

RESEARCH ARTICLE 

# An energy-based numerical continuation approach for quasi-static mechanical manipulation

Lin Yang<sup>1</sup> , Huu-Thiet Nguyen<sup>1</sup>, Chen Lv<sup>1</sup>, Domenico Campolo<sup>1</sup> and Franco Cardin<sup>2</sup>

<sup>1</sup>School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, Singapore

<sup>2</sup>Dipartimento di Matematica “Tullio Levi-Civita”, Università degli Studi di Padova, Padova, Italy

**Corresponding author:** Domenico Campolo; Email: [d.campolo@ntu.edu.sg](mailto:d.campolo@ntu.edu.sg)

**Received:** 04 October 2024; **Revised:** 29 January 2025; **Accepted:** 04 February 2025


**Keywords:** adaptive ordinary differential equations; energy-based approach; equilibrium manifold; haptic metric

## Abstract

Robotic manipulation inherently involves contact with objects for task accomplishment. Traditional motion planning techniques, while having shown their success in collision-free scenarios, may not handle manipulation tasks effectively because they typically avoid contact. Although geometric constraints have been introduced into classical motion planners for tasks that involve interactions, they still lack the capability to fully incorporate contact. In addition, these planning methods generally do not operate on objects that cannot be directly controlled. In this work, building on a recently proposed framework for energy-based quasi-static manipulation, we propose an approach to manipulation planning by adapting a numerical continuation algorithm to compute the equilibrium manifold (EM), which is implicitly derived from physical laws. By defining a manipulation potential energy function that captures interaction and natural potentials, the numerical continuation approach is integrated with adaptive ordinary differential equations that converge to the EM. This allows discretizing the implicit manifold as a graph with a finite set of equilibria as nodes interconnected by weighted edges defined via a haptic metric. The proposed framework is evaluated with an inverted pendulum task, where the explored branch of the manifold demonstrates effectiveness.

## Impact statement

This study addresses the problem of robotic manipulation by combining energy-based quasi-static assumptions with numerical continuation techniques. The quasi-static assumption that all objects remain in equilibrium leads to the generation of an EM. We define the configuration space using natural split variables that encompass both the robot and other objects. Meanwhile, by introducing a manipulation potential that effectively captures the mechanical interactions between the robot and its environment, we enhance the robots' capabilities to plan and execute tasks that require contact, which is a notable gap in conventional motion planning techniques. To efficiently explore the EM under physical constraints, adaptive ordinary differential equations (ODEs) are developed, which help avoid the burden of random sampling large datasets in ambient space. Additionally, we introduce a haptic obstacle, which is represented by a squared-Hessian metric, to terminate the ODE when the mechanical system approaches physical instability. The effectiveness of the proposed framework is demonstrated through a manipulation task involving an inverted pendulum, where the inherent property of the inverted pendulum is ultimately proved.

 This research article was awarded Open Materials badges for transparent practices. See the Data Availability Statement for details.

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.



## 1. Introduction

Manipulation is the process of using one's hands to rearrange one's environment (Mason, 2001). Robotic manipulation requires the robot to establish (typically stable) contact with objects to complete specific tasks (Suomalainen et al., 2022). Consequently, motion planning is essential for the generation of a trajectory that successfully accomplishes these tasks. However, classical motion planning algorithms may not fully capture force interactions, as they typically avoid contact (S. M. LaValle, 2006). In manipulation scenarios, while avoiding unintended collisions is crucial, making contact with objects is often necessary to complete tasks. A prevalent approach to modeling contact between objects is the quasi-static assumption, where equilibrium equations are derived from the point of contact and inertial terms are considered negligible in the dynamic equation (Whitney et al., 1982). This approach is popular in mechanical assembly tasks (Salem and Karayiannidis, 2020; Turlapati and Campolo, 2022; Yang et al., 2023). Notably, a recently proposed framework for energy-based quasi-static manipulation reframes quasi-static manipulation as a planning problem (Campolo and Cardin, 2023a, 2023b), which: (i) captures interaction energy between the robot and the object, where the gradient of energy represents the interaction force, referred to as haptics, and (ii) treats the quasi-static assumption as a physical law, ensuring that all objects remain in equilibrium. In the framework, the equilibria are defined through an implicit manifold. As our work builds on this framework, determining the implicit manifold and thoroughly exploring it becomes the focus of our research.

Navigation on implicit manifolds can be considered a planning problem. For conventional kinematics planning, sampling-based methods have been around for decades, such as rapidly exploring random trees (RRT) (S. LaValle, 1998). These methods typically involve sampling and branching within the configuration space but could be blocked in the presence of narrow corridors (Jaillet and Porta, 2012). Meanwhile, the classical sampling-based approach is likely to generate invalid samples. Hence, the numerical continuation method is introduced, with a local planner generating a motion in the local tangent plane, and projecting back to the implicitly defined manifold, which is always derived from the constraints. Subsequently, this step is iteratively repeated until the solution is found or the manifold is fully explored (Allgower and Georg, 2012; S. LaValle, 1998). Consequently, efforts have been made to integrate kinematic constraints within the classical configuration space in motion planning (Kingston et al., 2018), where the constraint function is usually defined as a Riemannian manifold embedded in the configuration space (Spivak, 1999). Several approaches have been proposed to locate such implicit manifolds, including relaxation with increased tolerance for the constraint function (Bonilla et al., 2017), projecting onto the manifold's surface (Dennis Jr and Schnabel, 1996), and sampling in the tangent space to generate valid motions (Kim et al., 2016). Additionally, continuation-based methods have been introduced to explore the entire implicit manifold (Henderson, 2002; Jaillet and Porta, 2012), and also used to find the optimal trajectory of a manifold (Safta et al., 2022). Although these methods are effective for geometrically constrained motion problems, they typically lack a detailed description of contact interactions.

Incorporating physical laws into ordinary differential equations (ODEs) has proven effective for real-world problems like heat transfer (Conti et al., 2023) and oscillation (Cross et al., 2024). In this article, we introduce an adaptive ODE with a quasi-static assumption to guide robotic exploration on an implicit constrained manifold which has been formalized in a recent framework (Campolo and Cardin, 2023a, 2023b). This ODE computes (i) the location of the nodes on the equilibrium manifold and (ii) the haptic distance using a *haptic metric*. The ODE can be terminated by a haptic obstacle, ensuring stability and continuity of the manipulation. With this ODE, we then propose a numerical continuation algorithm that explores the implicit manifold and discretizes it into a graph of equilibria. Manipulation planning is then recast as an optimal process on this finite graph.

We compute distances in our configuration space by considering both the robot and object states, forming a graph that connects nearby nodes. Similarly to the "loop closure" concept in SLAM (Arshad and Kim, 2021), manipulation tasks like valve turning and screwing, which involve circular or helical motions, require precise classification. We validate our algorithm with an inverted pendulum example, demonstrating that the pendulum's internal space is topologically equivalent to  $S^1$  rather than  $\mathbb{R}^1$ .

The rest of the article is organized as follows. Our configuration space and haptic metric are detailed in Section 2. Our methodology, including the extension via ODEs terminated by the haptic obstacle, and the comprehensive algorithm for the continuation approach, is elaborated in Section 3. The manipulation of the inverted pendulum, including the contact model and potential expression, is discussed in Section 4. This section also showcases the algorithm's application in exploring and trimming the control space, exploring one branch of the manifold, and connecting it into a circle. Finally, conclusions are presented in Section 5.

## 2. Quasi-static manipulation and numerical continuation

In this section, we first review the quasi-static framework (Campolo and Cardin, 2023a, 2023b), including the potential modeling and the haptic metric. Additionally, we recall the numerical continuation approach for Euclidean space (Henderson, 2002).

### 2.1. Quasi-static mechanical manipulation system

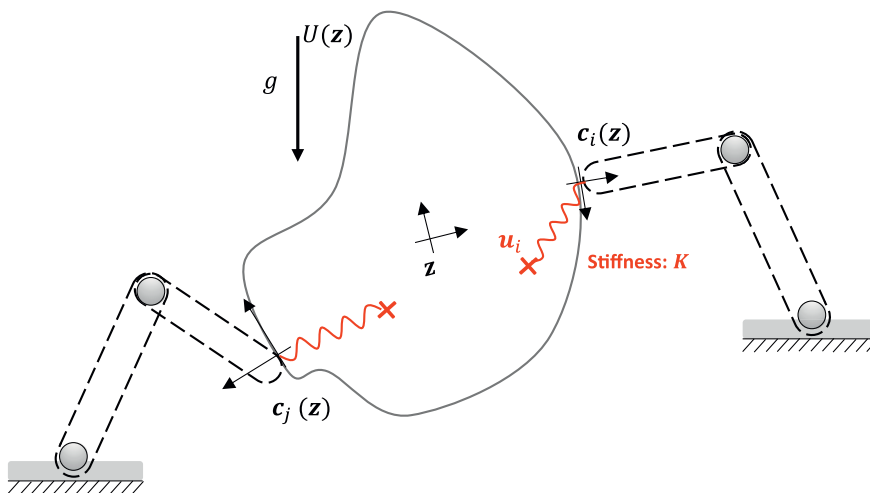
As illustrated in Figure 1, starting with (Campolo and Cardin, 2023a, 2023b), the key steps for analyzing a manipulation task involve (i) splitting of degrees of freedom into nondirectly controllable variables (e.g., environment)  $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^N$  and directly controllable variables  $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^K$ . (ii) defining a *manipulation potential*  $W(\mathbf{z}, \mathbf{u})$ :

$$W: \mathbb{R}^N \times \mathbb{R}^K \rightarrow \mathbb{R},$$

$$(\mathbf{z}, \mathbf{u}) \mapsto \sum_i W_i^{el}(\mathbf{z}, \mathbf{u}) + U(\mathbf{z}) \quad (2.1)$$

Assuming there are  $i$  contact points in total, each contact point is symbolized as  $\mathbf{c}_i(\mathbf{z})$ . The body  $\mathcal{B}$ , which denotes the internal states  $\mathbf{z}$ , decides these contact points  $\mathbf{c}_i(\mathbf{z})$ . The control inputs  $\mathbf{u}_i$ , which connect to the contact points via a virtual spring, capture these interactions. In general, the dimensionality of the control space is often smaller than the dimensionality of the uncontrollable state, since  $K$  depends on the number of actuators.

The configuration of the system is then determined solely by its manipulation potential  $W(\mathbf{z}, \mathbf{u})$ . The potential of objects  $U(\mathbf{z})$  includes the contact energy between objects and the gravity potential. The elastic



**Figure 1.** Robots manipulate an object under equilibrium. The object  $\mathbf{z}$  is an indirectly controllable variable. Robots contact with the object at contact points  $\mathbf{c}_i(\mathbf{z})$ , and control inputs  $\mathbf{u}_i$  denote the desired position of the robot. Virtual springs represent impedance control.

potential  $W_i^{el}(\mathbf{z}, \mathbf{u})$  captures the robot–environment interaction in terms of non-linear yet smooth interactions elastic potentials, which is crucial, and corresponds to a regularisation of otherwise non-smooth mechanical interaction. In particular, mechanical contact is modeled using nonlinear springs that exhibit high stiffness when the robot end-effector penetrates objects in the environment and highly flexible otherwise, with smooth transitions in-between. Each spring energy can be defined as

$$W_i^{el}(\mathbf{z}, \mathbf{u}) = \frac{1}{2} \|\mathbf{u}_i - \mathbf{c}_i(\mathbf{z})\|_{\mathbf{K}_i}^2 \tag{2.2}$$

where  $\|\mathbf{a}\|_A^2 := \mathbf{a}^T \mathbf{A} \mathbf{a}$  denotes the *Mahalanobis distance*, and  $\mathbf{K}_i$  denotes the stiffness matrix for the virtual spring. Once a robot-environment potential function  $W(\mathbf{z}, \mathbf{u})$  is defined for a problem, as outlined in (Campolo and Cardin, 2023a, 2023b), quasi-static manipulation can be seen as an optimization problem taking place on the so-called *Equilibrium Manifold* (EM)  $\mathcal{M}^{eq}$ , defined as the subspace space of mechanical equilibria with equilibrium condition as

$$\nabla_z W(\mathbf{z}_m^*, \mathbf{u}) = \mathbf{0} \in \mathbb{R}^N \tag{2.3}$$

We define  $\nabla_q W \equiv [\partial_{q_1} W, \dots, \partial_{q_a} W]^T$ , where the *nabla* (column) operator is defined as  $\nabla_q = [\partial_{q_1}, \dots, \partial_{q_a}]^T$ . Meanwhile, define the shorthand notation  $\nabla_z^2 \equiv (\nabla_z \nabla_z^T) = \nabla_z \nabla_z^T$  for Hessians and mixed-derivative operators. Meanwhile,  $\mathbf{z}_m^*$  represents the potentially multiple solutions, with  $m \geq 1$  indicating the multiplicity of equilibria. In other words, for the same control input of the system, the uncontrollable state can have multiple situations. As such, the EM is defined when  $\mathbf{u}$  is seen as a parameter, that is,

$$\mathcal{M}^{eq} \{(\mathbf{z}, \mathbf{u}) \in \mathcal{Z} \times \mathcal{U} \mid \nabla_z W(\mathbf{z}, \mathbf{u}) = \mathbf{0}\} \tag{2.4}$$

is a smooth embedded submanifold in the ambient space  $\mathcal{Z} \times \mathcal{U}$ . The state transitions are purely controlled by the robotic agent  $\mathbf{u}$ . Meanwhile, A point is strictly stable when its Hessian is positive definite, i.e.,  $\nabla_{zz}^2 W|_* > 0$ . Meanwhile, following (Campolo and Cardin, 2023a, 2023b), the haptic distance  $S$  between any two points  $\mathbf{u}_0$  to  $\mathbf{u}_1$  on control space is defined as,

$$S[\mathbf{u}(c)] = \int_{\mathbf{u}_0}^{\mathbf{u}_1} \sqrt{\frac{d\mathbf{u}^T}{dc} \mathbf{G}_m^2(\mathbf{u}) \frac{d\mathbf{u}}{dc}} dc, \tag{2.5}$$

where  $S[\mathbf{u}(c)]$  represents the robot’s cost when following a control policy  $\mathbf{u}(c)$ . The greater the force required by the robot during manipulation, the larger the value of  $S$ , and  $\mathbf{G}_m(\mathbf{u})$  is the control Hessian (Campolo and Cardin, 2023a, 2023b) defined at equilibria  $(\mathbf{z}_m^*, \mathbf{u})$ :

$$\mathbf{G}_m(\mathbf{u}) \equiv \nabla_{uu}^2 W - \nabla_{uz}^2 W (\nabla_{zz}^2 W)^{-1} \nabla_{zu}^2 W \tag{2.6}$$

### 2.2. Numerical continuation approach

In differential geometry, it is established that a manifold can be characterized by a set of local parametrizations known as charts, which are collectively organized within an atlas (Do Carmo, 2016). (Henderson, 2002) introduced a continuation method to explore a manifold which is implicitly defined by a series of overlapping neighborhoods. The concept involves covering a flat manifold,  $\mathbb{R}^k$ , with spherical balls, or projecting from the tangent space onto a general manifold. Therefore, each chart is defined as  $C_i(\mathbf{u}_i)$ , including a center  $\mathbf{u}_i$  and a polygon  $P_i^m$  as an approximation, where  $m$  denotes the number of steps.

The relationships between these balls can be categorized based on their spatial interactions: they may be external to each other, one ball may be completely inside another, or they may intersect. These relationships are determined by comparing the distance between the centers of two balls with their radius  $R$ . To compute the intersection of a sphere that lies within a convex polyhedron defined by the intersection of half spaces, Henderson employs a finite convex polyhedral,  $C_{R_i}(\mathbf{u}_i)$ , to cover each spherical ball. Consequently, the boundary of this collection of balls is formed by the union of all the individual ball boundaries and the polygons. Efficient exploration of the entire space hinges on expansion along these boundaries. A practical approach involves extending from the exterior vertices of the polygons. Here, “exterior” refers to vertices where the distance from the vertex to the center exceeds the radius (Eq. 2.7).

To manage this process, a waiting list for spheres is maintained, which helps in prioritizing expansion points.

$$B_{\text{list}}^m = \{i | \exists \mathbf{v} \in \text{vertex}(P_i^m) \text{ s.t. } |\mathbf{v}| > R_i\} \tag{2.7}$$

From this exterior vertex, a new ball is generated along with its corresponding polygon. Subsequently, a trimming function is applied to reduce the overlapping volume between nearby polyhedra.

$$P_i^m = \begin{cases} \emptyset & \text{if for any } j \in J_i^m, |\mathbf{u}_j - \mathbf{u}_i| < R_j - R_i, \\ C_{R_i}(\mathbf{u}_i) \cap_{j \in J_i^m} (HS_{ij}) & \text{otherwise} \end{cases} \tag{2.8a}$$

$$HS_{ij} = \left\{ \mathbf{u} | (\mathbf{u} - \mathbf{u}_i) \cdot (\mathbf{u}_j - \mathbf{u}_i) \leq \frac{1}{2} (R_i^2 - R_j^2 + |\mathbf{u}_j - \mathbf{u}_i|^2) \right\} \tag{2.8b}$$

This entails identifying the intersection or cross set between a half-space ( $HS_{ij}$ ) and the polyhedron. This step of the framework is called POLYTRIM. For any two polyhedra, we trim them by Algorithm 1.

---

**Algorithm 1** Trim two polyhedra

---

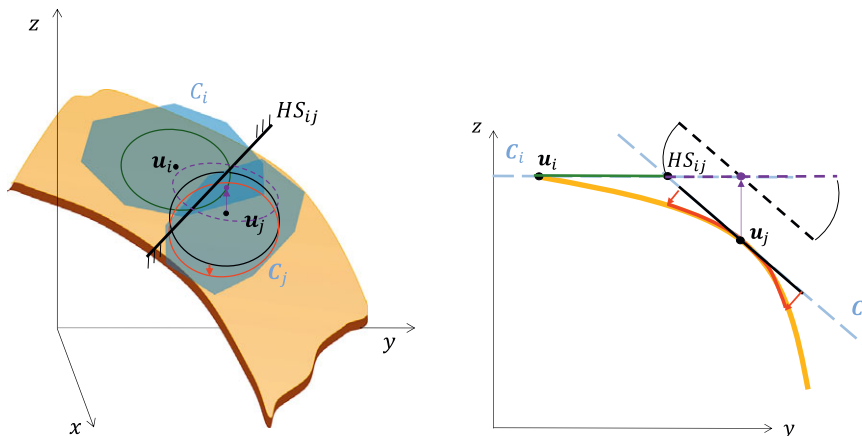
```

1: procedure POLYTRIM( $C_i, C_j$ )
2:    $\mathbf{u}_i, R_i \leftarrow C_i, \mathbf{u}_j, R_j \leftarrow C_j$ 
3:   if  $|\mathbf{u}_j - \mathbf{u}_i| < R_j - R_i$  then
4:      $P_i^m \leftarrow \emptyset$ 
5:   else
6:      $P_i^m \leftarrow C_{R_i}(\mathbf{u}_i) \cap HS_{ij}$ , where  $HS_{ij}$  is derived from Eq. 2.8.
7:    $C_i.P_i^m \leftarrow P_i^m$ 
8:   return  $C_i$ 

```

---

Subsequently, the classical numerical continuation approach to move on the constrained manifold is depicted in Figure 2. Each chart  $C_i$  is approximated by polygon covering circle with centers  $\mathbf{u}_i$  and  $\mathbf{u}_j$ . During the



(a) Two near charts in 3D space on a 2D manifold.

(b) Observe from in  $y - z$  plane to illustrate the projection.

**Figure 2.** Chart  $C_i$  and  $C_j$  are approximated by polygon covering circle with centers  $\mathbf{u}_i$  and  $\mathbf{u}_j$ . Each chart  $C_i$  is expressed in the tangent space at its center  $\mathbf{u}_i$ . Black circle in the tangent space at  $\mathbf{u}_j$  is projected to the manifold as the red circle, then onto the tangent space at  $\mathbf{u}_i$  as the pink circle. The half space  $HS_{ij}$  pass through the intersection points between the green circle and the pink circle.

(Henderson, 2002) process, two projection steps are performed on the manifold. One is to project the black circle in the tangent space at  $\mathbf{u}_i$  to manifold as the red circle. The other is to project the black circle onto the tangent space at  $\mathbf{u}_i$  as the pink circle. The former is used to cover the manifold, the latter is aimed at finding the half space  $HS_{ij}$  to execute the trimming process. The projection step requires iterations to find the solution on the manifold. Moreover, each step during trim and growth requires a projection operation, and this projection-based growth is widely used in other constrained motion planning techniques (Jaillet and Porta, 2012; Kingston et al., 2019).

### 3. Haptic continuation exploration

The necessity of the projection operation in the methods mentioned above is due to the inherent property of the classical numerical continuation approach, where the new node is far from the parent node on the manifold. As such, to remain on the manifold, the new node needs the iterative projection operation.

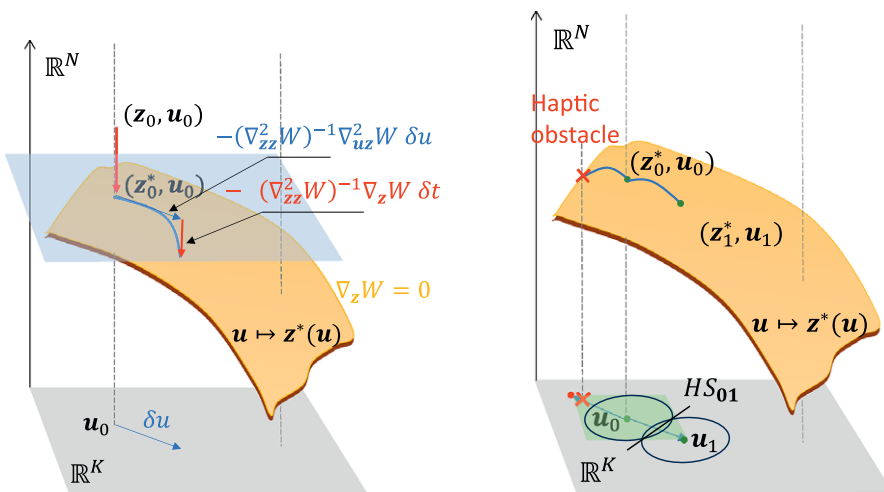
Different from the classical approach, we employ an adaptive ODE to simultaneously compute nodes and geodesics as we explore through our configuration space without using the iterative projection. The exploration strategy, derived from (Henderson, 2002), is adapted within our control space  $\mathcal{U} \in \mathbb{R}^K$ . This method is integrated into our framework, with vertex properties as both frontier and haptic obstacles, which are specialized for manipulation tasks. The process solely requires a potential function for the system, from this point, all subsequent steps are derived.

#### 3.1. Exploring via adaptive ODE

The Equilibrium Manifold (EM)  $\mathcal{M}^{eq}$  is therefore *implicitly defined* and, given an initial solution  $(\mathbf{z}_0, \mathbf{u}_0) \in \mathcal{M}^{eq}$ , a first-order approximation on the equilibrium manifold is given as,

$$\delta \mathbf{z} = -(\nabla_{zz}^2 W)^{-1} \nabla_{uz}^2 W \delta \mathbf{u} \tag{3.1}$$

Eq. 3.1 is depicted as a blue arrow in Figure 3(a), which illustrates the linear relationship with an infinitesimal change between  $\mathbf{z}$  and  $\mathbf{u}$ . The calculation of an implicit manifold from a given set of (nonlinear) equations invariably relies on standard iterative methods such as Newton–Raphson, as in the case of (Henderson, 2002; Jaillet and Porta, 2012). Recently, it was shown that the classical iterative Newton–Raphson method can be transformed into an adaptive ODE (Schneebeil and Wihler, 2011).



(a) Blue arrow denotes  $\mathbf{z}$  linear approximation as the variation of  $\mathbf{u}$ , red arrow represents Newton-Raphson ‘infinitesimal’ adjustment.

(b) Using continuation approach in control space: ODE can reach the vertex or be terminated by haptic obstacle.

**Figure 3.** The evolution of physics-informed adaptive ODE exploring manifold.

Therefore, based on quasi-static assumptions, we propose evaluating the evolution  $t \rightarrow \mathbf{z}(t) \in \mathbb{R}^N$  whenever the control parameters  $t \rightarrow \mathbf{u}(t) \in \mathbb{R}^K$  are (quasi-statically) varied by (numerically) solving a set of adaptive ODE. The Newton-Raphson “infinitesimal” adjustments, when  $u = \text{const}$  gives rise to the out-of-equilibrium dynamics, depicted as a red arrow in Figure 3(a).

$$\delta \mathbf{z} = -\alpha (\nabla_{\mathbf{z}\mathbf{z}}^2 W)^{-1} \nabla_{\mathbf{z}} W \delta t, \tag{3.2}$$

where  $\alpha$  represents step size in the ODE (Schneebeli and Wihler, 2011), which influences the convergence of the ODE. If  $\alpha$  is too small, the ODE may fail to converge within the given time. Conversely, if  $\alpha$  is too large, the ODE might jump to a different branch of the EM.

Clearly, if out of equilibrium (i.e.  $W_{\mathbf{z}} = 0$ ), for a fixed  $\mathbf{u}$  and by virtue of the positive-definiteness of the Hessian  $W_{\mathbf{z}\mathbf{z}} > 0$ , this term drives the variables toward the equilibrium manifold  $\mathcal{M}^{eq}$ .

From a given initial solution  $(\mathbf{z}_0, \mathbf{u}_0) \in \mathcal{M}^{eq}$ , a new nearby solution  $(\mathbf{z}_1, \mathbf{u}_1)$  for a new parameter  $u_1$  at constant rate  $\dot{\mathbf{u}} = (\mathbf{u}_1 - \mathbf{u}_0)/T$ , can be evaluated by numerical integration of the adaptive ODE. Hence, the adaptive ODE becomes

$$\frac{d}{dt} \begin{bmatrix} \mathbf{z} \\ \mathbf{u} \\ \phi \end{bmatrix} = \begin{bmatrix} -(\nabla_{\mathbf{z}\mathbf{z}}^2 W)^{-1} \nabla_{\mathbf{z}\mathbf{z}}^2 W \dot{\mathbf{u}} - \alpha (\nabla_{\mathbf{z}\mathbf{z}}^2 W)^{-1} \nabla_{\mathbf{z}} W \\ (\mathbf{u}_1 - \mathbf{u}_0)/T \\ \sqrt{\dot{\mathbf{u}}^T \mathbf{G}_m^2(\mathbf{u}) \dot{\mathbf{u}}} \end{bmatrix}, \begin{bmatrix} \mathbf{z}(0) \\ \mathbf{u}(0) \\ \phi(0) \end{bmatrix} = \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{u}_0 \\ 0 \end{bmatrix} \tag{3.3}$$

Meanwhile, we introduce the concept of a haptic obstacle,

$$\det(\nabla_{\mathbf{z}\mathbf{z}}^2 W(\mathbf{z}^*(\mathbf{u}), \mathbf{u})) \geq \lambda > 0 \tag{3.4}$$

where  $\lambda$  is a threshold. The haptic constraint (Eq. 3.4) is imposed because the inverse of the Hessian is required, so  $-(\nabla_{\mathbf{z}\mathbf{z}}^2 W)^{-1}$  should be sufficiently small to avoid significant changes of  $\mathbf{z}$ . Meanwhile,  $\lambda > 0$  indicates the Hessian is positive definite, suggesting the node is stable on the equilibrium manifold. The stable points are also referred to as nondegenerate critical points (Poston and Stewart, 2014). The adaptive ODE with haptic obstacles always starts from a stable point and avoids exploring unstable critical points, as the surrounding region of an unstable critical point is inherently unstable. To prevent numerical instabilities, the ODE can be terminated by haptic obstacle, ensuring stability and continuity in the manipulation process while avoiding singularities.

### 3.2. Numerical continuation in control space

In our framework, we can only control the robot  $\mathbf{u}$ , the directly controllable variables, rather than the entire implicit EM. Hence, we explore the control space via the numerical continuation approach. Each chart  $\mathcal{C}_i$  is represented as a node in a graph. These charts include information about  $\mathcal{C}_i \equiv (\mathbf{u}_i, \mathbf{z}_i, R_i, P_i^m)$ , where  $i$  denotes the index of this chart, and  $m$  represents the number of exploration steps. It is important to note that only the polyhedral can be revised during exploration, as  $P_i^m$  is determined by both  $i$  and  $m$ .

Similarly to Henderson’s method, our waiting list stores the polyhedra to be explored. Each polyhedron contains several vertices and is associated with two lists of properties: *isExteriorVertex* and *isDeadVertex*, abbreviated as ExVTX and DdVTX. The ExVTX label is initially set to TRUE for all vertices, but is reassessed after every trimming session. This label indicates whether a vertex is a frontier that needs to be explored, defined as

$$\text{ExVTX}(k) = \{ \text{TRUE if } \|v_k\| > R_i, \text{ else FALSE, } \forall v_k \in \text{vertex}(P_i^m) \} \tag{3.5}$$

where  $k$  is the index of vertex,  $i$  denotes the index of polygon, ExVTX includes all the vertex of a polygon.

The second label, DdVTX, is initialized for all the vertex as FALSE, and it is evaluated during the growth phase to determine if a vertex leads to a dead end; this status is maintained even after trimming.

When selecting a polyhedral from the waiting list, we choose a vertex where ExVTX is true and DdVTX is false. Unlike Henderson’s method, which directly grows a new sphere, our strategy involves extending

from the center  $\mathbf{u}_i$  to the vertex  $v_i$  using the adaptive ODE from Eq. 3.3. This ensures that all steps in the manipulation are quasi-static, enhancing the precision and stability of the robot’s movement. For each ODE, representing the growth step from the center to the vertex over a specified time range, termination can occur due to a haptic obstacle or upon reaching the target vertex. The label DdVtx is updated based on the outcome  $\mathbf{u}(T)$  from ODE (Eq. 3.3), defined as:

$$\text{DDVTX}(k) = \{ \text{FALSE if } \mathbf{u}(T) = v_k \text{ else TRUE } \mid v_k \in \text{vertex}(P_i^m) \} \tag{3.6}$$

This also implies that we can only update one vertex  $i$  for the list DdVtx within each growth. Meanwhile, the edge weight between two nodes is evaluated as  $\mathcal{E}_{ij} = S_{ij} = \phi(T)$ . As such, the continuation exploration can be visualized in Figure 3(b), where each growth begins from the center of a polyhedron and targets a frontier vertex. The entire ODE (Eq. 3.3) stays on EM, shown as the curve on EM. There is sometimes a risk that the ODE may leave the EM, resulting in loss of stability. In such cases, the corresponding vertex will be labeled as DdVtx. Conversely, the ODE may be completed successfully, indicating that the growth step is finished.

---

**Algorithm 2** Extend a node in the Graph

---

**Input:** A chart  $C_i$  on the waiting list.

**Output:** Updated graph  $\mathcal{G}$ .

```

1: procedure EXTEND_GRAPH( $C_i$ )
2:    $\mathbf{u}_0, z_0, P_i^m \leftarrow C_i$ 
3:    $\mathbf{u}_1 \leftarrow v_k, v_k \in P_i^m, C_i.\text{ExVtx}(k) = \text{TRUE}$  and  $C_i.\text{DdVtx}(k) = \text{FALSE}$ 
4:    $[z(T), \mathbf{u}(T), \phi(T)] \leftarrow \text{ODE}(z_0, u_0, u_1, T)$ 
5:   if  $\mathbf{u}(T) = u_1$  then
6:      $\mathcal{G}, m \leftarrow \text{ADD\_NODE}(\mathbf{u}(T), z(T))$ 
7:      $B_{\text{list}}^m = B_{\text{list}}^m \cup \{m\}$ 
8:     for  $j \in 1$  to  $(m - 1)$  do
9:        $C_m, C_j \leftarrow \mathcal{G}$ 
10:      if NEARBY_CHARTS( $C_m, C_j$ ) then
11:         $z_m, u_m \leftarrow C_m, u_j \leftarrow C_j$ 
12:         $\phi \leftarrow \text{ODE}(z_m, u_m, u_j, T)$ 
13:         $\mathcal{E} \leftarrow \text{ADDEDGE}(m, j, \phi)$ 
14:         $C_m \leftarrow \text{POLYTRIM}(C_m, C_j)$ 
15:         $C_j \leftarrow \text{POLYTRIM}(C_j, C_m)$ 
16:        if  $\sum_k C_m.\text{ExVtx}(k) = 0$  or  $\sum_k C_j.\text{ExVtx}(k) = 0$  then
17:          Remove  $m$  or  $j$  from  $B_{\text{list}}^m$ 
18:      else
19:         $C_i.\text{DdVtx}(k) = \text{TRUE}$ 
20:      return  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 

```

---

Hence, the algorithm for EXTEND\_GRAPH becomes Algorithm 2, the function starts by inputting a chart  $C_i$ , where we extract the centre point  $(\mathbf{u}_0, z_0)$  and the polyhedron  $P_i^m$  (line 2). We also select a vertex labeled as ExVtx is True and DdVtx is False, indicating that this vertex is a frontier and is achievable, having not encountered the haptic obstacle (line 3). Subsequently, we execute the ODE (Eq. 3.3) (line 4) to determine whether it can integrate to this vertex (line 5). If it is true, indicating that the ODE was not terminated by a haptic obstacle, we add a new node to our graph (line 6), and also append the index  $m$  of this new chart to the waiting list (line 7). After adding a new node, we enter the trimming phase. We loop through all existing charts (lines 8 and 9), and assess whether they are sufficiently close (line 10). If they are, we compute the ODE from the new chart  $C_m$  to the nearby existing charts (lines 11 and 12). A new edge is then added to the graph based on the geodesic  $\phi$  (line 13). Subsequently, we trim both polyhedrons



separately with the POLYTRIM function via Eq. 2.8 (lines 14 and 15). The final step involves checking the new polyhedron. If this polyhedron no longer has any frontier vertices, it is removed from the waiting list. Conversely, if the growth is terminated by a haptic obstacle, the corresponding DDVTX label is set to True.

The charts are only trimmed when they are near. Different from Henderson's method, we define the distance function as

$$\text{dist}(C_i, C_j) = \sqrt{\|u_i - u_j\|^2 + l(\text{mod}(z_i - z_j, 2\pi))^2} \quad (3.7)$$

where  $l$  relates to the size of the object to make same dimension for translation and rotation (Campolo et al., 2013). This formula reflects that the rotational periodicity of real-world components (e.g., motors) is  $2\pi$ . Hence, we define the function NEARBY\_CHARTS as in Algorithm 3 where  $R_i + R_j$  defines the distance in the control space,  $R_z$  defines the distance in the state space. As the difference between  $z_i$  and  $z_j$  equals  $2\pi$ , two nodes are still near each other.

---

#### Algorithm 3 Decide nearby charts

---

```

1: procedure NEARBY_CHARTS( $C_i, C_j$ )
2:    $u_i, R_i \leftarrow C_i, u_j, R_j \leftarrow C_j$ 
3:   if  $\text{dist}(C_i, C_j) < R_i + R_j + R_z$  (Eq. 3.7) then
4:     return TRUE
5:   else
6:     return FALSE

```

---

### 3.3. Overall algorithm

We start our exploration by placing a node on the EM and defining a radius  $R_i$  for the corresponding sphere to generate a polyhedron. We maintain a waiting list  $B_{\text{list}}^m$  of these polyhedra (line 1). In each iteration of our process, as long as the waiting list is not empty (line 2), we select the first index from the waiting list (line 3) and identify its charts (line 4), if this chart does not have an unexplored frontier vertex, which means  $C_i.\text{EXVTX} \subseteq C_i.\text{DDVTX}$ , we remove it from the waiting list. Otherwise, we start to grow once using Algorithm 2. Finally, we sort the sequence of polyhedra in the waiting list to determine the exploration strategy, such as whether to explore the most distant polyhedron first or start from the nearest to the origin.

---

#### Algorithm 4 Haptic graph exploration

---

**Input:**  $(z_{\text{start}}, u_{\text{start}}) \in \mathcal{M}^{eq}$  the starting point on the equilibrium manifold,  $\epsilon$  the geodesic size and  $N$  the maximum number of attempts.

**Output:** A search graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ .

```

1:  $\mathcal{V} \leftarrow C_{\text{start}} = \{z_{\text{start}}, u_{\text{start}}, R, P_{\text{start}}^m, \text{EXVTX} = \text{TRUE}, \text{DDVTX} = \text{FALSE}\}; \mathcal{E} \leftarrow \emptyset, B_{\text{list}}^m = [1]$ 
2: while  $B_{\text{list}}^m \neq \emptyset$  do
3:    $i \leftarrow B_{\text{list}}^m[1]$ 
4:    $C_i \leftarrow \mathcal{G}(i)$ 
5:   if  $C_i.\text{EXVTX} \subseteq C_i.\text{DDVTX}$  then
6:     Remove  $i$  from  $B_{\text{list}}^m$ 
7:   else
8:      $\mathcal{G} \leftarrow \text{EXTEND\_GRAPH}(C_i)$ 
9:      $B_{\text{list}}^m \leftarrow \text{SHIFT}(B_{\text{list}}^m)$ 
10: return  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 

```

---

#### 4. Simulation validation: manipulation of an inverted pendulum

We validate our algorithm through the inverted pendulum task. (Campolo and Cardin, 2023a, 2023b) has previously explored the EM of an elastically driven inverted pendulum, aiming to stabilize it in an arbitrary position. We apply the energy-based numerical continuation approach to this example, where the manipulation of the pendulum is similar to a “staircase” within the configuration space. We start with modeling the inverted pendulum task and show the performance of our method.

##### 4.1. Inverted pendulum model

In a 2D plane, the system comprises a pendulum modeled as a superquadric with length  $L_0$  and mass  $m$ . A virtual robot agent is connected to the pendulum via a *nonlinear* spring. The pendulum is hinged at one end to the origin of the world, and a body frame is attached to the center of mass (CoM). As depicted in Figure 4, the uncontrollable state variable of the system is the rotation angle  $z = \theta \in S^1$ , and the control vector is the position of the robot  $\mathbf{u} = [u_x, u_y]^T \in \mathbb{R}^2$ .

Hence, the total manipulation potential  $W(\mathbf{z}, \mathbf{u})$  consists of contact energy between the robot and the pendulum, and gravity potential.

$$\begin{aligned}
 W(\mathbf{z}, \mathbf{u}) &= W_{\text{grav}}(\mathbf{z}) + W_{\text{ctrl}}(\mathbf{z}, \mathbf{u}), \\
 &= \frac{1}{2}mgL_0 \sin \theta + \frac{1}{2}k \left( (u_x - L_0 \cos \theta)^2 + (u_y - L_0 \sin \theta)^2 \right),
 \end{aligned}
 \tag{4.1}$$

where  $k$  is the nonlinear spring stiffness, capturing the interaction between the robot and the pendulum.

$$k(d) = k_{\min} + \frac{1 - \tanh(d/d_0)}{2} k_{\max}.
 \tag{4.2}$$

The variable  $l$  serves as a signed distance for mechanical contact. When  $l < 0$ , it indicates that the robot  $\mathbf{u}$  has penetrated the pendulum, signifying that contact has occurred. This results in a large contact force due to the high stiffness parameter  $k_{\max}$ . Conversely, when there is no contact between the robot and the pendulum ( $l > 0$ ), the resultant force should be zero, indicated by a significantly lower stiffness ( $k_{\max} \gg k_{\min}$ ). The parameter  $l_0$  is utilized to facilitate a smooth transition between contact and no-contact states, to guarantee a smooth and differentiable manifold.

##### 4.2. Exploring and trimming in control space

We first show how the numerical continuation approach expands in the control space and how POLYTRIM operates. We set  $R_i$  for each polyhedron to be the same as the red circles in Figure 5(a). We observe that a convex polyhedron (represented as a square in 2D space) contains a sphere (represented as a circle in 2D space). Then, POLYTRIM identifies the half-space between two nearby polyhedra. The polyhedron is trimmed until the square fits within the circle, indicating no more frontier vertices.

The final result is shown in Figure 5(b), we plot the entire “staircase” branch of the inverted pendulum task in the configuration space. Each polyhedron is depicted as a blue square since control space  $\mathcal{U} \in \mathbb{R}^2$ . The polyhedra in control space is moved down to better indicate the idea of control space with respect to

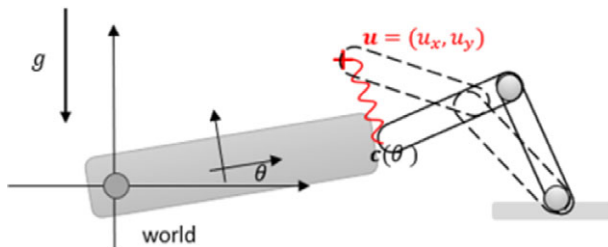
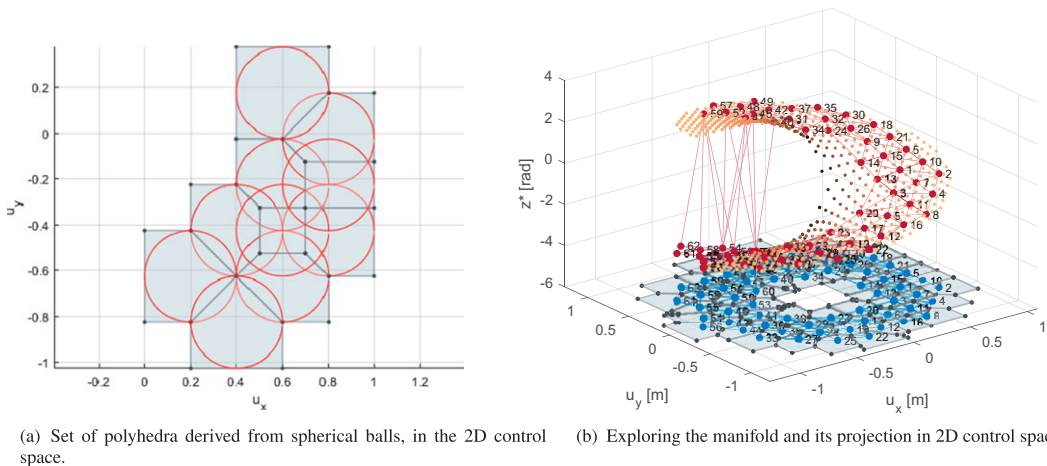


Figure 4. Inverted pendulum task, where the contact interaction is captured by non-linear stiffness.



**Figure 5.** The implicit manifold  $\mathcal{M}^{eq}$  is discretized via a Graph having as nodes a finite set of equilibria in  $\mathcal{M}^{eq}$  interconnected weighted edges defined via the  $G^2$  metric. Manipulation planning can therefore be recast into an optimal process on finite Graphs.

configuration space. The implicit manifold is illustrated with orange dots in the background, while the red dots represent the nodes in the graph. We observe that all the red nodes remain on the EM, which validates the accuracy of our adaptive ODE.

We identify two boundaries of red nodes. The outer circle, representing a haptic obstacle, indicates instability and helps avoid loss of singularity. The inner circle, also a haptic obstacle, suggests that the robot is exerting excessive force on the pendulum, leading to instability. Furthermore, the connectivity in the implicit manifold indicates that rotating the pendulum full circle is equivalent to returning to the starting point, thus forming a loop closure for manipulation. We demonstrate that this task inherently exhibits a circular nature, as opposed to a helical one, and confirm that our  $\mathcal{Z}$  space belongs to  $S^1$  instead of  $\mathbb{R}^1$ . Hence, our algorithm enables the robot to recognize that the inverted pendulum task is a circular motion, which is crucial for selecting the appropriate manipulation strategy. For instance, understanding that circular tasks (e.g., valve turning) require no translational motion, while helical tasks (e.g., bottle opening) necessitate it, ensures the robot can recognize these differences. Meanwhile, by applying Topological Data Analysis (TDA) (Joharinad and Jost, 2023; Nathaniel Saul, 2019), the graph has a genus of 1, which supports our results. This means the object has one hole in its structure from its topological perspective.

**Remark 1:** In the real-world scenario, observed  $z^*$  could be noisy but the control  $u$  is always accurate. Therefore, this means the initial guess of our ODE is noisy. However, with the help of the attractor properties of Newton–Raphson and the description of the embedded manifold within our ambient space ensure that our ODE still functions correctly, ultimately attracting the node back to the manifold.

**Remark 2:** It is worth noting that our work fundamentally differs from classical inverted pendulum studies (Irfan et al., 2024), which primarily focus on control, aiming to stabilize the pendulum without considering contact dynamics. In contrast, our method incorporates contact interactions between the robot and the pendulum, with the goal of exploring the entire manipulation manifold while emphasizing its connectivity and topological properties. From a broader perspective, our work aligns more closely with motion planning approaches for pendulum-like objects such as door handles (Shaikh-Mohammed et al., 2023).

**Remark 3:** Conventional motion planning methods typically default to directly controlled variables, which may not fully capture the interaction between robot commands and manipulated objects. In contrast, our approach considers both indirectly and directly controllable variables. Additionally,

constrained motion planning approaches rely on task-specific constraint functions. For example, in the case of manipulation of the inverted pendulum, a task-based constraint function (Kingston et al., 2018) on  $\mathbf{u}$  is required to generate a feasible control path to invert the pendulum, while our framework does not require any task-specific constraint function.

## 5. Conclusion

In this work, an energy-based numerical continuation approach is applied to a novel quasi-static manipulation framework. This approach integrates the classical numerical continuation method with an adaptive ODE, which is terminated by a haptic obstacle to ensure the stability of the exploration and guarantee that all nodes remain on the EM. Additionally, a distance measurement is employed to identify neighboring nodes in the graph, aligning with our framework that accounts for both robot and object states. The effectiveness of the proposed method has been validated through an inverted pendulum example. The results demonstrate that our method can successfully discover the branch of the EM while being terminated by a haptic obstacle. We also observe the connectivity of the graph, confirming the inherent property of the pendulum task as a circular motion. In future work, we plan to extend our framework to real-world scenarios, and the implicit manifold can be reconstructed from observed data, accounting for potential noise and practical challenges. Additionally, we aim to implement an adaptive mechanism for the exploration radius to improve computational efficiency.

**Data availability statement.** The code is available at [https://github.com/lillyyang/Henderson\\_DCE.git](https://github.com/lillyyang/Henderson_DCE.git)

**Acknowledgments.** This research is supported by the National Research Foundation, Singapore, under the NRF Medium Sized Centre scheme (CARTIN). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

We would like to express our sincere gratitude to Yipeng Zhang and Professor Kelin Xia for their valuable advice on Topological Data Analysis.

**Author contribution.** Conceptualization, L.Y., D.C. and F.C.; Methodology, L.Y., D.C. and F.C.; Software, L.Y. and D.C.; Validation, L.Y.; Formal analysis, L.Y.; Investigation, L.Y.; Resources, D.C.; Data curation, L.Y.; Writing—original draft, L.Y.; Writing—review and editing, L.Y., H.T.N., C.L. and D.C.; Visualization, L.Y.; Supervision, D.C. and C.L.; Project administration, D.C.; Funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

**Funding statement.** This research was supported by grants from the National Research Foundation, Singapore, under the NRF Medium Sized Centre scheme (CARTIN)

**Competing interest.** None

**Ethical standard.** The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

## References

- Allgower EL and Georg K (2012) *Numerical continuation methods: An introduction*, vol. 13. Springer Science & Business Media.
- Arshad S and Kim G-W (2021) Role of deep learning in loop closure detection for visual and lidar slam: A survey. *Sensors* 21(4), 1243.
- Bonilla M, Pallottino L and Bicchi A (2017) Noninteracting constrained motion planning and control for robot manipulators. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE. 4038–4043.
- Campolo D and Cardin F (2023a) A basic geometric framework for quasi-static mechanical manipulation. arXiv preprint arXiv:2307.10489.
- Campolo D and Cardin F (2023b) Quasi-static mechanical manipulation as an optimal process. In *2023 62nd IEEE Conference on Decision and Control (CDC)*. 4753–4758.
- Campolo D, Widjaja F, Xu H, Ang WT, and Burdet E (2013) Analysis of accuracy in pointing with redundant hand-held tools: A geometric approach to the uncontrolled manifold method. *PLoS Computational Biology* 9(4), e1002978.
- Conti ZX, Choudhary R and Magri L (2023) A physics-based domain adaptation framework for modeling and forecasting building energy systems. *Data-Centric Engineering* 4, e10.
- Cross EJ, Rogers TJ, Pitchforth DJ, Gibson SJ, Zhang S and Jones MR (2024) A spectrum of physics-informed gaussian processes for regression in engineering. *Data-Centric Engineering* 5, e8.
- Dennis Jr JE and Schnabel RB (1996) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM.

- Do Carmo MP** (2016) *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications.
- Henderson ME** (2002) Multiple parameter continuation: Computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos* 12(03), 451–476.
- Irfan S, Zhao L, Ullah S, Mehmood A and Fasih Uddin Butt M** (2024) Control strategies for inverted pendulum: A comparative analysis of linear, nonlinear, and artificial intelligence approaches. *Plos One*, 19(3), e0298093.
- Jaillet L and Porta JM** (2012) Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE Transactions on Robotics* 29(1), 105–117.
- Joharinad P and Jost J** (2023) *Mathematical Principles of Topological and Geometric Data Analysis*. Springer.
- Kim B, Um TT, Suh C and Park FC** (2016) Tangent bundle RRT: A randomized algorithm for constrained motion planning. *Robotica* 34(1), 202–225.
- Kingston Z Moll M and Kavraki LE** (2018) Sampling-based methods for motion planning with constraints. *Annual Review of Control, Robotics, and Autonomous Systems* 1, 159–185.
- Kingston Z, Moll M and Kavraki LE** (2019) Exploring implicit spaces for constrained sampling-based planning. *The International Journal of Robotics Research* 38(10–11), 1151–1178.
- LaValle S** (1998) Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*.
- LaValle SM** (2006) *Planning Algorithms*. Cambridge University Press.
- Mason MT** (2001) *Mechanics of Robotic Manipulation*. MIT Press.
- Nathaniel Saul CT** (2019) Scikit-tda: Topological data analysis for python. <https://doi.org/10.5281/zenodo.2533369>
- Poston T and Stewart I** (2014) *Catastrophe Theory and Its Applications*. Courier Corporation.
- Safta C, Ghanem RG, Grant MJ, Sparapany M and Najm HN** (2022) Trajectory design via unsupervised probabilistic learning on optimal manifolds. *Data-Centric Engineering* 3, e26.
- Salem A and Karayiannidis Y** (2020) Robotic assembly of rounded parts with and without threads. *IEEE Robotics and Automation Letters* 5(2), 2467–2474.
- Schneebeli HR and Wihler TP** (2011) The Newton–Raphson method and adaptive ODE solvers. *Fractals* 19(01), 87–99.
- Shaikh-Mohammed J, Alharbi Y and Alqahtani A** (2023) Door-opening technologies: Search for affordable assistive technology. *Technologies* 11(6), 177.
- Spivak MD** (1999) A comprehensive introduction to differential geometry. *A Comprehensive Introduction to Differential Geometry*, v. 1. <https://books.google.com.sg/books?id=ahSWQgAACAAJ>. ISBN: 9780914098706, lccn: 2002283516, Publish or Perish, Incorporated.
- Suomalainen M, Karayiannidis Y and Kyrki V** (2022) A survey of robot manipulation in contact. *Robotics and Autonomous Systems* 156, 104224.
- Turlapati SH and Campolo D** (2022) Towards haptic-based dual-arm manipulation. *Sensors* 23(1), 376.
- Whitney DE, et al** (1982) Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement, and Control* 104(1), 65–77.
- Yang L, Ariffin MZ, Lou B, Lv C and Campolo D** (2023) A planning framework for robotic insertion tasks via hydroelastic contact model. *Machines* 11(7), 741.