

Geodesic Tracing-Based Kinematic Integration of Rolling and Sliding Contact on Manifold Meshes for Dexterous In-Hand Manipulation

Sunyu Wang¹, Arjun S. Lakshmipathy², Jean Oh¹, and Nancy S. Pollard^{1,2}

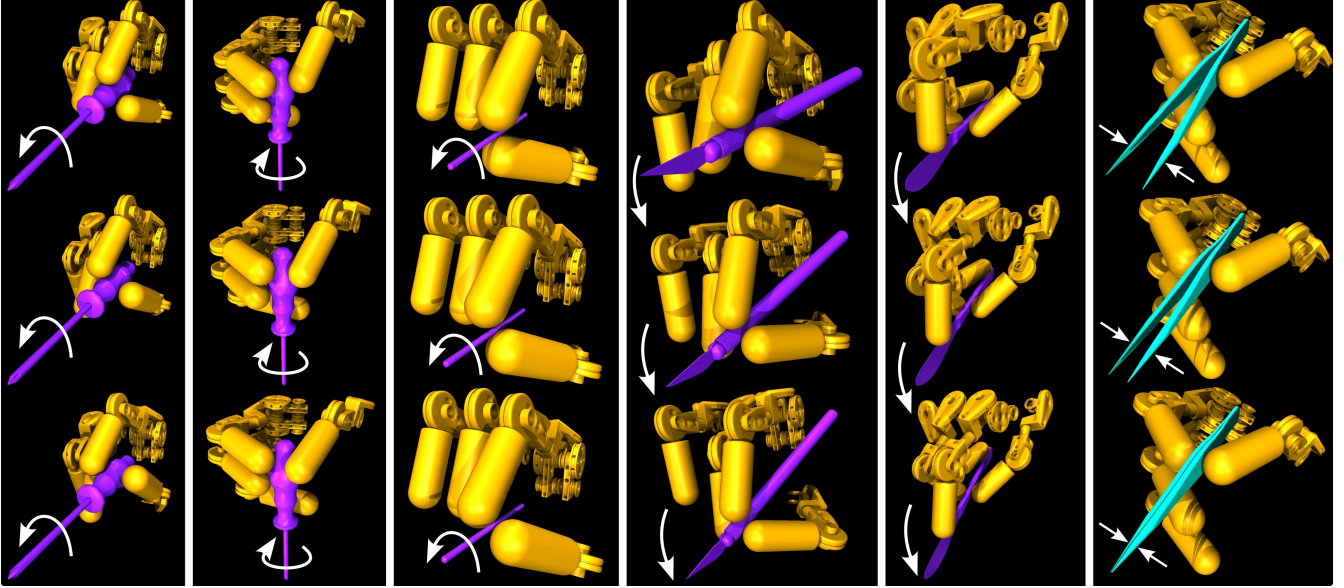


Fig. 1: Snapshots of a robotic hand performing dexterous in-hand manipulation tasks using the integration scheme we developed. Tasks from left to right: 1) horizontally turning a screwdriver, 2) vertically turning the screwdriver, 3) turning an M2 threaded rod, 4) pressing down a knurled hobby knife, 5) pressing down a dining knife, 6) closing tweezers.

Abstract—Reasoning about rolling and sliding contact, or roll-slide contact for short, is critical for dexterous manipulation tasks that involve intricate geometries. But existing works on roll-slide contact mostly focus on continuous shapes with differentiable parametrizations. This work extends roll-slide contact modeling to manifold meshes. Specifically, we present an integration scheme based on geodesic tracing to first-order time-integrate roll-slide contact directly on meshes, enabling dexterous manipulation to reason over high-fidelity discrete representations of an object’s true geometry. Using our method, we planned dexterous motions of a multi-finger robotic hand manipulating five objects in-hand in simulation. The planning was achieved with a least-squares optimizer that strives to maintain the most stable instantaneous grasp by minimizing contact sliding and spinning. Then, we evaluated our method against a baseline using collision detection and a baseline using primitive shapes. The results show that our method performed the best in accuracy and precision, even for coarse meshes. We conclude with a future work discussion on incorporating multiple contacts and contact forces to achieve accurate and robust mesh-based surface contact modeling.

I. INTRODUCTION

Societal expectations have been growing for humanoid robots to contribute to manufacturing, assembly, agriculture, and healthcare [1]–[3]. To meet such expectations, humanoid

robots need to have strong dexterous manipulation capabilities, which remain challenging to achieve largely because modeling and reasoning about contact is difficult.

A seminal work on contact modeling is the rigid-body single-point roll-slide contact kinematics model formulated by Montana in 1988 [4]. This model considers the contacting bodies’ detailed geometries, addresses both rolling and sliding motions, and has seen iterations of improvements and different formulations [5]–[13]. Meanwhile, significant research progress has been made on dexterous manipulation motion planning with roll-slide contact [14]–[28].

Yet, related prior works mostly focused on primitive shapes with globally differentiable parametric functions, whereas many real-world objects cannot be easily modeled in these approaches. Furthermore, decomposing a complex shape into primitive shapes introduces artificial discontinuities, making it difficult to model continuous global contact motion.

In contrast, manifold meshes are performant and versatile in discretely representing realistic shapes—they are the data structure of numerous CAD models and 3D digital art works, the blueprints for computer-aided manufacturing processes such as 3D printing and CNC machining. Hence, extending roll-slide contact modeling to manifold meshes can enable dexterous manipulation to reason over high-fidelity discrete representations of an object’s true geometry beyond contin-

The authors are with the ¹Robotics Institute and the ²Computer Science Department at Carnegie Mellon University, Pittsburgh, USA.

Corresponding author’s contact: sunyuw@andrew.cmu.edu

uous shapes. We believe that this has the potential to unlock new capabilities for multi-finger robotic hands.

To this end, we present a geodesic tracing-based integration scheme to first-order time-integrate roll-slide contact directly on manifold meshes. We used our method to plan motions of a multi-finger robotic hand performing dexterous in-hand manipulation tasks, as shown in Fig. 1. The planning was achieved with a least-squares optimizer that computes the most stable instantaneous grasp by minimizing contact sliding and spinning. Then, we evaluated our method against two baselines—one using collision detection and one using primitive shapes—in two simulation experiments: a hand turning a screwdriver, and a sphere rolling on a ring under different mesh resolutions. The results show that our method performed the best in accuracy and precision, even for coarse meshes, with limitations in handling multiple contacts. We conclude with a discussion on resolving these limitations while incorporating contact forces to create an accurate and robust mesh-based surface contact model.

II. RELATED WORKS

A. Roll-Slide Contact Modeling

The roll-slide contact kinematics model we refer to in this work is a set of first-order differential equations that characterizes the positions and velocities of contact points on two rigid surfaces that roll and slide on each other. Some oldest versions of this model date back to the 1980s [4], [29]. Since then, this subject has been extensively studied in rigid body kinematics [5]–[9], and the results are summarized in robotics textbooks [11], [30]–[33]. Meanwhile, improvements and alternative formulations have been proposed [10], [12], [13]. A key difference among these formulations is their surface parametrizations.

However, these prior works typically concern continuous geometries with global, differentiable parametrizations. For geometries that are difficult to parametrize with closed-form differentiable functions, the methods in these prior works may not be readily applicable. Our work builds on these prior works’ ideas—particularly Montana’s formulation [4]—but focuses on the *discrete* setting with manifold meshes, which do not always have closed-form global parametrizations.

B. Dexterous Motion Planning with Roll-Slide Contact

The roll-slide contact kinematics model has been linked to dexterous manipulation motion planning for multi-finger robotic hands since its inception. Early works on this subject absorbed the roll-slide contact model into the robotic hand’s kinematics and dynamics equations [14]–[18]. Through Jacobians, contact type models, grasp matrix, and inverse kinematics and dynamics, dexterous motions were generated. However, as existing roll-slide contact models mostly rely on differentiable surface parametrizations, these early works only considered primitive shapes as the manipulated objects.

Later, trajectory optimization and policy learning approaches surged in popularity in dexterous manipulation motion planning. As a more classical method, the roll-slide contact model has often acted as a foundational but

supporting part in a larger algorithmic framework. Several works did exploit rolling motions in hand motion planning, but few addressed roll-slide contact kinematics for detailed geometries of general objects [19]–[27].

A closely related work to ours is [28], which employed signed distance field to parametrize general shapes in a trajectory optimization framework that emphasizes differentiability. The authors generated dexterous motions for a multi-finger robotic hand, and impressively demonstrated them in the real world. However, a limitation is that [28] still modeled the manipulated objects as primitive shapes, with the authors themselves noting that accepting meshes as input is generally intractable. Our method offers a solution to this limitation by taking full manifold meshes as input.

III. METHODS

Our method uses Montana’s roll-slide contact kinematics model [4] as the basis. Hence, we will first recapitulate Montana’s theory, and then present our method on top.

A. Induced Velocity of Contact Frame

Consider a general 3D rigid body with a body frame $\{B\}$ fixed to it. Assume that the body has an orthogonal coordinate chart $f : \mathbb{R}^2 \mapsto \mathbb{R}^3$ that describes its surface relative to $\{B\}$. Define a contact frame $\{C\}$, whose origin moves on the surface and whose z -axis is parallel with the surface normal at $\{C\}$ ’s origin. Define $g \in \mathbb{R}^2$ as the minimal-coordinate position of $\{C\}$ ’s origin relative to $\{B\}$ through f , and $\dot{g}$ as its time derivative. Then, $\{C\}$ ’s generalized velocity relative to $\{B\}$ expressed in $\{C\}$, or $\{C\}$ ’s body twist, is

$$V_{BC}^b = \begin{bmatrix} \omega_{BC}^b \\ v_{BC}^b \end{bmatrix} = \begin{bmatrix} EKM \\ TM \\ M \\ 0_{1 \times 2} \end{bmatrix} \dot{g}, \text{ where } E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (1)$$

$\omega_{BC}^b, v_{BC}^b \in \mathbb{R}^3$ are $\{C\}$ ’s angular and linear velocities relative to $\{B\}$ expressed in $\{C\}$, respectively. $M \in \mathbb{R}^{2 \times 2}$ is the metric tensor, $K \in \mathbb{R}^{2 \times 2}$ is the curvature tensor, and $T \in \mathbb{R}^{1 \times 2}$ is the torsion form. Next, define a local frame $\{L\}$ that instantaneously coincides with $\{C\}$ but is *fixed* to the body. This means that $\{L\}$ has the same pose, i.e., position and orientation, as $\{C\}$, but zero velocities relative to $\{B\}$.

B. Roll-Slide Contact Kinematics

Now, consider two rigid bodies indexed 0 and 1. With the above definitions, we have six frames: $\{B_i\}, \{C_i\}, \{L_i\}, i \in \{0, 1\}$. Without loss of generality, define each body’s surface normal to be outward-pointing. Then, an ideal point contact between the two bodies means that their contact frames’ origins coincide and z -axes anti-align. If the contact is maintained, the relative velocities between $\{C_0\}$ and $\{C_1\}$ will be zero except the angular element about the z -axis, which is the spin element. Mathematically,

$$T_{C_0 C_1} = \begin{bmatrix} * & * & 0 & 0 \\ * & * & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, V_{C_0 C_1}^b = \begin{bmatrix} 0_{2 \times 1} \\ * \\ 0_{3 \times 1} \end{bmatrix}. \quad (2)$$

$T_{C_0C_1}$ is the homogeneous transformation representing $\{C_1\}$'s pose relative to $\{C_0\}$. A “*” means any scalar. We refer to (2) as the ideal point contact constraint.

Since $\{L_i\}$ coincides with $\{C_i\}$ but is fixed to body i , the relative twist between $\{L_0\}$ and $\{L_1\}$, $V_{L_0L_1}^b = [\omega_x \ \omega_y \ \omega_z \ v_x \ v_y \ v_z]^\top$, represents the Cartesian rolling and sliding velocities between the two bodies. ω_z is the spinning speed about the contact normal, v_z is the separation/interpenetration speed, and v_x, v_y are the sliding speeds in the contact tangent plane. By twist frame transformation [31], $V_{L_0L_1}^b$ and $V_{C_0C_1}^b$ are related by

$$V_{L_0L_1}^b + V_{B_1C_1}^b = \text{Ad}_{T_{C_1C_0}} V_{B_0C_0}^b + V_{C_0C_1}^b, \quad (3)$$

where Ad is the adjoint. (3) and (1) connect the Cartesian rolling and sliding velocities and the contacts' minimal-coordinate velocities, $\dot{g}_i$'s, of both bodies, forming the governing equations of roll-slide contact kinematics.

C. Body-to-Contact Integration with Collision Detection

After reviewing Montana's theory, we investigate first-order time integration of roll-slide contact on manifold meshes with the target application of dexterous manipulation motion planning. Since what matters is the relative motion between two contacting bodies, without loss of generality, we define body 0 as the reference body whose pose and body twist relative to a stationary world frame $\{W\}$ are known, and body 1 as the moving body. Then, the problem is to time-update the poses of $\{C_0\}$, $\{C_1\}$, and $\{B_1\}$ relative to $\{W\}$, given both bodies' poses, their contacts' poses, and the relative contact twist, $V_{L_0L_1}^b$, at the current time step.

An intuitive solution is to integrate the bodies' velocities first, and then update the contacts based on closest points. Applying twist frame transformation [31] to $V_{L_0L_1}^b$ yields

$$V_{L_0L_1}^b = \text{Ad}_{T_{L_1B_1}} V_{WB_1}^b - \text{Ad}_{T_{L_1B_0}} V_{WB_0}^b, \quad (4)$$

where $V_{WB_i}^b$ is $\{B_i\}$'s body twist relative to $\{W\}$. Since $T_{WB_0}, T_{WB_1}, T_{B_0L_0}, T_{B_1L_1}, V_{WB_0}^b$, and $V_{L_0L_1}^b$ are known, we can solve (4) for $V_{WB_1}^b$ and use any standard integrator to integrate the body twists and update the poses.

Next, we update the contact frames. If the two bodies interpenetrate, we update each body's contact point as the point on that body closest to the penetration depth-weighted average of the interpenetration points; the deeper the penetration, the larger the weight. If the two bodies separate, we update each body's contact point as the closest point on that body to the other. Then, we update the contact frames' z -axes as the linearly interpolated vertex normals at the updated contact points. The x - and y -axes can be updated arbitrarily as long as they form valid rotation matrices with the z -axes. For ease of visualization, we update the x -axes as the vectors pointing from the current contacts to the updated contacts, orthogonally projected onto the z -axes. Lastly, we update the y -axes with cross product and normalize all three axes to unit vectors. We implemented this method as a baseline using a collision detector [34]–[36], a closest point querier [37], and the 4th-order Runge-Kutta (RK4) integrator in MuJoCo [38], with all dynamics—including gravity—turned off.

D. Contact Stabilization

Despite its intuitiveness, the collision detection-based baseline cannot always keep the two bodies in an ideal point contact. This is because explicit time integration of velocities can produce drift and motions in the contact tangent plane, causing interpenetration or separation.

To resolve this issue, we implemented an inverse kinematics-inspired contact stabilizer [33], shown in Algorithm 1. At each time step, the stabilizer acts as a reactive feedback controller and generates a stabilizing twist V_{stbl} that, if added to $V_{L_0L_1}^b$, pushes the current $T_{C_0C_1}$ closer to the ideal $T_{C_0C_1}$ in (2). In Algorithm 1, $T_{C_0C_1}$ is the two contact frames' current relative pose. $T_{C_0C_1,ideal}$ is the $T_{C_0C_1}$ in (2). ω_{stbl}, v_{stbl} are the angular, linear velocities in V_{stbl} . $\text{expm}(\cdot)$ and $\text{logm}(\cdot)$ are matrix exponential and matrix logarithm.

Algorithm 1 Contact Stabilizer

$$\begin{aligned} T_{error} &\leftarrow T_{C_0C_1}^{-1} T_{C_0C_1,ideal} \\ T_{error}[:, : 3] &\leftarrow \text{expm}(a_{z_{error}}) \\ V_{stbl} &\leftarrow \text{logm}(T_{error}) \\ V_{L_0L_1}^b &\leftarrow V_{L_0L_1}^b + [k_\omega \omega_{stbl}^\top \quad k_v v_{stbl}^\top]^\top \end{aligned}$$

Unlike in the inverse kinematics case, the two contact frames' x - and y -axes do not need to coincide. Hence, before taking the matrix logarithm, we replace the rotation matrix in T_{error} in Algorithm 1 with a rotation matrix computed from $a_{z_{error}}$, an axis-angle vector that anti-aligns only the two contact frames' z -axes. Lastly, two independently tuned gains, k_ω and k_v , are applied to the angular and linear velocities in V_{stbl} , before it is added to $V_{L_0L_1}^b$. We implemented this contact stabilizer for the collision detection-based baseline.

E. Contact-to-Body Integration with Geodesic Tracing

Another approach to time-integrating roll-slide contact on meshes is to apply Montana's theory. However, a mesh is generally non-smooth and does not always have a differentiable parametrization in a global coordinate chart. Also, our target application is dexterous manipulation motion planning. It is inadequate to define a local coordinate chart based on a prescribed differentiable space curve as the contact's path.

Our solution is to use the contact frame itself as the coordinate chart that locally parametrizes the surface in the Euclidean space, and update the contact frame at every time step. Hence, this local coordinate chart only needs to stay valid for one time step. Specifically, we define the minimal-coordinate position g in (1) as the signed geodesic distances from the contact frame's origin in the positive x - and y -directions. To time-integrate the contact point, we trace the geodesic defined by g relative to the body frame $\{B\}$ using explicit Euler and a mesh-based geodesic tracer [39]. This ensures that the integrated contact point stays on the mesh. After the integration, we update the contact frame with the linearly interpolated vertex normals as its z -axis. The other two axes can be arbitrary as long as they form a valid rotation matrix with the z -axis, and we update them in the same

way as in the collision detection-based baseline. Algorithm 2 shows this procedure with time step size Δt .

Algorithm 2 Geodesic Tracing-Based Contact Integrator

$\dot{g}_0, \dot{g}_1 \leftarrow$ roll-slide kinematics $(V_{L_0L_1}^b, K_0, K_1)$ (1), (3)
 $p_{B_iC_i} \leftarrow$ trace geodesic $(\dot{g}_i \Delta t), i \in \{0, 1\}$ [39]
 $R_{B_iC_i} \leftarrow$ update contact frame $(p_{B_iC_i})$

$R_{B_iC_i}$ and $p_{B_iC_i}$ are $\{C_i\}$'s rotation matrix and Cartesian position relative to $\{B_i\}$, respectively. Based on g_i 's definition, the metric tensor M in (1) is the identity matrix. K_i is the curvature tensor at the current contact point on body i , which is estimated using a central difference based on geodesic tracing and linearly interpolated vertex normals. Since the x - and y -axes are not tracked, we ignore the torsion form and the spin element in (1).

After updating $R_{B_iC_i}$ and $p_{B_iC_i}$, since the reference body's updated pose is known, a rigid body transformation would yield the moving body's pose that exactly satisfies the ideal point contact constraint (2)

$$T_{WB_1} = T_{WB_0} T_{B_0C_0} T_{C_0C_1}(\psi) T_{B_1C_1}^{-1}. \quad (5)$$

We refer to (5) as exact mating. However, since spin is not tracked, the top-left 2×2 block of $T_{C_0C_1}$ is unknown, as in (2). This means that the moving body has spin as an unconstrained degree of freedom (DoF), which we parametrize with a scalar ψ . To resolve this issue, we compute a "shadow orientation" of the moving body, R'_{WB_1} , by integrating its angular velocity with RK4. Though R'_{WB_1} may not satisfy the contact constraint, it helps determine ψ , which we solve for by minimizing the axis-angle vector error between R'_{WB_1} and R_{WB_1} with iterative inverse kinematics [33].

The method above ensures that the integrated frames exactly satisfy the ideal point contact constraint (2). Yet, for closed kinematic chains, such as a multi-finger grasp, additional structural constraints are present, which may render the entire kinematic system overdetermined. In those cases, we use the contact stabilizer in Algorithm 1 instead.

F. Hand Motion Planning with Roll-Slide Contact

Consider a multi-finger hand in contact with an object. The kinematic objective of dexterous in-hand manipulation is to achieve desired poses and twists of the object via the hand's movement. Hence, we define the object as the reference body with a known initial pose and twist trajectory.

To plan the hand's motion, we let the hand maintain the most stable instantaneous grasp. Kinematically, this means to minimize the sliding and spinning speeds at each contact while preventing separation and interpenetration, which translates into a constrained least-squares optimization

$$\dot{u}_H = \operatorname{argmin} (\omega_z^2 + v_x^2 + v_y^2) \text{ subject to } v_z = 0. \quad (6)$$

ω_z and v_x, v_y, v_z are the spin and the linear elements of $V_{L_0L_1}^b$, respectively. $\dot{u}_H$ is the hand's minimal-coordinate velocities, usually finger joint and palm velocities. To apply (6) to a hand holding an object, we employ (3) and

Jacobians to express $V_{L_0L_1}^b$ in terms of minimal-coordinate velocities: $V_{L_0L_1}^b = J_H \dot{u}_H + J_O \dot{u}_O$, where J_H and J_O are the hand's and the object's Jacobians, respectively, and $\dot{u}_O$ is the object's minimal-coordinate velocity, which is known. Substituting this equation into (6) yields $\dot{u}_H$. We implemented this procedure with an off-the-shelf constrained least-squares optimizer [40], [41]. In addition, we added two regularization terms to (6), one to encourage finger movement and discourage palm movement, and the other to smooth the trajectory of $\dot{u}_H$.

IV. RESULTS AND DISCUSSION

A. Hand Motion Planning with Meshes from the Internet

We downloaded five objects' meshes from online [42] and imported them to the simulation for hand motion planning. The objects are shown in Fig. 1. The screwdriver has concavities for grip, the threaded rod has threads, and the hobby knife's handle is knurled—all are intricate geometric features difficult to model with primitive shapes or differentiable parametrizations. The tweezers have one revolute joint for each arm at its base. The hand consists of four identical 3-DoF fingers, indexed 1–4 from the thumb to the ring finger. Each finger has a capsule-shape distal link, can flex/extend and adduct/abduct at the MCP joint, and flex/extend at the IP joint. The palm is modeled as a 6-DoF floating base.

To plan the hand's motion, we first manually configured the hand into the pose for holding each object. Then, we prescribed each object's twist as a constant, and planned the finger joint and palm velocities with (6) at every time step. Each object moved in one direction for 5 seconds, and immediately moved in the opposite direction with the same speed for another 5 seconds. The simulation ran at 100 Hz. Fig. 1 shows the motion snapshots in the first 5 seconds.

Overall, the hand motions are realistic, especially the twisting motions for the screwdriver and the threaded rod, which engage the fingers' adduction/abduction DoF. However, the collision detection-based baseline produced noisy contact movements, whereas the geodesic tracing-based method produced interpenetration between the fingers and the threaded rod.

B. Quantitative Experiment: Hand Turning Screwdriver

To quantitatively compare these methods, we computed four metrics for each contact during the horizontal screwdriver motion shown in the first column of Fig. 1. The metrics are explained below. The ideal values mean the metrics for an ideal point contact with zero sliding.

- 1) Separation, computed as the mean of all interpenetration points' depths if two meshes interpenetrate, or the minimum distance between two meshes if they separate. A positive, negative value means separation, interpenetration, respectively. Ideal value = 0 mm.
- 2) Alignment, computed as the angle between the contact frames' z -axes on two meshes. Ideal value = 180° .
- 3) Slippage, computed in three steps: a) record the geodesic distance traveled by the contact on each mesh at every time step, b) sum these distances from start

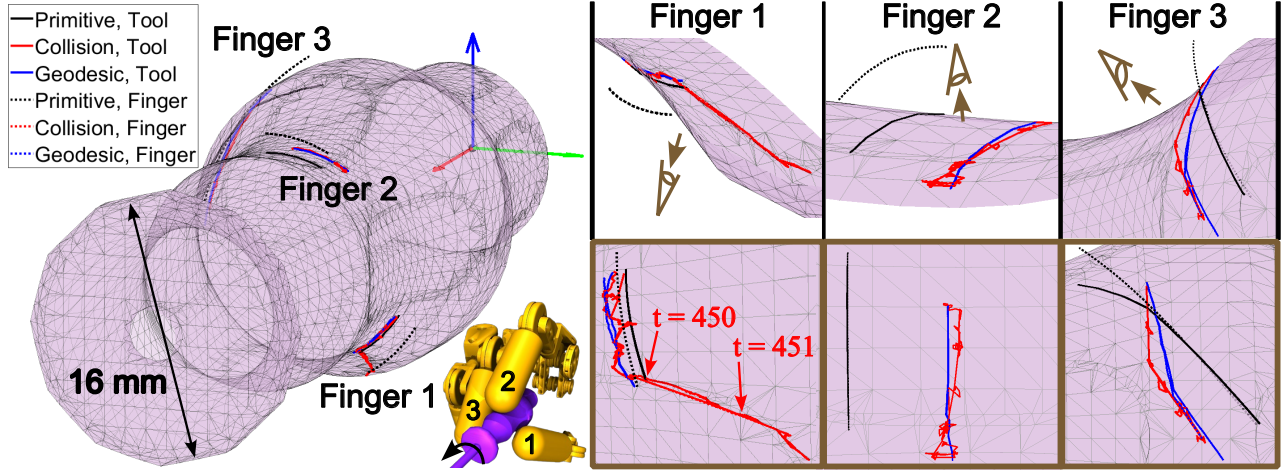


Fig. 2: Contact paths on the screwdriver’s handle relative to the screwdriver’s body frame. On the right, each column contains a detail view and a normal view of each finger’s contact paths. The brown eye symbol and arrow indicate the incident light direction in which the normal view in the corresponding bottom row in brown box is captured. The red arrows point at finger 1’s contact points produced by the collision baseline at time steps 450 and 451, respectively. The “Tool” and “Finger” suffixes in the legend mean that the corresponding contact path is on the screwdriver or the finger.

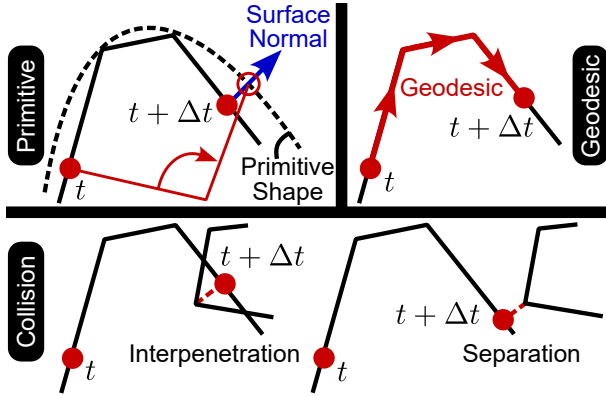


Fig. 3: One-time step integration from the same contact point at time t with the primitive baseline (top left), the geodesic method (top right), and the collision baseline (bottom) on a 3-edge 2D mesh. The black solid lines are the mesh. The red points are the contact points.

time to current time, c) subtract the sums between the two meshes. Ideal value = 0 mm.

- 4) Sliding, computed as the sliding speed $\sqrt{v_x^2 + v_y^2}$ from the least-squares (6). Ideal value = 0 mm/s.

Additionally, we added a primitive shape-based baseline by fitting an ellipsoid to the screwdriver’s handle and a cylinder and a hemisphere to the finger’s distal link. Then, we computed the contacts’ minimal coordinate velocities with (1) and cylindrical- and spherical-coordinate parametrizations. Lastly, we integrated these velocities using explicit Euler without contact stabilization as in Algorithm 1 or exact mating as in (5). Since the primitive shapes are abstract models and the meshes are the true geometries, we computed the metrics for this baseline by projecting the contact point

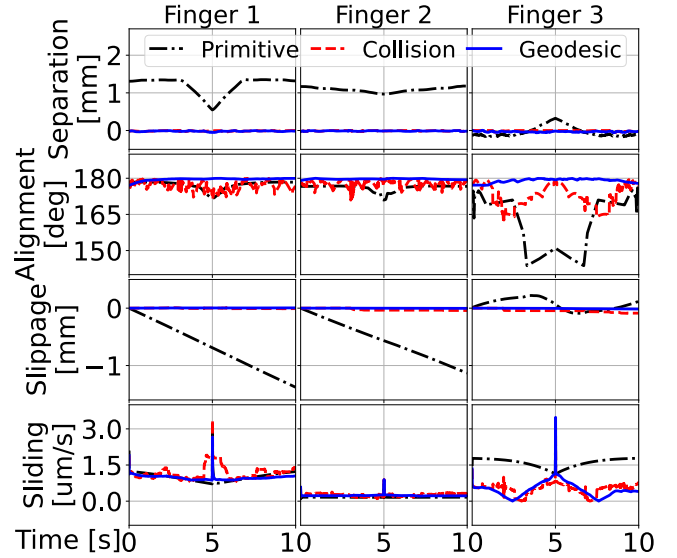


Fig. 4: Screwdriver experiment metrics. Each column represents a finger. The subplots on each row have the same vertical axis ticks. “um” means micrometer.

on the primitive shape onto its corresponding mesh using ray tracing. Fig 3 compares the primitive shape-based baseline with the collision detection-based baseline and the geodesic tracing-based method. For conciseness, we abbreviate the three methods’ names to “the primitive baseline”, “the collision baseline”, and “the geodesic method”.

Fig. 2 and 4 show the results. We make the following observations: 1) The fingers maintained contact with the screwdriver for the collision baseline and the geodesic method, but lost contact for the primitive baseline, as shown by the “Separation” metric in Fig. 4. Meanwhile in Fig. 2, if a finger is in contact with the screwdriver, the contact path

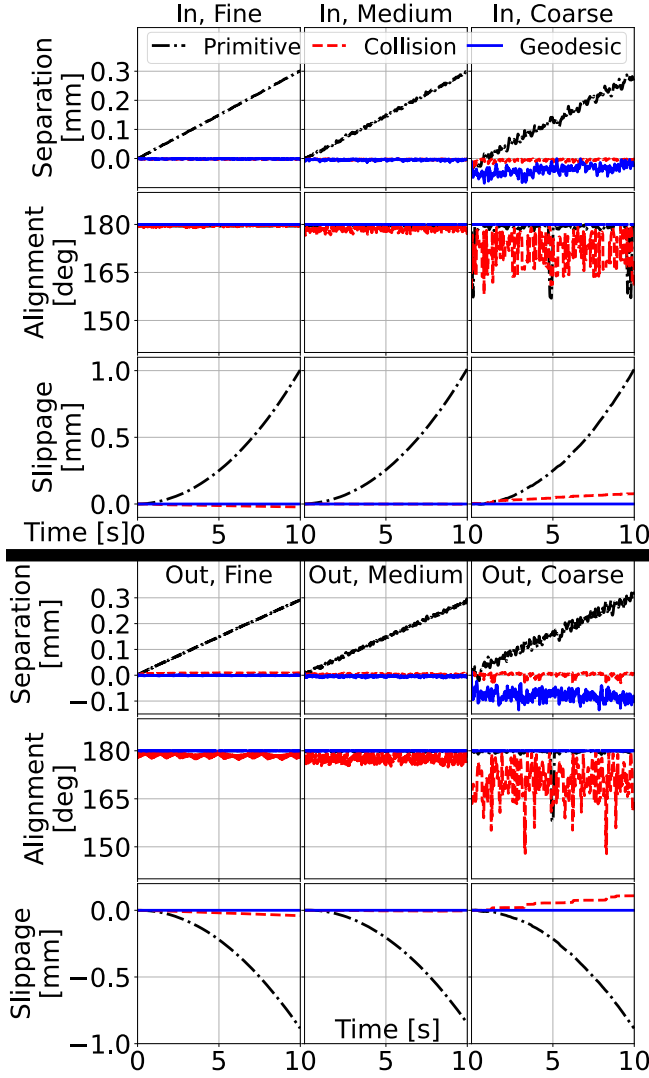


Fig. 5: Sphere experiment metrics. The top and bottom 9 plots with “In” and “Out” in the titles are for the sphere rolling inside and outside the ring, respectively. Each column represents a mesh resolution.

on the screwdriver and the contact path on the finger should coincide, which is not the case for the primitive baseline, indicating separation. This is expected, as an ellipsoid cannot accurately fit the screwdriver’s detailed geometries, and nothing in the primitive baseline prevents the explicit integration drift. 2) Both the collision baseline and the geodesic method produced metrics close to the ideal values. But in Fig. 4, the “Alignment” and “Sliding” trajectories for the geodesic method are smoother. Also, in Fig. 2’s bottom row, the geodesic method’s contact paths resemble smooth versions of the collision baseline’s contact paths in parts where the two methods’ paths overlap. Both phenomena indicate that the geodesic method is more precise. This is reasonable, as the geodesic method traces the incremental geodesics integrated from the geodesic velocities exactly on the meshes, preserving a sense of spatial continuity for the contact frames. In

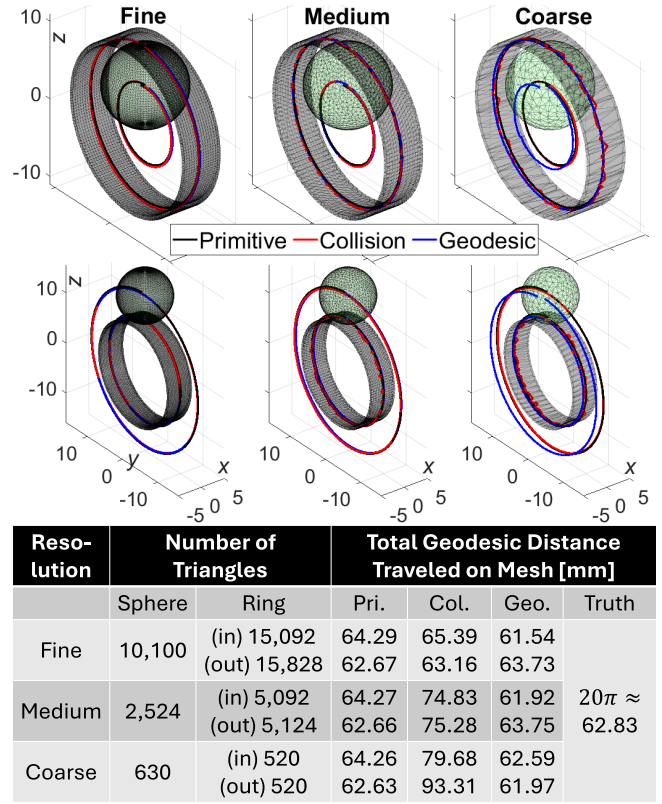


Fig. 6: (Top) Sphere experiment paths. Each row represents a side of the ring. Each column represents a mesh resolution. The paths on the rings are the contacts’ paths. The paths not on the rings are the paths of the sphere’s centroid. The length unit is millimeters. (Bottom) Sphere experiment tabular data. Pri., Col., Geo., and Truth stand for the primitive baseline, the collision baseline, the geodesic method, and the ground truth, respectively.

contrast, the collision baseline incorporates only positions without considering velocities. Consequently, a contact point updated by the collision baseline can be anywhere on the mesh. 3) Shown in Fig. 2’s bottom row, from time step $t = 450$ to $t = 451$, finger 1’s contact point produced by the collision baseline traveled for a considerably long geodesic distance. When this happened, finger 1’s “Separation”, “Alignment”, and “Slippage” metrics remained close to their ideal values. Only the collision baseline exhibited this behavior. We interpret this as a discontinuous contact shift in a situation where more than one possible contact point exist due to the geometry around the contact. This behavior occurred only for the collision baseline because it does not preserve spatial continuity and considers all possible contact points on the entire mesh. In contrast, the geodesic method obeys the point contact model and integrates velocities to positions, making it unlikely to discontinuously break a contact and establish a distinct new contact. This also explains why the collision baseline is less prone to interpenetration than the geodesic method for the threaded rod. 4) In Fig. 4, the sliding speeds are small relative to the screwdriver’s size,

indicating that the fingers mostly rolled on the screwdriver. The spikes at the 5th second are due to the abrupt reversal of the screwdriver’s prescribed velocity.

C. Quantitative Experiment: Sphere Rolling on Ring

We compared the three methods in a simpler experiment of a sphere rolling on the inside and outside of two rings. The side of the ring on which the sphere rolls always has a diameter of 20 mm. The sphere’s diameter is 10 mm. Both the sphere and the rings have fine, medium, and coarse mesh resolutions. The simulation ran at 100 Hz for 10 seconds. The sphere’s angular velocity was prescribed so that the sphere would travel for exactly one revolution around the ring in 10 seconds if the shapes were continuous. The sphere purely rolls, meaning that the contact sliding speed is always zero. Hence, we changed the “Sliding” metric from the screwdriver experiment to the total geodesic distance traveled, which is averaged between the contacts on the sphere and the ring. This metric’s ground truth is the ring’s circumference under continuous geometry: 20π . Since the kinematic system is an open chain, we used the exact mater in (5) instead of the contact stabilizer in Algorithm 1 for the geodesic method.

Fig. 5 and 6 show the results. Our observations are as follows: 1) Without contact stabilization or exact mating, the primitive baseline drifted quickly and could not maintain the contact. Though these drifts are not obvious on the contact paths in Fig. 6, they are significantly more severe than the results from the other two methods, as reflected by the “Separation” metric in Fig. 5. 2) The collision baseline’s metrics and contact paths became noisier, and its total geodesic distance traveled drastically deviated from the ground truth as mesh resolution decreased. This is because, for the collision baseline, the contact traveled increasingly laterally on the ring for coarser meshes, generating the zig-zag contact paths in Fig. 6. 3) The geodesic method almost exactly maintained the pure rolling motion and the ideal point contact for fine and medium mesh resolutions. For coarse meshes, interpenetration became larger and the sphere’s traveling direction was off from the ring’s center, causing a slightly spiral path of the sphere’s centroid. This is because the meshes developed flatter faces and sharper corners as their resolutions decreased, whereas their vertex normals were still linearly interpolated and the exact mater (5) was used. When the meshes were too coarse, the interpolation could no longer obtain surface normals close to those on a smooth sphere or ring. Thus, interpenetration and movement direction drift became more likely. Nevertheless, the geodesic method’s total geodesic distance traveled remained close to the ground truth for coarse meshes, as shown in Fig. 6.

D. Limitations and Future Work

The geodesic method has limitations in handling multiple contact points. As the screwdriver experiment showed, multi-contact situations are hard to avoid for complex shapes. Hence, a meaningful future direction would be to extend roll-slide contact modeling to surface contact between meshes. A promising approach to achieving this is to combine the

geodesic method and the collision baseline, as they appear complementary. The geodesic method excels at accuracy and precision, but is local. The collision baseline excels at recognizing all contacts on the entire mesh, but lacks precision and robustness against mesh resolution degradation. Therefore, the two methods can be combined into a hierarchical integration scheme: The collision baseline globally detects the most likely contacts in low precision, and the geodesic method finely integrates each contact.

Another limitation is that this work concerns first-order kinematic integration, which involves poses and velocities but not forces. This means that the planned motions are kinematically feasible, but may not be feasible when gravity, friction, and inertial forces are present. Despite this, considering the complex kinematic and contact constraints in dexterous multi-finger in-hand manipulation, the planned motions contain detailed information about roll-slide behaviors, and could be close to well-conditioned trajectories feasible in the real world, which could be useful for downstream trajectory optimization, policy learning, and animation tasks.

Meanwhile, we envision the incorporation of soft contact models, such as pressure field contact [43], [44], to produce the center of pressure as a representative contact point. This could enable a point contact model to describe the essential movement trend of a surface contact patch for motion planning, which is an exciting direction to explore.

V. CONCLUSIONS

This work presents a geodesic tracing-based method to first-order time-integrate roll-slide contact directly on manifold meshes. The significance is that our method enables roll-slide contact modeling and dexterous manipulation motion planning to reason over high-fidelity discrete representations of an object’s true geometry, beyond primitive shapes and shapes with differentiable parametrizations. Our method’s technical core is to use the contact frame as the coordinate chart to locally parametrize the surface in the Euclidean space, and apply Montana’s theory with a mesh-based geodesic tracer. We used our method to plan the motion of a multi-finger robotic hand performing dexterous in-hand manipulation tasks. The results are qualitatively realistic. Then, we quantitatively evaluated our method in two simulation experiments against a baseline using collision detection and a baseline using primitive shapes. The results show that our method performed the best in accuracy, precision, and robustness against low mesh resolution, with limitations in handling multiple contacts. Lastly, we discuss future directions to complementarily combine our geodesic method and the collision baseline, and to incorporate contact forces via soft contact models and the center of pressure.

ACKNOWLEDGMENT

This research was partially supported by NSF Convergence Accelerator award ITE-2344109, by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health under Award Number R01EB036842, and by the RAI Institute.

REFERENCES

- [1] “Humanoid robotics will transform manufacturing and the global economy,” *Forbes*, 2024, accessed May 9 2024. [Online]. Available: <https://www.forbes.com/sites/amirhusain/2024/11/04/humanoid-robotics-will-transform-manufacturing-and-the-global-economy/>
- [2] “Humanoid robots at work: What executives need to know,” *Bain & Company*, 2025, accessed May 9 2024. [Online]. Available: <https://www.bain.com/insights/humanoid-robots-at-work-what-executives-need-to-know/>
- [3] B. K. Burke, “Humanoid robots in healthcare: The future is here—but are we ready?” *California Association of Healthcare Leaders*, 2025, accessed May 14 2024. [Online]. Available: <https://ache-cahl.org/articles/humanoid-robots-in-healthcare-the-future-is-here-but-are-we-ready/>
- [4] D. J. Montana, “The kinematics of contact and grasp,” *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 17–32, 1988. [Online]. Available: <https://doi.org/10.1177/027836498800700302>
- [5] Z. Li and J. Canny, “Motion of two rigid bodies with rolling constraint,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 62–72, 1990.
- [6] L. Cui and J. S. Dai, “A coordinate-free approach to instantaneous kinematics of two rigid objects with rolling contact and its implications for trajectory planning,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 612–617.
- [7] —, “A darboux-frame-based formulation of spin-rolling motion of rigid objects with point contact,” *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 383–388, 2010.
- [8] —, “A polynomial formulation of inverse kinematics of rolling contact,” *Journal of Mechanisms and Robotics*, vol. 7, no. 4, p. 041003, 11 2015. [Online]. Available: <https://doi.org/10.1115/1.4029498>
- [9] L. Cui, J. Sun, and J. S. Dai, “In-hand forward and inverse kinematics with rolling contact,” *Robotica*, vol. 35, no. 12, p. 2381–2399, 2017.
- [10] V. Bulut, “Geometric kinematic approach to rigid objects with point contact based on a local-surface frame,” *Sakarya University Journal of Science*, vol. 22, no. 6, p. 1787–1792, 2018.
- [11] L. Cui and J. S. Dai, *Sliding–Rolling Contact and In-Hand Manipulation*. WORLD SCIENTIFIC (EUROPE), 2020. [Online]. Available: <https://www.worldscientific.com/doi/abs/10.1142/q0249>
- [12] M. Aydinalp, M. Kazaz, and H. H. Uğurlu, “The inverse kinematics of rolling contact of timelike curves lying on timelike surfaces,” *Communications Faculty of Sciences University of Ankara Series A1 Mathematics and Statistics*, vol. 68, no. 2, p. 1879–1894, 2019.
- [13] —, “The forward kinematics of rolling contact of timelike curves lying on timelike surfaces,” *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, vol. 16, pp. 215–224, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:221903831>
- [14] Z. Li, P. Hsu, and S. Sastry, “Grasping and coordinated manipulation by a multifingered robot hand,” *The International Journal of Robotics Research*, vol. 8, no. 4, pp. 33–50, 1989. [Online]. Available: <https://doi.org/10.1177/027836498900800402>
- [15] A. Cole, J. Hauser, and S. Sastry, “Kinematics and control of multi-fingered hands with rolling contact,” *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 398–404, 1989.
- [16] J. Trinkle, “A quasi-static analysis of dextrous manipulation with sliding and rolling contacts,” in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 788–793 vol.2.
- [17] D. Montana, “The kinematics of multi-fingered manipulation,” *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 491–503, 1995.
- [18] J. Chen and M. Zribi, “Control of multifingered robot hands with rolling and sliding contacts,” *International Journal of Advanced Manufacturing Technology*, vol. 16, pp. 71–77, 01 2000.
- [19] H. Jeong and J. Cheong, “In-hand rolling motion planning using independent contact region (icr) with guaranteed grasp quality margin,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3239–3244.
- [20] Y. Bai and C. K. Liu, “Dexterous manipulation using both palm and fingers,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1560–1565.
- [21] E. Noohi, H. Moradi, S. Parastegari, and M. N. Ahmadabadi, “Object manipulation using unlimited rolling contacts: 2-d kinematic modeling and motion planning,” *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 790–797, 2015.
- [22] A. Nakashima, “A motion planning for grasp and manipulation system based on rolling motion,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 167–174.
- [23] T. Endo, N. Shiratani, K. Yamaguchi, and F. Matsuno, “Grasp and orientation control of an object by two euler–bernoulli arms with rolling constraints,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 12, p. 121010, 09 2019. [Online]. Available: <https://doi.org/10.1115/1.4044718>
- [24] W. S. Cortez, C. K. Verginis, and D. V. Dimarogonas, “A distributed, event-triggered, adaptive controller for cooperative manipulation with rolling contacts,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3120–3133, 2023.
- [25] L. Zong, G. Liang, and T. L. Lam, “Kinematics modeling and control of spherical rolling contact joint and manipulator,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 738–754, 2023.
- [26] L. Tang, Y.-B. Jia, and Y. Xue, “Robotic manipulation of hand tools: The case of screwdriving,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 883–13 890.
- [27] D. Livnat, M. M. Bilevich, and D. Halperin, “Tight motion planning by riemannian optimization for sliding and rolling with finite number of contact points,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 333–14 340.
- [28] F. Yang, T. Power, S. A. Marinovic, S. Iba, R. S. Zarrin, and D. Berenson, “Multi-finger manipulation via trajectory optimization with differentiable rolling and geometric constraints,” *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 5170–5177, 2025.
- [29] C. Cai and B. Roth, “On the spatial motion of a rigid body with point contact,” in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 686–695.
- [30] M. Mason and J. Salisbury, *Robot Hands and the Mechanics of Manipulation*. MIT Press, 1985.
- [31] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. CRC Press, 1994.
- [32] M. Mason, *Mechanics of Robotic Manipulation*, ser. Intelligent Robotics and Autonomous Agents series. MIT Press, 2001. [Online]. Available: <https://books.google.com/books?id=Ngdeu3go014C>
- [33] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. USA: Cambridge University Press, 2017.
- [34] D.-H. et al., “trimesh.” [Online]. Available: <https://trimesh.org/>
- [35] mikeh, “python-fcl.” [Online]. Available: <https://github.com/BerkeleyAutomation/python-fcl>
- [36] —, “Fcl – the flexible collision library.” [Online]. Available: <https://github.com/flexible-collision-library/fcl>
- [37] R. Sawhney, “Fcpw: Fastest closest points in the west,” 2021. [Online]. Available: <https://github.com/rohan-sawhney/fcpw>
- [38] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
- [39] N. Sharp, K. Crane et al., “Geometry central: A modern c++ library of data structures and algorithms for geometry processing,” 2019. [Online]. Available: <https://geometry-central.net>
- [40] S. Caron, D. Arnström, S. Bonagiri, A. Dechaume, N. Flowers, A. Heins, T. Ishikawa, D. Kenefake, G. Mazzamuto, D. Meoli, B. O’Donoghue, A. A. Oppenheimer, A. Pandala, J. J. Quiroz Omaña, N. Rontsis, P. Shah, S. St-Jean, N. Vitucci, S. Wolfers, F. Yang, @bdehaisse, @MeindertHH, @rimaddo, @urob, @shaolanlu, A. Khalil, L. Kozlov, and A. Groudiev, “qpsolvers: Quadratic Programming Solvers in Python,” 2025. [Online]. Available: <https://github.com/qpsolvers/qpsolvers>
- [41] P. J. Goulart and Y. Chen, “Clarabel: An interior-point solver for conic programs with quadratic objectives,” 2024.
- [42] “Free cad designs, files & 3d models — the grabcad community library,” 2025, accessed on May 8 2025. [Online]. Available: <https://grabcad.com/library>
- [43] R. Elandt, E. Drumwright, M. Sherman, and A. Ruina, “A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 8238–8245.
- [44] J. Masterjohn, D. Guoy, J. Shepherd, and A. Castro, “Velocity level approximation of pressure field contact patches,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 593–11 600, 2022.