IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 1, 2020, accepted February 8, 2020, date of publication February 12, 2020, date of current version February 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973416

Motion Planning and Feedback Control of Rolling Bodies

J. ZACHARY WOODRUFF, (Member, IEEE), SHUFENG REN, AND KEVIN M. LYNCH, (Fellow, IEEE)
Mechanical Engineering Department, Northwestern University, Evanston, IL 60208, USA
Center for Robotics and Biosystems, Northwestern University, Evanston, IL 60208, USA

Corresponding author: J. Zachary Woodruff (jzwoodruff@u.northwestern.edu)

This work supported in part by the NSF Graduate Research Fellowship Program under Grant DGE-1324585, and in part by the NSF under Grant IIS-1527921.

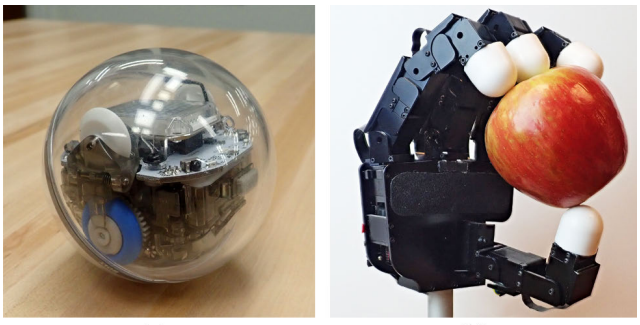
ABSTRACT This paper examines the problem of planning and stabilizing the trajectory of one smooth body rolling on the surface of another. The two control inputs are the angular velocity of the moving body about two orthogonal axes in the contact tangent plane; spinning about the contact normal is not allowed. To achieve robustness and computational efficiency, our approach to trajectory planning is based on solving a series of optimization problems of increasing complexity. To stabilize the trajectory in the face of perturbations, we use a linear quadratic regulator. We apply the approach to examples of a sphere rolling on a sphere and an ellipsoid rolling on an ellipsoid. Finally, we explore the robustness and performance of the motion planner. Although the planner is based on non-convex optimization, in practice the planner finds solutions to nearly all randomly-generated tasks, and the solution trajectories are smoother and shorter than those found in previous work in the literature.

INDEX TERMS Robot kinematics, motion planning, controllability, state feedback, rolling.

I. INTRODUCTION

This paper examines the problem of planning and stabilizing the trajectory of one smooth body rolling on the surface of another. This is relevant for systems such as a ball-type mobile robot rolling over smooth terrain (Figure 1(a)) or a robot hand planning multi-finger rolling motions to reorient an object (Figure 1(b)). Much research on rolling motion planning has been limited to specialized geometries such as planes and spheres. In this work we present a method to generate motion plans and stabilizing feedback controllers for general, smooth, three-dimensional objects in rolling contact.

The two rolling objects are modeled as a well-known nonholonomic system with five degrees of freedom and two controls. Collectively the contact configuration is written $q = (u_1, v_1, u_2, v_2, \psi)$, which gives the contact location (u_1, v_1) on object 1, (u_2, v_2) on object 2, and the angle of "spin" ψ between contact frames (Figure 2). The two control inputs are the angular velocities $\Omega = (\omega_x, \omega_y)$ of the moving body about two orthogonal axes in the contact tangent plane; spinning about the contact normal is not allowed. We refer to the no-spin assumption as "pure rolling." Since we assume the velocities are directly controlled, we refer to the equations of motion as the "first-order kinematics."



(a) (b)

FIGURE 1. Examples of robot tasks that can be modeled as objects in rolling contact. (a) A ball-type mobile robot on a smooth surface. (b) Robot fingers rolling over a smooth object.

Our approach to trajectory planning is based on solving a series of optimization problems of increasing complexity. We first solve a convex problem that uses the two rolling velocity inputs to drive two of the five configuration variables directly to their desired values. This motion serves as the initial trajectory guess for direct-collocation constrained

The associate editor coordinating the review of this manuscript and approving it for publication was Fei Chen.

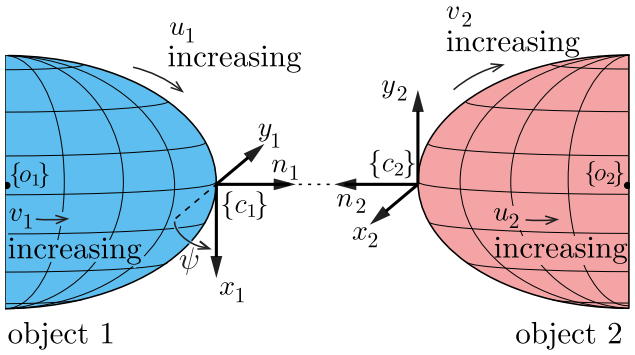


FIGURE 2. Objects 1 and 2 contact at the origin of frames $\{c_1\}$ and $\{c_2\}$, but are shown separated for clarity. Collectively the contact configuration is written $q = (u_1, v_1, u_2, v_2, \psi)$. The surfaces of objects 1 and 2 are orthogonally parameterized by (u_1, v_1) and (u_2, v_2), respectively. At the point of contact, the unit x_i- and y_i-axes of the coordinate frame $\{c_i\}$ are in the direction of increasing u_i (and constant v_i) and increasing v_i (and constant u_i), respectively, and the contact normal n_i is the cross product $x_i \times y_i$. Rotating frame $\{c_2\}$ by ψ about the n_1-axis of frame $\{c_1\}$ aligns the x_2-axis of frame $\{c_2\}$ and the x_1-axis of frame $\{c_1\}$. The controls for pure-rolling (no relative spin about the contact normal) are the relative angular velocities $\Omega = (\omega_x, \omega_y)$ about the x_2- and y_2-axes of the contact frame $\{c_2\}$ [1].

optimization. Using this initial guess, the optimization solves the full five-dimensional trajectory-planning problem. The optimization first solves for a trajectory history that is represented coarsely, using a small number of state and control segments. The solved-for controls are then simulated by a more accurate, higher-order numerical integration method than the integrator implicit in the constraints in the nonlinear optimization. If the simulated trajectory satisfies the error tolerances, the problem is solved. If not, the previous solution is used as an initial guess, the number of state and control segments is increased, and the optimization is called again.

The motion planner is structured this way to balance three goals: 1) increasing the likelihood of finding a solution; 2) decreasing the computation time required to find a solution; and 3) optimizing the quality of the solution, as measured by the trajectory length and control cost. In our tests, an initial optimization with a fine control discretization often takes an unnecessarily long time to converge or even fails to converge to a feasible solution. The coarse initial guess followed by successive refinement yields higher-quality solutions faster and more consistently. The iterative refinement process acts as a form of regularization.

To stabilize a planned trajectory to small perturbations, we use linear feedback control based on a linear quadratic regulator. For this to be successful, the linearized trajectory must be controllable, so we examine the controllability of rolling trajectories and provide examples of uncontrollable trajectories.

Our primary contributions are a robust motion planner for generating rolling motions of general smooth objects and an approach to stabilize those trajectories.

A. PAPER OUTLINE

Section II reviews previous work related to this paper. Section III summarizes the rolling kinematics, and Section IV formally states the problem we are solving. Section V outlines the design of the motion planner, and Section VI analyzes the controllability of rolling trajectories. Section VII demonstrates planning for a sphere rolling on a sphere and an ellipsoid on an ellipsoid, applying feedback controllers to stabilize the planned trajectories from perturbations to the initial configurations. Section VIII analyzes the robustness and performance of the planner for random goal states and different methods of generating initial trajectory guesses. Section IX summarizes the results and describes planned future work. Some background on differential geometry and derivations of terms used in the kinematics expressions are given in the Appendix.

II. RELATED WORK

A. MODELING OF ROLLING SURFACES

First-order kinematics addresses the rolling problem where the relative contact velocities are directly controlled. The second-order kinematics is a generalization of the first-order model where the relative accelerations at the contact are controlled. Dynamic rolling assumes that forces and torques are controlled.

Cai and Roth derive the first- and second-order contact kinematic equations for two objects in contact [2]. They focus on the special cases of pure translation and pure rotation about the contact point. They only consider the four-dimensional evolution of the contact points on the objects, not the full five-dimensional configuration.

Montana derives the first-order contact kinematics for two 3D objects in contact [3]. His method models the full five-dimensional configuration space, but it is not easily generalized to second-order kinematics or dynamics. First- and second-order contact equations were derived by Sarkar et al. in [4] and published again in later works [5], [6]. Errors in the published equations for second-order contact kinematics in [4]–[6] were corrected in our recent work [1]. Each of [1], [3]–[6] assumes an orthogonal parameterization, as shown in Figure 2. Chitour et al. survey results on the pure rolling problem for surfaces represented as manifolds and analyzed using Riemannian geometry and geometric control theory [7].

B. MOTION PLANNING FOR ROLLING SYSTEMS

First- and second-order roll-slide kinematics, as discussed above, allow sliding at the contact, but we focus on planning for first-order systems in pure-rolling contact. This is a well-known driftless nonholonomic system, where the rolling constraints do not necessarily translate to constraints on the achievable relative configuration.

Lafferriere and Sussmann give a method for motion planning and feedback control for nonholonomic systems without drift, and the methods are exact for systems with nilpotent, or feedback-nilpotentizable, Lie algebras [8]. Murray provides a method of finding a nilpotent basis for nonholonomic systems and shows how to generate plans for systems including a disk rolling on a plane [9]. The general pure-rolling

problem does not satisfy the nilpotent condition, so Oriolo and Vendittelli generalize the method and present an algorithm based on nilpotent approximation and iterative steering to achieve asymptotic stability for the plate-ball system [10].

Murray and Sastry outline a special class of nilpotentizable, nonholonomic systems called chained-form systems. They introduce sufficient conditions to check if a system can be converted to chained form, and they give a method to plan motions between arbitrary states for these systems (e.g., a kinematic car or a car pulling a trailer) [11]. Fliess et al. outline methods to represent nonlinear systems as differentially flat [12]. Such systems are amenable to simplified motion planning methods. Bicchi and Sorrentino show that the rolling system cannot be put into chained form and is not differentially flat, so those methods cannot be applied [13].

There are many works on motion planning for rolling systems that assume special geometries such as planes and spheres. Li and Canny derive the first-order contact equations for rolling objects parameterized by orthogonal coordinate systems, analyze the controllability properties, and provide a geometric motion planning algorithm for a sphere on a plane [14]. Marigo and Bicchi plan motions for general surfaces on a plane in the presence of obstacles using local approximations of Li and Canny's method [15]. Alouges et al. demonstrate the use of numerical continuation methods to generate open-loop motion plans for a general surface rolling on a plane without slipping [16]. Rehan and Reyhanoglu derive geometric planners for a sphere rolling on a smooth surface and demonstrate the method for a sphere rolling on a plane and on another sphere [17].

C. ROLLING CONTROLLABILITY AND FEEDBACK

A pure-rolling system is locally controllable from a given initial configuration if the set of locally reachable configurations, using only the two controls, is five dimensional. Li and Canny study the controllability of rolling bodies by taking Lie brackets of the rolling vector fields to generate higher-order vector fields [14]. They conclude that a sphere can reach any contact configuration on the plane by pure rolling, and that a sphere can reach any contact configuration by pure rolling on another sphere if their radii are different. For more general body geometries, however, deriving symbolic Lie bracket vector fields is cumbersome.

Marigo and Bicchi study the controllability of rolling bodies with regular surfaces, derive admissibility conditions for rolling contacts, and define necessary conditions for the reachability of rolling contacts [18]. They provide an in-depth analysis of the types of surfaces and initial conditions that cause the reachable set to drop from five to two dimensions. Agrachev and Sachkov (Section 24.4 of [19]) show that two objects in rolling contact are controllable when their local Gaussian curvatures are not equal. When the local Gaussian curvatures are equal, the three nonholonomic constraints become integrable, reducing the reachable set to a two-dimensional subset of the full five-dimensional configuration space. Krakowski et al. provide examples of when the rolling system fails to be controllable [20]. Feedback stabilization of rolling is addressed by Walsh *et al.*, who present a control law to exponentially stabilize linearized trajectories [21]; Sarkar *et al.* who demonstrate the use of feedback linearization to control dynamic rolling motions for two planes in contact with a sphere [5]; and Choudhury and Lynch, who stabilize the orientation of a ball rolling in an ellipsoidal dish actuated along a single degree of freedom [22].

III. ROLLING KINEMATICS

In this paper an object is a two-dimensional surface S embedded in 3D space. An open, connected subset of a surface S is defined as S_k. For a given S_k, the surface is parameterized by the coordinates $(u_k, v_k) \in U_k \subset \mathbb{R}^2$, and the shape is given by $f_k : U_k \to \mathbb{R}^3 : (u_k, v_k) \mapsto (x_k, y_k, z_k)$, where the (x_k, y_k, z_k) coordinates are expressed in a frame fixed to the body. The triplet (S_k, f_k, U_k) is called a coordinate chart, and a set of coordinate charts is called an atlas for S if the union of the S_k fully covers the surface S.

Throughout this paper we assume that rolling motion is confined to a single coordinate chart of object 1, (S_1, f_1, U_1), and a single coordinate chart of object 2, (S_2, f_2, U_2).

An example coordinate chart for a sphere covers all points of the sphere except for the poles, with

$$f : U \to \mathbb{R}^3 : (u, v) \mapsto (\rho \sin(u)\cos(v), \rho \sin(u)\sin(v), \rho \cos(u)), \quad (1)$$

where ρ is the radius of the sphere, u satisfies $0 < u < \pi$, and v satisfies $-\pi < v < \pi$.

It is assumed that f is continuous up to the second derivative (class C^2) so that the local contact geometries (contact frames and curvature associated with the first and second derivatives of f, respectively) are uniquely defined. We also assume that coordinate charts are orthogonal ($\frac{\partial f}{\partial u} \cdot \frac{\partial f}{\partial v} = 0$). Any smooth, regular surface can be locally represented this way.

The configuration space of objects in rolling contact (see Figure 2) can be parameterized by $q = (u_1, v_1, u_2, v_2, \psi)$, where $U_1 = (u_1, v_1)$ describes the contact point on the surface of object 1, $U_2 = (u_2, v_2)$ describes the contact point on the surface of object 2, and ψ describes the angle of "spin" between contact frames $\{c_1\}$ and $\{c_2\}$ about their common normal.

The linear velocity relating the relative motion between objects expressed in $\{c_2\}$ can be written as $V = (V_x, V_y, V_z)$. Similarly, the relative angular velocity at the contact expressed in $\{c_2\}$ can be written as $\omega = (\omega_x, \omega_y, \omega_z)$. The controls for pure rolling are the relative angular velocities $\Omega = (\omega_x, \omega_y)$ about the x- and y-axes of the contact frame at object 2, which defines the contact tangent plane. We use the first-order kinematics derived by Sarkar et al. in [4] with the pure-rolling assumptions applied ($V_x = V_y = V_z = \omega_z = 0$). These are equivalent to the equations in [3] for the first-order analysis, but are chosen to allow for direct extension to

second-order planning in future work. The pure-rolling kinematics reproduced from [4] are

$$\begin{aligned}\dot{U}_1 &= (\sqrt{\mathbf{G}_1})^{-1}\mathbf{R}_\psi(\widetilde{\mathbf{H}}_1+\mathbf{H}_2)^{-1}\begin{bmatrix}-\omega_y\\ \omega_x\end{bmatrix},\\ \dot{U}_2 &= (\sqrt{\mathbf{G}_2})^{-1}(\widetilde{\mathbf{H}}_1+\mathbf{H}_2)^{-1}\begin{bmatrix}-\omega_y\\ \omega_x\end{bmatrix},\\ \dot{\psi} &= \sigma_1\Gamma_1\dot{U}_1+\sigma_2\Gamma_2\dot{U}_2,\end{aligned} \tag{2}$$

where $\mathbf{G}_i$ is the metric tensor of object i, the 2×2 rotation matrix $\mathbf{R}_\psi$ is defined as

$$\mathbf{R}_\psi = \begin{bmatrix}\cos(\psi) & -\sin(\psi)\\ -\sin(\psi) & -\cos(\psi)\end{bmatrix},$$

$\mathbf{H}_i$ is a 2×2 matrix that gives the curvature of the surface, $\widetilde{\mathbf{H}}_1$ is defined as $\widetilde{\mathbf{H}}_1 = \mathbf{R}_\psi\mathbf{H}_1\mathbf{R}_\psi$, the scalar σ_i is defined as $\sigma_i = \sqrt{g_{22,i}/g_{11,i}}$ where $g_{11,i}$ and $g_{22,i}$ are the diagonal entries of the metric tensor $\mathbf{G}_i$, and Γ_i is a 1×2 matrix of the Christoffel symbols of the second kind. Bold, capitalized letters indicate matrices, and derivations of these expressions are given in the Appendix and in [4].

Eq. (2) can be expressed in control-affine form as:

$$\dot{q} = \mathcal{F}(q)\Omega = \begin{bmatrix}(\sqrt{\mathbf{G}_1})^{-1}\mathbf{R}_\psi\\ (\sqrt{\mathbf{G}_2})^{-1}\\ \mathbf{T}_1\mathbf{R}_\psi+\mathbf{T}_2\end{bmatrix}\mathbf{H}_{\text{rel}}^{-1}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\Omega, \tag{3}$$

where $q = [u_1\ v_1\ u_2\ v_2\ \psi]^\mathsf{T}$ are the states, $\Omega = [\omega_x\ \omega_y]^\mathsf{T}$ are the controls, $\mathbf{T}_i = \sigma_i\Gamma_i(\sqrt{\mathbf{G}_i})^{-1}$, and $\mathbf{H}_{\text{rel}} = (\widetilde{\mathbf{H}}_1+\mathbf{H}_2)$ represents the relative curvature at the contact.

An example of two spheres rolling with constant relative rotational velocity $\Omega = (\omega_x, 0) = (\frac{4\pi}{3}, 0)$ along their respective "equators" is shown in Figure 3. Figure 3(a) shows a visualization of the spheres, the contact paths on the surface of each object $U_i(t)$, and the coordinate chart for object 2. Figure 3(b) shows the values of the contact coordinates during the rolling motion.

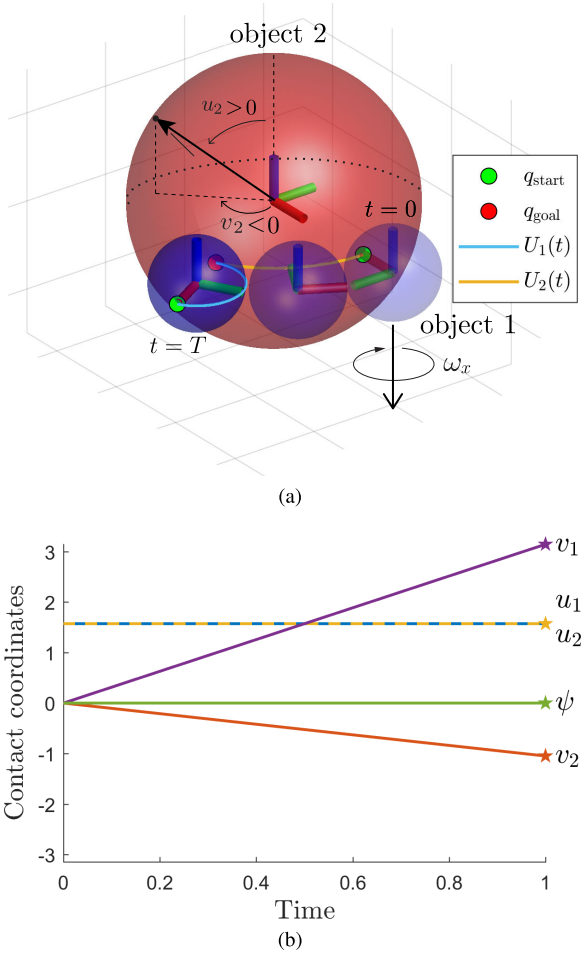


FIGURE 3. Example of object 1, the blue sphere of radius $\rho_1 = 1$, rolling on the equator of object 2, the red sphere of radius $\rho_2 = 3$. The coordinate charts are given by Eq. (1), the initial conditions are $q(0) = (\pi/2, 0, \pi/2, 0, 0)$, and the constant relative rotational velocity is $\Omega = (\omega_x, 0) = (\frac{4\pi}{3}, 0)$. A visualization with the start and goal locations and the contact trajectories $U_i(t)$ is shown in (a). Note that the controls Ω are measured in the object 2 contact frame $\{c_2\}$ with the x_2-axis pointing downwards and the contact normal n_2 pointing out of object 2. A plot of the contact coordinates is shown in (b), with the desired goal states q_{goal} represented by stars. Note that u_1 and u_2 are equal throughout the trajectory and therefore overlap.

IV. PROBLEM STATEMENT

The goal is to find a pure-rolling trajectory and a stabilizing feedback controller from an initial state q_{start} to a goal state q_{goal} for two bodies in pure-rolling contact. We assume that a path between q_{start} and q_{goal} exists within a single pair of coordinate charts (f_1 on object 1 and f_2 on object 2). An "admissible" trajectory is defined as a set of states and controls $\xi(t) = (q(t), \Omega(t))$, from $t = 0$ to the final time $t = T$, that satisfies the first-order pure-rolling kinematics in Eq. (3). A "valid" trajectory is defined as an admissible trajectory that also satisfies $q_{\text{error}}(T) < \eta$, where η is the tolerance on the final state error and $q_{\text{error}}(T) = ||q(T) - q_{\text{goal}}||$, where $||\cdot||$ corresponds to a weighted norm that puts contact parameter errors and spin angle errors in common units. (Throughout the rest of this paper, we use the Euclidean norm.) A stabilizing state-feedback controller about a trajectory is defined as $\Omega_{\text{fbk}}(q, t)$. With these definitions, the problem can be stated as follows:

Given: The surface parameterizations (f_1, f_2), the states $(q_{\text{start}}, q_{\text{goal}})$, and the rolling time T,

find: (1) a valid rolling trajectory $\xi_{\text{sol}}(t)$ for $t \in [0, T]$ that brings the system from $q(0) = q_{\text{start}}$ to $q(T) = q_{\text{goal}}$ and (2) a feedback controller $\Omega_{\text{fbk}}(q, t)$ that stabilizes that trajectory.

In the following section we outline the multi-step process we developed to solve the pure-rolling motion planning and control problem.

V. MOTION PLANNING

A multi-step algorithm is established to solve the motion planning problem presented in Section IV. Given the model parameters, the start state q_{start}, and goal state q_{goal}, the motion planner solves multiple problems of increasing complexity to find a valid trajectory.

The first step is a two-state control (TSC) method that solves a simplified motion planning problem that

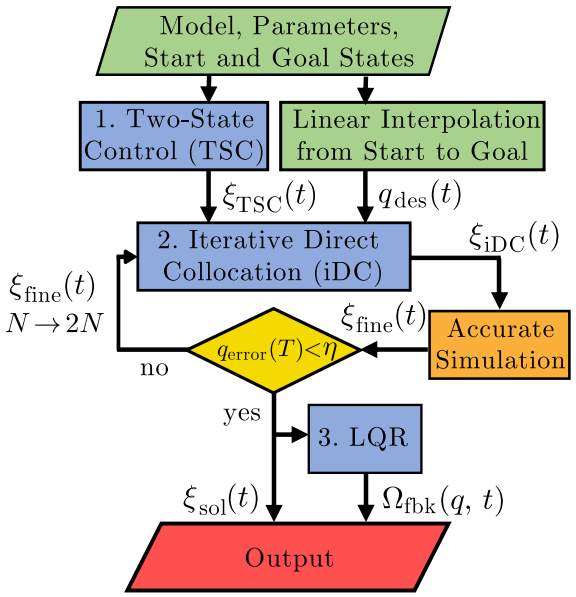


FIGURE 4. Flowchart of the multi-step motion planning algorithm and stabilization method outlined in Sections V and VI, respectively. The inputs are the start/goal states, the model, and the parameters. Two-state control (see Section V-A) is used to generate the initial trajectory guess $\xi_{\text{TSC}}(t) = (q_{\text{TSC}}(t), \Omega_{\text{TSC}}(t))$ for the iterative direct collocation, and $q_{\text{des}}(t)$ is the straight-line desired path for the cost function in Eq. (5). Each output trajectory $\xi_{\text{iDC}}(t) = (q_{\text{iDC}}(t), \Omega_{\text{iDC}}(t))$ is recalculated using a higher-order integration method and the goal error tolerance is checked ($q_{\text{error}}(T) < \eta$). If the trajectory is not valid, it is used as the initial trajectory guess for the next iteration of the direct collocation method with twice as many segments ($N \to 2N$). This is repeated until a valid trajectory is found, the maximum number of iDC iterations is reached, or the optimization converges to an invalid point. The linear quadratic regulator (LQR) step outputs a feedback control law $\Omega_{\text{fbk}}(q, t)$ that stabilizes the solution trajectory $\xi_{\text{sol}}(t)$.

directly controls two of the five configuration variables. The output of the TSC planner is the trajectory $\xi_{\text{TSC}}(t) = (q_{\text{TSC}}(t), \Omega_{\text{TSC}}(t))$ consisting of the configurations and controls as a function of time. The second step is the iterative direct collocation (iDC) method that takes ξ_{TSC} as the initial trajectory guess and runs an optimization to find a coarse rolling trajectory (large time steps, simple integration method) for the full five-dimensional configuration. The control output from the optimization $\Omega_{\text{iDC}}(t)$ is then used to numerically integrate the first-order kinematics using a higher-order integrator (MATLAB's `ode45`), and the goal error $q_{\text{error}}(T)$ is calculated. If the trajectory is valid ($q_{\text{error}}(T) < \eta$), then the optimization is stopped and the trajectory is returned. If the terminal condition is not satisfied, the previous trajectory serves as the initial trajectory guess for a finer direct collocation optimization with twice as many collocation segments ($N \to 2N$). This is repeated until a valid trajectory is found, the maximum number of iDC iterations is reached, or the optimization converges to an invalid point. A flowchart of the algorithm is shown in Figure 4, and the details of the individual methods are given below.

A. TWO-STATE CONTROL (TSC)

The purpose of the two-state control method is to obtain the control input that moves two states along the shortest path to the goal state. The choice of which coordinates to control will bias the optimization solutions toward different paths in the state-space. In this paper we choose to control the coordinates $U_2(t) = [u_2(t)\, v_2(t)]^{\mathsf{T}}$. An analysis of the planner performance for different initial trajectory guess methods is included in Section VIII-B.

We first linearly interpolate the object 2 coordinates from $U_2(0)$ to $U_2(T)$. With the trajectory $U_2(t)$ known, we calculate the control input $\Omega(t)$ from the second expression in Eq. (2), which results in:

$$\begin{bmatrix}\omega_x(t)\\ \omega_y(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}(\widetilde{\mathbf{H}}_1(t)+\mathbf{H}_2(t))\sqrt{\mathbf{G}_2}(t)\dot{U}_2(t). \tag{4}$$

We then use q_{start}, $\Omega(t)$, and the other two kinematics expressions of Eq. (2) to calculate the trajectories of the remaining three states $u_1(t)$, $v_1(t)$, and $\psi(t)$. This method generates a trajectory $\xi_{\text{TSC}}(t) = (q_{\text{TSC}}(t), \Omega_{\text{TSC}}(t))$ that tracks the straight-line path between the start and goal contact locations on object 2. Examples are shown in Figures 3(b) and 8(a).

B. ITERATIVE DIRECT COLLOCATION (iDC)

The initial trajectory $\xi_{\text{TSC}}(t)$ is admissible, but only two of the five states reach the desired goal states. We therefore perturb the input trajectory from the TSC method so that $q(T) = q_{\text{goal}}$, and to do this we use direct collocation. We first describe the details of the direct collocation method, and then outline our iterative version.

Direct collocation is a method for trajectory optimization that optimizes an objective function $J(\xi(t)) = J(q(t), \Omega(t))$ using polynomial spline approximations of the continuous states and controls. We chose to use trapezoidal collocation where the control trajectory $\Omega(t)$ is represented by piecewise-linear splines, the state trajectory $q(t)$ is represented by a quadratic spline, and the trapezoidal rule is used for integration. Higher-order representations such as Hermite-Simpson collocation can also be used but with increased computational cost [23]. We define the objective function $J(q(t), \Omega(t))$ as the sum of the terminal cost and the running cost and omit the dependence on t for clarity:

$$\begin{aligned} J(q, \Omega) &= m(q(T)) + \int_0^T l(q, \Omega)dt,\\ m(q(T)) &= \frac{1}{2}(q(T) - q_{\text{goal}})^{\mathsf{T}}\mathbf{P}_1(q(T) - q_{\text{goal}}),\\ l(q, \Omega) &= \frac{1}{2}(q - q_{\text{des}})^{\mathsf{T}}\mathbf{Q}(q - q_{\text{des}}) + \frac{1}{2}\Omega^{\mathsf{T}}\mathbf{R}\Omega, \end{aligned} \tag{5}$$

where $\mathbf{P}_1$, $\mathbf{Q}$, and $\mathbf{R}$, penalize goal-state error, desired trajectory deviation, and control cost respectively, and $q_{\text{des}}(t)$ is a nominal trajectory. The path $q_{\text{des}}(t)$ is chosen as the linear interpolation from q_{start} to q_{goal}, which penalizes motions that do not move q towards the goal. Note that $q_{\text{des}}(t)$ is not admissible in general.

The collocation method divides the trajectory $\xi(t)$ into N segments, and the $N+1$ nodes at the ends of each segment are called collocation points. Each collocation point is expressed as $\xi_k(t) = (q(t_k), \Omega(t_k))$ for $k \in [0, \ldots, N]$. For systems with

m state variables and n control variables there are a total of $(N+1)(m+n)$ collocation points. The dynamics between each pair of sequential collocation points are enforced by the following condition:

$$q_{k+1} - q_k = \frac{1}{2}\Delta t_k(\mathcal{F}(q_{k+1})\Omega_{k+1} + \mathcal{F}(q_k)\Omega_k), \quad k \in [0, \ldots, N-1], \quad (6)$$

where $\Delta t_k = (t_{k+1} - t_k)$ indicates the interval duration and $\mathcal{F}(q)\Omega$ is the first-order kinematics function from Eq. (3). Equation (6) is unique to the choice of trapezoidal collocation, and other integration methods require a different constraint [23].

The optimal control problem can be represented as the following nonlinear programming problem:

$$\begin{aligned} &\underset{q(t_k),\ \Omega(t_k)}{\arg\min}\ m(q(T)) + \sum_{i=0}^{N} l(q(t_k), \Omega(t_k))\Delta t_k \\ &\text{such that } h(q(t_0) : q(t_N); \Omega(t_0) : \Omega(t_{N-1})) = 0, \\ &\qquad g(q(t_0) : q(t_N); \Omega(t_0) : \Omega(t_{N-1})) \leq 0, \end{aligned} \quad (7)$$

where $h(\cdot)$ gives the equality constraints $q(0) = q_{\text{start}}$ and $q(T) = q_{\text{goal}}$ and enforces the first-order kinematics in Eq. (6). The expression $g(\cdot)$ gives the inequality constraints which constrain the controls ($\Omega_{\min} \leq \Omega \leq \Omega_{\max}$) and enforces any constraints on the configurations (e.g., due to singularities in the coordinate chart). Equation (7) is a finite-dimensional nonlinear optimization problem, and a solution $\xi_{\text{iDC}}(t)$ can be found using nonlinear optimizers such as SNOPT, IPOPT, or MATLAB's `fmincon`.

The integration error can be determined by comparing the trajectory $q_{\text{iDC}}(t)$ from the direct collocation method with the trajectory $q_{\text{fine}}(t)$, where $q_{\text{fine}}(t)$ is obtained by integrating the initial state over the interval $t = [0, T]$ using Eq. (3), the piecewise-linear output controls $\Omega_{\text{iDC}}(t)$, and a higher-order integrator with small time steps ($dt \leq 0.001$). With fewer segments N, the integration error is larger, but there are fewer constraints for the nonlinear solver. This means that the optimizer is more likely to find a solution, and with less computational cost. The choice of N is therefore a trade-off between computational cost/optimizer convergence and integration error. We implemented the iterative direct collocation (iDC) method to address this.

We first run the nonlinear optimization method using MATLAB's `fmincon` for a small number of segments ($N = 25$) to find a trajectory $\xi_{\text{iDC}}(t)$. The recalculated path $q_{\text{fine}}(t)$ is found using smaller integration timesteps and a higher-order integrator (`ode45`), and the planner is terminated if the goal-state tolerance of the fine trajectory is satisfied ($q_{\text{error}}(T) < \eta$). If the goal-state error is too large, the previous output trajectory serves as the initial trajectory guess for the next iteration with twice as many segments ($N \to 2N$). This is repeated until a valid trajectory $\xi_{\text{sol}}(t)$ is found, the maximum number of iDC iterations is reached, or the optimization converges to an invalid point. Small values of N result in good solutions but may require many iDC iterations to converge to a solution, and large values will increase N too quickly resulting in slower computation time and convergence to invalid points. We chose to double the segments between each iteration so the exact solution from the previous iteration could be used, but N could be increased by a fixed value ΔN between each iteration to add an additional tuning parameter for the planning method.

VI. FEEDBACK CONTROL OF ROLLING SURFACES

To stabilize the rolling trajectory to state disturbances, we construct a linear time-varying feedback controller based on a linearization of the rolling kinematics about the planned trajectory $q_{\text{sol}}(t)$. For this approach to be appropriate, the system should be linearly controllable about the planned trajectory, which may not always be the case, even if the system is nonlinearly controllable.

A. LINEARIZATION ABOUT A TRAJECTORY

Given a nominal trajectory $\xi_{\text{nom}}(t) = (q_{\text{nom}}(t), \Omega_{\text{nom}}(t))$, we define perturbations about the trajectory as:

$$\widetilde{q}(t) = q(t) - q_{\text{nom}}(t), \quad (8)$$

$$\widetilde{\Omega}(t) = \Omega(t) - \Omega_{\text{nom}}(t). \quad (9)$$

The perturbed version of the dynamics in Eq. (3) can be written using a first-order Taylor expansion (and omitting the dependence on t) as:

$$\begin{aligned} \dot{q}_{\text{nom}} + \dot{\widetilde{q}} = {} & \mathcal{F}(q_{\text{nom}})\Omega_{\text{nom}} \\ & + \left[\frac{\partial(\mathcal{F}(q)\Omega)}{\partial q}\right]_{\text{nom}} \widetilde{q} \\ & + \left[\frac{\partial(\mathcal{F}(q)\Omega)}{\partial \Omega}\right]_{\text{nom}} \widetilde{\Omega} + \text{h.o.t.}, \end{aligned} \quad (10)$$

where $[\cdot]_{\text{nom}}$ means the enclosed expressions are evaluated along the nominal trajectory, and h.o.t. represents higher-order terms. Because $\dot{q}_{\text{nom}} = \mathcal{F}(q_{\text{nom}})\Omega_{\text{nom}}$ and h.o.t. are negligible for nearby trajectories, Eq. (10) simplifies to:

$$\dot{\widetilde{q}} = \underbrace{\left[\frac{\partial(\mathcal{F}(q)\Omega)}{\partial q}\right]_{\text{nom}}}_{\widetilde{\mathbf{A}}(t)_{m\times m}} \widetilde{q} + \underbrace{[\mathcal{F}(q)]_{\text{nom}}}_{\widetilde{\mathbf{B}}(t)_{m\times n}} \widetilde{\Omega}, \quad (11)$$

where m is the number of state variables and n is the number of controls. We analyze the controllability properties of the linear time-varying (LTV) system $(\widetilde{\mathbf{A}}(t), \widetilde{\mathbf{B}}(t))$ to determine whether the nominal trajectory error $(\widetilde{q}(t), \widetilde{\Omega}(t))$ can be stabilized to zero by a simple linear controller.

B. CONTROLLABILITY OF LINEAR TIME-VARYING (LTV) SYSTEMS

The controllability of an LTV system along a nominal trajectory can be checked using the controllability gramian (e.g., Ch. 11.6 of [24]),

$$\mathbf{W}_{\mathbf{c}}(t_1, t_0) = \int_{t_0}^{t_1} \Phi(t_1, \tau)\widetilde{\mathbf{B}}(\tau)\widetilde{\mathbf{B}}(\tau)^{\mathsf{T}}\Phi(t_1, \tau)^{\mathsf{T}} d\tau, \quad (12)$$

where the state-transition matrix[1] $\Phi(t_1, \tau)$ and $\widetilde{\mathbf{B}}$ come from Eq. (11) and correspond to the linearization about the nominal trajectory. If the square matrix $\mathbf{W_c}(t_1, t_0)$ is non-singular on the interval $t \in [t_0, t_1]$, then the linearized trajectory can be stabilized by an appropriate controller. For a system with m state variables, any system with $\text{rank}(\mathbf{W_c}(t_1, t_0)) < m$ is rank deficient, and therefore uncontrollable.

C. CONTROLLABILITY OF ROLLING TRAJECTORIES

The state q and controls Ω for the rolling system are five-dimensional and two-dimensional, respectively, so $m = 5$ and $n = 2$. Therefore $\widetilde{\mathbf{A}}(t)$ is a 5×5 matrix, $\widetilde{\mathbf{B}}(t)$ is a 5×2 matrix, and full rank of the controllability gramian is 5.

While the linearized rolling dynamics of "generic" objects are controllable about "generic" rolling trajectories, there are at least three degenerate situations where the controllability gramian fails to achieve full rank, as outlined below.

1) DEGENERATE GEOMETRY

An example of this case is two spheres of equal radius. As shown by Li and Canny [14], the relative configuration of two spheres of equal radius is uncontrollable by pure rolling, and therefore the linearization of the kinematics about any rolling trajectory is also uncontrollable. This conclusion is independent of the initial contact configuration of the bodies.

2) DEGENERATE INITIAL CONFIGURATION

In other cases, the relative configuration of two objects may be controllable by rolling from most configurations but not from others. An example is shown in Figure 5, where two identical ellipsoids, each with two equal principal semi-axes and one longer principal semi-axis, make initial contact such that the body geometries are symmetric about the contact tangent plane. Regardless of the rolling motion chosen from this initial configuration, the system will be confined to a lower-dimensional subset of its five-dimensional configuration space. More details can be found in [18] and Section 24.4 of [19].

3) DEGENERATE TRAJECTORY

Finally, even if the two bodies are not geometrically degenerate and their initial configuration is not degenerate, a rolling trajectory may be chosen such that the rank of the controllability gramian of the linearized dynamics about the rolling trajectory never exceeds four. A trivial example is a stationary trajectory. For stationary trajectories ($\dot{q}(t) = 0 \; \forall \; t \in [0, T]$), the matrix $\widetilde{\mathbf{A}}(t)$ is zero and the matrix $\widetilde{\mathbf{B}}(t) = \widetilde{\mathbf{B}}(t_0)$ is constant. $\widetilde{\mathbf{B}}(t_0)$ is always full rank because rolling is allowed in two directions, so the controllability gramian is rank two for all stationary trajectories. The linearized rolling system is not controllable about stationary trajectories.

[1]The state-transition matrix $\Phi(t_1, \tau)$ can be calculated for the LTV system using methods in Section 9.5-9.6 of [24] such as the the Peano-Baker Series or the fundamental solution matrix.

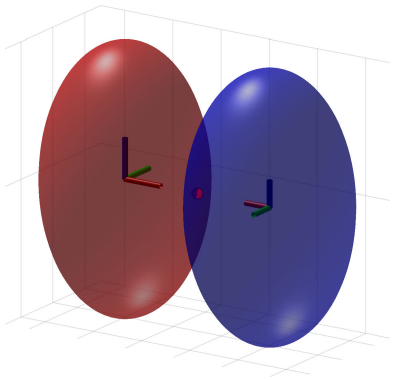


FIGURE 5. Example of uncontrollable initial condition for two identical ellipsoids.

Another example is shown in the sphere-on-sphere trajectory of Figure 3. This trajectory illustrates constant Ω, u_1, u_2, and ψ. The linearized dynamics are governed by the linear time invariant (LTI) matrices $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$:

$$\widetilde{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 & 0 & \pi \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\pi & 0 & \frac{\pi}{3} & 0 & 0 \end{bmatrix}, \quad \widetilde{\mathbf{B}} = \begin{bmatrix} 0 & \frac{3}{4} \\ \frac{3}{4} & 0 \\ 0 & \frac{1}{4} \\ -\frac{1}{4} & 0 \\ 0 & 0 \end{bmatrix}. \tag{13}$$

The Kalman controllability matrix $[\widetilde{\mathbf{B}} \; \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} \; \widetilde{\mathbf{A}}^2\widetilde{\mathbf{B}} \; \widetilde{\mathbf{A}}^3\widetilde{\mathbf{B}} \; \widetilde{\mathbf{A}}^4\widetilde{\mathbf{B}}]$ is only rank four.

Even a trajectory about which the linearized system is controllable can be problematic to stabilize if the controllability gramian is ill-conditioned; large controls may be required to recover from small errors in certain state directions. Various metrics on the controllability gramian can be used to quantify controllability, such as the minimum eigenvalue ($\lambda_{\min}(\mathbf{W_c}(t_1, t_0))$), the trace of the inverse, ($\text{tr}(\mathbf{W_c}^{-1}(t_1, t_0))$), and the determinant ($\det(\mathbf{W_c}(t_1, t_0))$) [25], [26]. One of these measures could be included in the objective function in Eq. (5) to bias the trajectory optimization away from nearly-degenerate trajectories.

D. STABILIZATION OF ROLLING TRAJECTORIES

We use the linear quadratic regulator (LQR) to stabilize the linearized dynamics in Eq. (11). LQR computes a time-varying gain matrix $\mathbf{K}(t)$ that optimally reduces the total cost for small perturbations about the nominal trajectory. LQR requires a cost function, and we use the one given in Eq. (5). We solve the matrix Riccati equation to find the time varying feedback control matrix $\mathbf{K}(t)$ (see Section 2.3 of [27]).

$$\begin{aligned} -\dot{\mathbf{P}}(t) &= \mathbf{P}(t)\widetilde{\mathbf{A}}(t) + \widetilde{\mathbf{A}}(t)^{\mathsf{T}}\mathbf{P}(t) \\ &\quad - \mathbf{P}(t)\widetilde{\mathbf{B}}(t)\,\mathbf{R}_{\text{LQR}}^{-1}\widetilde{\mathbf{B}}(t)^{\mathsf{T}}\mathbf{P}(t) + \mathbf{Q}_{\text{LQR}}, \\ \mathbf{P}(T) &= \mathbf{P}_{1,\text{LQR}} \\ \mathbf{K}(t) &= \mathbf{R}_{\text{LQR}}^{-1}\widetilde{\mathbf{B}}(t)^{\mathsf{T}}\mathbf{P}(t). \end{aligned} \tag{14}$$

TABLE 1. Parameters used in Section VII for the iterative direct collocation algorithm.

Description	Value
Trajectory execution time T	1 s
Initial # of segments N	25 ($\Delta t = 0.04$ s)
Goal error tolerance η	0.01
Max iDC iterations	4
Max `fmincon` func evals/iteration	25,000
Control limits	$\|\|\omega_x\|\| \leq 30, \|\|\omega_y\|\| \leq 30$
Constraint integration method	trapezoidal
$\mathbf{P}_1$ (terminal state weight)	$\text{diag}(100, 100, 100, 100, 100)$
$\mathbf{Q}$ (tracking weight)	$\text{diag}(1, 1, 1, 1, 1)$
$\mathbf{R}$ (control weight)	$\text{diag}(0.1, 0.1)$

The matrix $\mathbf{K}(t)$ is then used in the feedback control law

$$\Omega_{\text{fbk}}(q, t) = \Omega_{\text{nom}}(t) - \mathbf{K}(t)(q(t) + q_{\text{nom}}(t)) \qquad (15)$$

to stabilize the nominal trajectory. The performance of this feedback controller depends on the controllability properties of the linearized dynamics.

VII. SIMULATION EXAMPLES

We now demonstrate the motion planning and feedback control method for a sphere rolling on a sphere and an ellipsoid rolling on an ellipsoid. The sphere example demonstrates the ability to generate shorter paths for nontrivial trajectories when compared to the recent geometric planner for a sphere rolling on a sphere in [17]. The ellipsoid example demonstrates motion planning and feedback control for shapes with spatially-varying curvature.

We use the SQP algorithm of MATLAB's `fmincon` as our nonlinear optimization solver, and a list of the parameters used is included in Table 1. The code was run on an i7-4700MQ CPU @ 2.40 GHz with 16 GB of RAM. For each example we present the number of iDC iterations, computation time, final state error ($q_{\text{error}}(T) = ||q(T) - q_{\text{goal}}||$), and trajectory cost from the cost function in Eq. (5).

A. SPHERE ON SPHERE

We compare the results of our planner to a geometric trajectory planner for a sphere rolling on a sphere presented in Section 4.2 of [17]. Geometric trajectory planners use properties of the surface geometries to derive analytic expressions for the motion plans between start and goal states. While they are guaranteed to find exact solutions, such approaches are limited to specific geometries and solution paths are often unnecessarily long. Convergence is not guaranteed for our method, but in practice it works well and finds shorter paths.

Eq. (1) gives the parametric model of the spheres, and their radii are $\rho_1 = 2$ and $\rho_2 = 10$. The start and goal states are chosen as $q_{\text{start}} = (\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{2}, 0, 0)$ and $q_{\text{goal}} = (2.19, \frac{-3\pi}{4}, 0.96, \frac{\pi}{4}, 0)$ to match the sphere-on-sphere example in Section 4.2 of [17].

A solution was found after four iterations of the iDC method, with a total computation time of 47 seconds, a final state error of $q_{\text{error}}(T) = 0.002$, and a trajectory cost of 5.3. A visualization of the resulting motion plan is shown in Figure 6. The length of the path $U_2(t)$ (the path on the larger sphere) is approximately three times shorter than the solution presented in [17]. An animation of the initial guess and final trajectory is in the attached supplemental media.

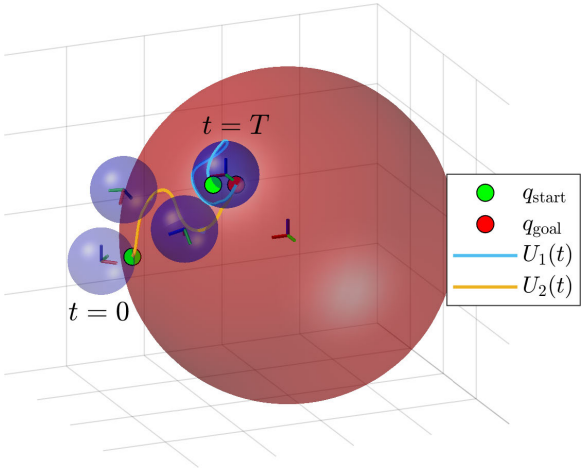


FIGURE 6. The sphere-on-sphere solution trajectory from Section VII-A. The smaller blue object 1 is rolling from bottom left to top right on the larger red object 2, and is shown at times $t = (0, \frac{T}{3}, \frac{2T}{3}, T)$. The contact path $U_1(t)$ is shown on the object at $t = T$, and the contact path $U_2(t)$ is shown on object 2. The initial and goal states were chosen to compare to the results from Rehan et al. [17], and our planned path $U_2(t)$ on object 2 is approximately three times shorter than the solution from the geometric planning method in that paper.

B. ELLIPSOID ON ELLIPSOID

We now demonstrate the planner on the more complex example of an ellipsoid rolling on an ellipsoid. Eq. (16) gives the parametric model of the ellipsoids

$$f_i : U_i \to \mathbb{R}^3 : (u_i, v_i) \mapsto (\rho_{ia} \sin(u_i) \cos(v_i),\ \rho_{ib} \sin(u_i) \sin(v_i),\ \rho_{ic} \cos(u_i)), \qquad (16)$$

where u_i satisfies $0 < u_i < \pi$, v_i satisfies $-\pi < v_i < \pi$, and the principal semi-axes are chosen as $(\rho_{1a}, \rho_{1b}, \rho_{1c}) = (1, 1, 1.5)$ and $(\rho_{2a}, \rho_{2b}, \rho_{2c}) = (3, 3, 5)$. The coordinate systems are orthogonal because $\rho_{1a} = \rho_{1b}$ and $\rho_{2a} = \rho_{2b}$. The start and goal states are chosen as $q_{\text{start}} = (\pi/2, 0, \pi/2, 0, 0)$ and $q_{\text{goal}} = (\pi/2, 0, \pi/4, -\pi/2, -\pi/4)$. A solution was found after four iterations of the iDC method, with a total computation time of 61 seconds, a final state error of $q_{\text{error}}(T) = 0.003$, and a trajectory cost of 12.8. The resulting path is shown in Figure 7. The contact coordinates and controls for the initial trajectory guesses, the results from the first iDC iteration, and the final trajectory are shown in Figure 8. An animation of the initial guess and final trajectory is in the attached supplemental media.

C. FEEDBACK CONTROL

This section demonstrates the performance of the LQR controller in stabilizing trajectories with initial state perturbations. The weighting parameters for the LQR feedback controller are given in Table 2. The weights $\mathbf{Q}_{\text{LQR}}$ and $\mathbf{P}_{1,\text{LQR}}$

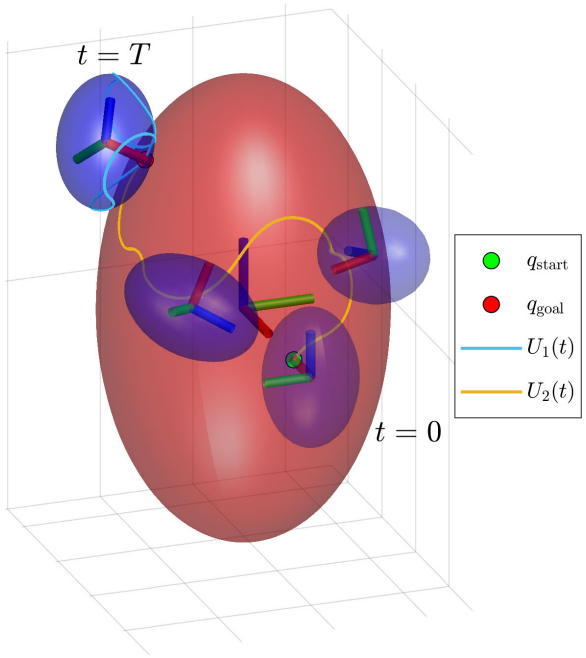


FIGURE 7. Ellipsoid-on-ellipsoid visualization for the motion plan in Section VII-B and Figure 8. The smaller blue object 1 rolls from bottom right to top left on the larger red object 2, and is shown at times $t = (0, \frac{T}{3}, \frac{2T}{3}, T)$. The contact path $U_1(t)$ is shown on object 1 at $t = T$, and the contact path $U_2(t)$ is shown on object 2. The $U_1(t)$ and $U_2(t)$ trajectories are shown in Figure 8 (c).

TABLE 2. Parameters used in Section VII-C for the LQR feedback control of rolling trajectories.

Description	Value
$\mathbf{P}_{\text{LQR}}$ (terminal state weight)	$\text{diag}(1e5, 1e5, 1e5, 1e5, 1e5)$
$\mathbf{Q}_{\text{LQR}}$ (tracking weight)	$\text{diag}(100, 100, 100, 100, 100)$
$\mathbf{R}_{\text{LQR}}$ (control weight)	$\text{diag}(0.1, 0.1)$

were both increased from the direct collocation optimization weights to improve tracking performance and decrease the final state error.

1) SPHERE-ON-SPHERE

We first demonstrate feedback control on the simple sphere-on-sphere equator trajectory in Figure 3 with an initial state perturbation of $\epsilon(q(0)) = (0.1, 0.05, -0.05, -0.1, 0)$, where $\epsilon(q(t))$ is the difference between the current and nominal reference trajectory and is defined as $\epsilon(q(t)) = q(t) - q_{\text{nom}}(t)$. We set the nominal trajectory to $\xi_{\text{nom}}(t) = \xi_{\text{sol}}(t)$, and as mentioned in Section VI-C, the linearized dynamics are not controllable about this degenerate trajectory.

Figure 9(a) shows the individual and total coordinate error over time. The norm of the initial state error is $||\epsilon(q(0))|| = 0.16$ and the norm of the final state error is $||\epsilon(q(T))|| = 0.08$. LQR is ineffective at eliminating the error because the linearized dynamics are uncontrollable about the nominal trajectory. Animations of the open- and closed-loop performance for the perturbed sphere-on-sphere trajectories in Figures 3 and 6 can be seen in the supplemental media.

TABLE 3. Testing the planning method on 100 random goal states for sphere-on-sphere and ellipsoid-on-ellipsoid rolling. Results are given as "mean (standard deviation)".

Geometry	Planning time (s)	$q_{\text{error}}(T)$	Cost	% Success
Spheres	16 (9)	0.045 (0.027)	13 (6)	99
Ellipsoids	17 (8)	0.04 (0.028)	12 (5.7)	99

TABLE 4. Comparing the effect of the initial trajectory guess method for 100 random goal states for ellipsoid-on-ellipsoid rolling. The different initial guess methods were two-state control on object 1, two-state control on object 2, linear interpolation ($q(t) = q_{\text{des}}$, $\Omega(t) = 0$), and stationary ($\dot{q}(t) = 0$, $\Omega(t) = 0$). Results are given as "mean (standard deviation)".

Geometry	Planning time (s)	$q_{\text{error}}(T)$	Cost	% Success
TSC Object 1	17 (13)	0.059 (0.18)	13.6 (17)	99
TSC Object 2	17 (8)	0.04 (0.028)	12 (5.7)	99
Linear Interp.	31 (54)	0.24 (0.65)	33 (96)	88
Stationary	26 (29)	0.21 (0.59)	30 (85)	91

The controllability gramian of the linearized dynamics about the sphere-on-sphere trajectory in Figure 6 is full rank, and therefore LQR eliminates the state error.

2) ELLIPSOID-ON-ELLIPSOID

Figure 9(b) shows the individual and total coordinate error over time for the nominal trajectory of Figure 7 and an initial perturbation $\epsilon(q(0)) = (0.1, 0.05, -0.05, -0.1, 0)$. The norm of the initial state error is $||\epsilon(q(0))|| = 0.16$ and the norm of the final state error is $||\epsilon(q(T))|| = 0.0004$. We see that the feedback controller effectively recovers from the initial error. An animation of the open and closed-loop performance for the perturbed ellipsoid-on-ellipsoid trajectory can be seen in the supplemental media.

VIII. DISCUSSION

A. ROBUSTNESS

To test the robustness of the proposed planner, we generated 100 random trajectory planning tasks for the sphere-on-sphere and ellipsoid-on-ellipsoid. The initial state was fixed at $q_{\text{start}} = (\pi/2, 0, \pi/2, 0, 0)$ and goal states were chosen in the range $(0, -\pi, 0, -\pi, -\pi) < q_{\text{goal}} < \pi$. The results are shown in Table 3. The planner used the parameters in Table 1, other than maximum `fmincon` function evaluations set to 15,000 and η set to 0.1 to decrease computation time.

These results demonstrate the ability of the planner to reliably find solutions for random goal states. While the method is not guaranteed to find a solution, it works well in practice.

B. EFFECT OF THE INITIAL TRAJECTORY GUESS

The initial trajectory guess to the iDC optimization is "two-state control," which drives the contact coordinates of object 2 directly to their desired final value. This trajectory is admissible, i.e., it satisfies the rolling conditions. Other initial trajectory guesses could be used, like two-state control for the contact coordinates of object 1, which is admissible; linear interpolation ($q(t) = q_{\text{des}}(t)$,

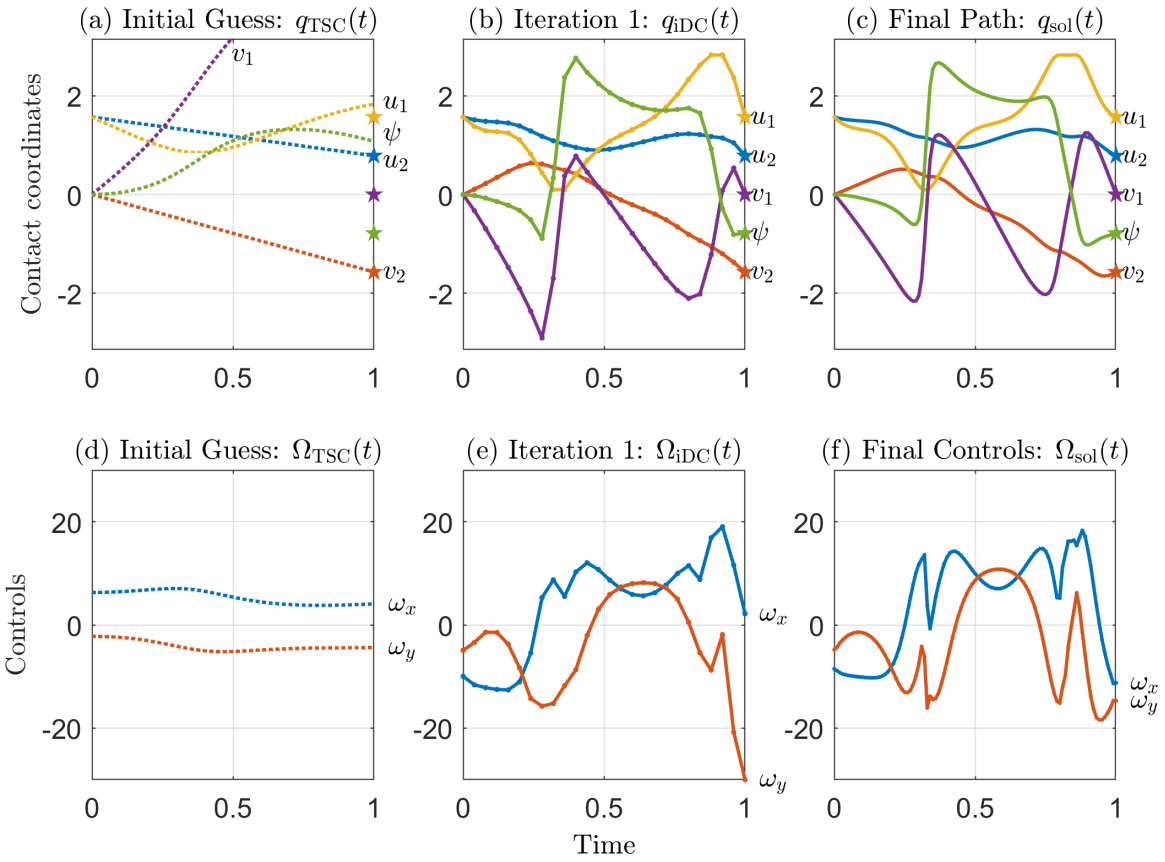


FIGURE 8. Contact coordinate plots and control plots for the ellipsoid-on-ellipsoid rolling plan in Figure 7. Column one shows the initial trajectory guess from the two-state control method, column two shows the output trajectory from the first iteration of the iterative direct collocation method (which fails to satisfy the tolerance criterion after accurate simulation), and column three shows the solution trajectory. The stars in (a)-(c) show the desired goal states q_{goal}.

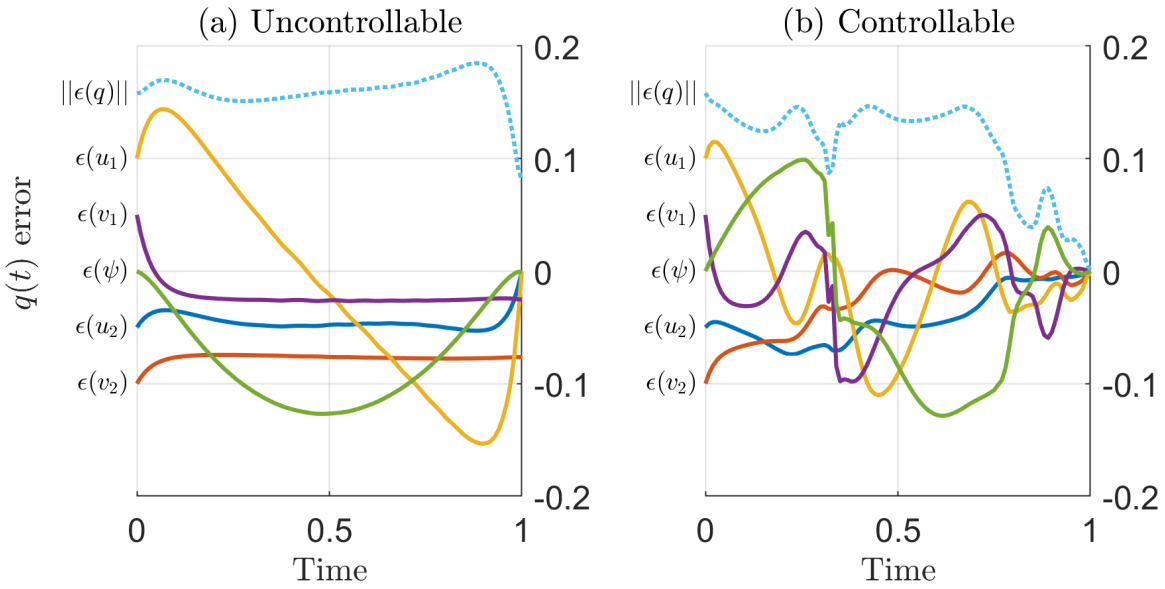


FIGURE 9. Error recovery of sphere-on-sphere (a) and ellipsoid-on-ellipsoid (b) under feedback control with an initial state perturbation of $\epsilon(q(0)) = (0.1, 0.05, -0.05, -0.1, 0)$. The function $\epsilon(\cdot)$ calculates the difference between input coordinate(s) and the reference coordinate(s) in $q_{\mathrm{nom}}(t)$, and $||\epsilon(q)||$ is the norm of the total coordinate error. The sphere trajectory in (a) is the equator example given in Figure 3, where the controllability gramian is not full rank, and therefore LQR cannot eliminate the state error. The ellipsoid trajectory in (b) is for the ellipsoid example in Figure 7. The controllability gramian for this trajectory is full rank and therefore the controller is able to reduce the error to zero.

$\Omega(t) = 0$), which is inadmissible; and the stationary trajectory ($\dot{q}(t) = 0$, $\Omega(t) = 0$), which is admissible. While all types of initial trajectory guesses led to solutions in most cases, the two-state control on object 2 performed the best with the shortest planning times, lowest costs, smallest errors, and highest success rate for random planning problems for the ellipsoid-on-ellipsoid (Table 4). The size difference between the objects could explain the difference between the performance of TSC on objects 1 and 2. Because object 1 is smaller than object 2, TSC1 generates short contact paths on object 2, so the initial guess for TSC1 more closely resembles the stationary initial guess.

IX. CONCLUSION AND FUTURE WORK

This paper presents a motion planner and feedback controller for pure-rolling motions between general smooth rigid bodies. The methods were demonstrated for a sphere rolling on a sphere and an ellipsoid rolling on an ellipsoid. Future work includes extending motion planning and control to second-order kinematic and dynamic rolling. We are also interested in methods that allow planning for more general object parameterizations (non-orthogonal), or smooth approximations of general surfaces represented by meshes.

APPENDIX
LOCAL GEOMETRY OF SMOOTH BODIES

Below are some expressions for the geometry of a surface that are used to define the first-order kinematics in Section III. References and derivations of these expressions can be found in [4].

We represent the surface of each body in contact as a mapping $f_i : (u_i, v_i) \mapsto (x_i, y_i, z_i)$ for objects $i \in [1, 2]$ (see Section III). It is assumed that f_i is continuous up to the second derivative (class C^2) so that the local contact geometries (contact frames and curvature associated with the first and second derivatives of f_i, respectively) are uniquely defined. The natural bases at a point on a body are given as $x_i = \partial f_i/\partial u_i$ and $y_i = \partial f_i/\partial v_i$. We also assume that coordinate charts are orthogonal ($x_i \cdot y_i = 0$), and note that x_i and y_i are not necessarily unit vectors. The normal is given as $n_i = (x_i \times y_i)/||x_i \times y_i||$.

The normalized Gauss frame at a point U_i on object i is defined as the coordinate frame $\{c_i\}$ with origin at $f_i(U_i)$ and coordinate axes given by

$$\mathbf{R}_{o_i c_i} = \left[\frac{x_i}{||x_i||}, \frac{y_i}{||y_i||}, n_i \right], \tag{17}$$

where $\mathbf{R}_{o_i c_i}$ expresses the Gauss frame in the object i frame $\{o_i\}$. The metric tensor $\mathbf{G}_i$ is a 2×2 positive-definite matrix defined as

$$\mathbf{G}_i = \begin{bmatrix} x_i \cdot x_i & x_i \cdot y_i \\ y_i \cdot x_i & y_i \cdot y_i \end{bmatrix}. \tag{18}$$

The coefficients $g_{jk,i}$ reference the indices of matrix $\mathbf{G}_i$, and $\mathbf{G}_i$ is diagonal ($g_{12,i} = g_{21,i} = 0$) when the coordinate chart f_i is orthogonal. The 2×2 matrix L_i is the second fundamental form given by the expression

$$L_i = \begin{bmatrix} \dfrac{\partial^2 f_i}{\partial u_i^2} \cdot n_i & \dfrac{\partial^2 f_i}{\partial u_i \partial v_i} \cdot n_i \\ \dfrac{\partial^2 f_i}{\partial v_i \partial u_i} \cdot n_i & \dfrac{\partial^2 f_i}{\partial v_i^2} \cdot n_i \end{bmatrix}. \tag{19}$$

$\mathbf{H}_i$ combines the metric tensor $\mathbf{G}_i$ with the second fundamental form $\mathbf{L}_i$ and is given by,

$$\mathbf{H}_i = (\sqrt{\mathbf{G}_i})^{-1} \mathbf{L}_i (\sqrt{\mathbf{G}_i})^{-1}. \tag{20}$$

The 1×2 array Γ_i is given by the expression

$$\Gamma_i = \begin{bmatrix} \Gamma^2_{11,i} & \Gamma^2_{12,i} \end{bmatrix}, \tag{21}$$

where $\Gamma^2_{11,i}$ and $\Gamma^2_{12,i}$ are christoffel symbols of the second kind given by

$$\begin{aligned} \Gamma^2_{11,i} &= \left(\frac{\partial x_i}{\partial u_i} \cdot x_i \right) g_i^{12} + \left(\frac{\partial x_i}{\partial u_i} \cdot y_i \right) g_i^{22}, \\ \Gamma^2_{12,i} &= \left(\frac{\partial x_i}{\partial v_i} \cdot x_i \right) g_i^{12} + \left(\frac{\partial x_i}{\partial v_i} \cdot y_i \right) g_i^{22}, \end{aligned} \tag{22}$$

where g_i^{jk} are entries (j, k) of the metric tensor inverse $(\mathbf{G}_i)^{-1}$.

ACKNOWLEDGMENT

The authors would like to thank P. Umbanhowar for helpful discussions while developing this work.

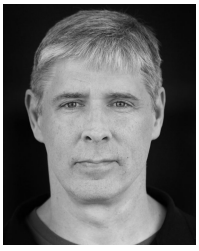
REFERENCES

[1] J. Z. Woodruff and K. M. Lynch, "Second-order contact kinematics between three-dimensional rigid bodies," *J. Appl. Mech.*, vol. 86, no. 8, pp. 1–3, May 2019.
[2] C. Cai and B. Roth, "On the spatial motion of a rigid body with point contact," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 4, Mar. 1987, pp. 686–695.
[3] D. J. Montana, "The kinematics of contact and grasp," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 17–32, Jun. 1988.
[4] N. Sarkar, V. Kumar, and X. Yun, "Velocity and acceleration analysis of contact between three-dimensional rigid bodies," *J. Appl. Mech.*, vol. 63, no. 4, pp. 974–984, Dec. 1996.
[5] N. Sarkar, X. Yun, and V. Kumar, "Dynamic control of 3-D rolling contacts in two-arm manipulation," *IEEE Trans. Robot. Automat.*, vol. 13, no. 3, pp. 364–376, Jun. 1997.
[6] N. Sarkar, X. Yun, and V. Kumar, "Control of contact interactions with acatastatic nonholonomic constraints," *Int. J. Robot. Res.*, vol. 16, no. 3, pp. 357–374, Jun. 1997.
[7] Y. Chitour, M. G. Molina, and P. Kokkonen, "The rolling problem: Overview and challenges," in *Geometric Control Theory and Sub-Riemannian Geometry*. Cham, Switzerland: Springer, 2014, pp. 103–122.
[8] G. Lafferriere and H. Sussmann, "Motion planning for controllable systems without drift," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 1991, pp. 1148–1153.
[9] R. M. Murray, "Nilpotent bases for a class of nonintegrable distributions with applications to trajectory generation for nonholonomic systems," *Math. Control Signal Syst.*, vol. 7, no. 1, pp. 58–75, Mar. 1994.
[10] G. Oriolo and M. Vendittelli, "A framework for the stabilization of general nonholonomic systems with an application to the plate-ball mechanism," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 162–175, Apr. 2005.
[11] R. Murray and S. Sastry, "Steering nonholonomic systems in chained form," in *Proc. 30th IEEE Conf. Decis. Control*, Dec. 1991, pp. 1121–1126.
[12] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, Jun. 1995.
[13] A. Bicchi and R. Sorrentino, "Dexterous manipulation through rolling," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, May 1995, pp. 452–457.

[14] Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 62–72, Feb. 1990.
[15] A. Marigo and A. Bicchi, "A local-local planning algorithm for rolling objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, Jun. 2002, pp. 1759–1764.
[16] F. Alouges, Y. Chitour, and R. Long, "A motion-planning algorithm for the rolling-body problem," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 827–836, Oct. 2010.
[17] M. Rehan and M. Reyhanoglu, "Global formulation and motion planning for a sphere rolling on a smooth surface," *Int. J. Control Autom. Syst.*, vol. 16, no. 6, pp. 2709–2717, Dec. 2018.
[18] A. Marigo and A. Bicchi, "Rolling bodies with regular surface: Controllability theory and applications," *IEEE Trans. Autom. Control*, vol. 45, no. 9, pp. 1586–1599, Sep. 2000.
[19] A. A. Agrachev and Y. L. Sachkov, *Control Theory From the Geometric Viewpoint*, vol. 87. Berlin, Germany: Springer-Verlag, 2004.
[20] K. A. Krakowski, F. Atima, and S. Leite, "Why controllability of rolling may fail: A few illustrative examples," in *Pre-Publicacoes do Departamento de Matematica*, nos. 12–26. Univ. Coimbra, Coimbra, Portugal, 2012, pp. 1–30.
[21] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J. P. Laumond, "Stabilization of trajectories for systems with nonholonomic constraints," *IEEE Trans. Autom. Control*, vol. 39, no. 1, pp. 216–222, Jan. 1994.
[22] P. Choudhury and K. M. Lynch, "Rolling manipulation with a single control," *Int. J. Robot. Res.*, vol. 21, nos. 5–6, pp. 475–487, May 2002.
[23] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Rev.*, vol. 59, no. 4, pp. 849–904, Jan. 2017.
[24] W. L. Brogan, *Modern Control Theory*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1991.
[25] P. C. Müller and H. I. Weber, "Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems," *Automatica*, vol. 8, no. 3, pp. 237–246, May 1972.
[26] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 40–52, Mar. 2014.
[27] B. D. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Chelmsford, MA, USA: Courier Corporation, 2007.

J. ZACHARY WOODRUFF (Member, IEEE) received the B.S. degree in mechanical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2013, and the M.S. degree in mechanical engineering from Northwestern University, Evanston, IL, USA, in 2016, where he is currently pursuing the Ph.D. degree in mechanical engineering with the Center for Robotics and Biosystems. His research focuses on modeling, motion planning, and feedback control for dynamic and graspless robotic manipulation. In 2015, he was awarded the National Science Foundation Graduate Research Fellowship (NSF GRF).

SHUFENG REN received the B.S. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017. He is currently pursuing the master's degree in mechanical engineering with the Center for Robotics and Biosystems, Northwestern University. His research interests include trajectory planning, and feedback control for nonholonomic systems.



KEVIN M. LYNCH (Fellow, IEEE) received the B.S.E. degree in electrical engineering from Princeton University and the Ph.D. degree in robotics from Carnegie Mellon University. He is currently a Professor and the Chair of the Mechanical Engineering Department, Northwestern University, and the Director of the Center for Robotics and Biosystems. His research is on robotic manipulation, locomotion, human–robot systems, and robot swarms. He was a recipient of the IEEE Robotics and Automation Early Career Award, the Harashima Award for Innovative Technologies, the National Science Foundation CAREER Award, and the Northwestern's Charles Deering McCormick Professorship of Teaching Excellence. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON ROBOTICS, an alum of the DARPA/IDA Defense Science Study Group, a coauthor of textbooks on robotics and mechatronics, and an instructor of six Coursera online courses forming the Modern Robotics specialization.