing and Optimization*, ORF363, Princeton University (2015).