

1

Introduction to Continuum Robots and Theoretical Foundations

1.1 Introduction

Continuum robots have received increasing attention in recent decades. Their compliance and dexterity make them valuable in various important areas such as medical treatment [1], industry [2], agriculture [3], and aerospace [4]. Fully exploiting their capabilities necessitates the design of effective, efficient, and robust control systems for continuum robots, which remains challenging due to their continuous nature.

Numerous attempts have been made to establish analytical models for continuum robots and develop control strategies based on these models [5–8]. Representative approaches for modeling continuum robots include the piecewise constant-curvature (PCC) model [9], the finite element method [10], the Cosserat rod model [11], and the Euler–Bernoulli beam theory. Based on these established models, control of continuum robots is typically performed using closed-form analytical inverse kinematics solutions [12], solutions computed from the robot Jacobian [13], or solutions obtained through iterative searching. Although model-based control of continuum robots has made significant progress, analytical modeling of continuum robots remains challenging because their configurations are rarely sensed completely and are easily affected by environmental interactions [14], which hinders the application of model-based approaches.

Due to the complexity and difficulty of analytical modeling, numerous data-driven control methods have been proposed to estimate or learn the kinematics, inverse kinematics, or differential kinematics of continuum robots [15–18]. For instance, Thuruthel et al. [19] reported a neural-network-based method to directly learn the inverse kinematics of continuum robots. Kasaei et al. [20] developed a data-efficient neural network to approximate the differential kinematic model and estimate the robot Jacobian for control purposes.

Table 1.1 Notations.

Symbol	Definition
t	Time
\mathbf{u}	Actuation state vector of the continuum robot
\mathbf{p}	End-effector position of the continuum robot
\mathbf{p}_d	Reference position of the continuum robot
\mathcal{F}	Matrix of activation functions
\mathbf{f}	Kinematic mapping function of the continuum robot
J	Jacobian matrix of the continuum robot
$\ \cdot\ $	2-norm of a vector or Frobenius norm of a matrix
$(\cdot)^\dagger$	Pseudoinverse operation
$(\cdot)^T$	Transpose operation

Neurodynamic methods have been employed in the control of continuum robots in recent years due to their advantages such as high efficiency and potential for parallel computation. Existing neurodynamic approaches mainly include gradient neurodynamics (GND) and zeroing neurodynamics (ZND) [21]. The GND method seeks the optimal solution to a control problem by defining a scalar error function and computing its gradient with respect to the task variables. In contrast, the ZND method leverages the time derivative of time-varying variables and specific design formulations to solve control problems. Both methods have been widely applied to various numerical computation problems, such as matrix equations [22], quadratic programming [23], and matrix inversion [24, 25], and have demonstrated promising performance in the control of continuum robots.

This book primarily focuses on how neurodynamic methods can be used to achieve control of continuum robots. To provide readers with a preliminary understanding of the content of this book, this chapter first introduces a basic modeling method for continuum robots based on the PCC model. Then, the theoretical foundations of neurodynamic methods are briefly presented. Some of the notations used in this book are listed in Table 1.1.

1.2 Modeling Continuum Robots Using PCC

For simplicity, this part introduces the kinematic modeling of a multi-segment continuum robot based on the PCC assumption [26], using a series of constant-curvature arcs to approximate the backbone of continuum robots. Based on the PCC assumption, the relationship between actuation space and task space is

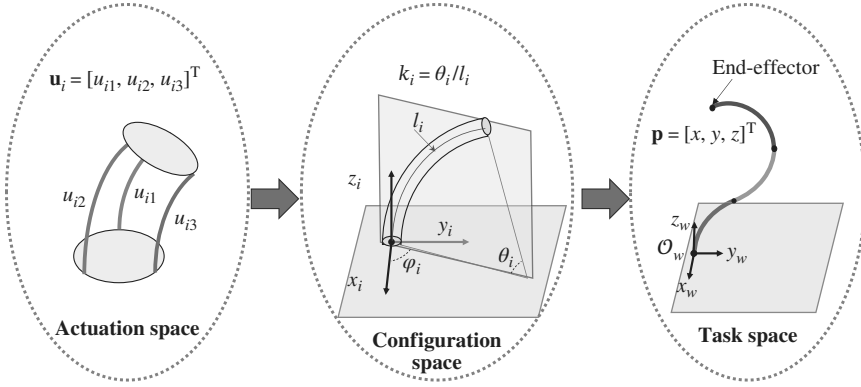


Figure 1.1 Kinematic modeling convention of continuum robots under PCC assumption involves three spaces and two mappings.

decomposed into two parts, as illustrated in Figure 1.1. This involves two mappings: the mapping from actuation space to configuration space, and the mapping from configuration space to task space. The first mapping uses the actuation state vector \mathbf{u} to compute the configuration state (φ, κ, l) , where arc parameters φ , κ , and l represent the rotational deflection angle, curvature, and arc length of the arc that denotes a segment of the continuum robot, respectively. In actuation space, $\mathbf{u}_i = [u_{i1}, u_{i2}, u_{i3}]^T$ represents the actuation state of a single segment, where i denotes the i th segment of the multi-segment continuum robot, and u_{ik} describes the state of the k th actuator in the i th segment. In configuration space, a triplet $(\varphi_i, \kappa_i, l_i)$ is defined to represent the arc parameters of the i th segment.

The mapping from actuation space to configuration space is robot-specific, depending on the actuation strategy and generating unique arc parameters in configuration space. This chapter considers a multi-segment continuum robot. The relationship between actuator state and arc parameters of the i th segment is formulated as

$$\varphi_i = \arctan\left(\frac{\sqrt{3}(u_{i2} + u_{i3} - 2u_{i1})}{3(u_{i2} - u_{i3})}\right), \quad (1.1)$$

$$\kappa_i = \frac{2\sqrt{u_{i1}^2 + u_{i2}^2 + u_{i3}^2 - u_{i1}u_{i2} - u_{i1}u_{i3} - u_{i2}u_{i3}}}{d(u_{i1} + u_{i2} + u_{i3})}, \quad (1.2)$$

$$l_i = \frac{u_{i1} + u_{i2} + u_{i3}}{3}, \quad (1.3)$$

where d is the distance between the central backbone of a segment and the actuator.

The mapping from configuration space to task space is robot-independent, which means that it can be applied to all robots that can be approximated as PCC arcs. With the known configuration state, the end-effector position \mathbf{p} can be computed and represented by $\mathbf{p} = [x, y, z]^T$ in task space. The coordinate frames \mathcal{O}_i is attached to the base of the i th segment and the world coordinate frame \mathcal{O}_w is displayed. The transformation between coordinate frames \mathcal{O}_i and \mathcal{O}_{i+1} can be expressed by a transformation matrix $T_i^{i+1} \in SE(3)$. Therefore, the end-effector position $\mathbf{p}_e \in \mathbb{R}^3$ in the coordinate frame \mathcal{O}_{n+1} is transformed to the position $\mathbf{p} \in \mathbb{R}^3$ with respect to the world coordinate frame \mathcal{O}_w based on the following formulation:

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \prod_{i=1}^n T_i^{i+1} \begin{bmatrix} \mathbf{p}_e \\ 1 \end{bmatrix}, \quad (1.4)$$

where n is the number of segments and

$$T_i^{i+1} = \begin{bmatrix} C_{\varphi_i}^2 (C_{\kappa_i l_i} - 1) + 1 & S_{\varphi_i} C_{\varphi_i} (C_{\kappa_i l_i} - 1) & C_{\varphi_i} S_{\kappa_i l_i} & \frac{C_{\varphi_i} (1 - C_{\kappa_i l_i})}{\kappa_i} \\ S_{\varphi_i} C_{\varphi_i} (C_{\kappa_i l_i} - 1) & C_{\varphi_i}^2 (1 - C_{\kappa_i l_i}) + C_{\kappa_i l_i} & S_{\varphi_i} S_{\kappa_i l_i} & \frac{S_{\varphi_i} (1 - C_{\kappa_i l_i})}{\kappa_i} \\ -C_{\varphi_i} S_{\kappa_i l_i} & -S_{\varphi_i} S_{\kappa_i l_i} & C_{\kappa_i l_i} & \frac{S_{\kappa_i l_i}}{\kappa_i} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.5)$$

with $S_x = \sin(x)$ and $C_x = \cos(x)$.

1.3 Neurodynamic Models

To begin with, the basic design procedures regarding the ZND model and the GND model are introduced.

1.3.1 ZND Model

The foundation of ZND is the error-monitoring function [27], which differs from conventional norm-based scalar-valued error/energy functions. Specifically, the error-monitoring function is indefinite and can be vector- or matrix-valued, as well as real- or complex-valued, depending on the requirements. As an efficient method for solving time-variant problems, ZND involves the following design procedures.

First, an error-monitoring function $\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t)$, where t denotes time, \mathbf{x}_d denotes the desired system state and \mathbf{x} denotes the actual system state, must be defined to monitor the process of solving time-variant problems. For brevity, the argument t is dropped hereinafter if no confusion is caused. Then, to ensure the function converges to zero exponentially and globally, its first-order time derivative $\dot{\mathbf{e}}$ is determined using the design formula of ZND as follows:

$$\dot{\mathbf{e}} = \frac{d\mathbf{e}}{dt} = -\lambda f(\mathbf{e}), \quad (1.6)$$

where $\lambda \in \mathbb{R}^+$ is a convergence parameter and $f(\cdot)$ is a monotonically increasing odd activation function. Finally, by incorporating the error-monitoring function into the design formula of ZND, the dynamic equation of a ZND model for solving time-variant problems is established.

1.3.2 GND Model

As a typical form of neurodynamics, GND has been extensively studied and applied to solve various problems [28, 29]. For a system where \mathbf{x} denotes the state of the system and \mathbf{x}_d denotes the desired state, a norm-based scalar energy function $\epsilon = \|\mathbf{x}_d - \mathbf{x}\|$ related to the GND model is defined based on the specific task, where $\|\cdot\|$ is the 2-norm of a vector or Frobenius norm of a matrix [30]. Then, to evolve along the descent direction (i.e., the negative gradient direction) of the energy function, the GND design formula is as follows:

$$\dot{\mathbf{v}} = -\lambda \frac{\partial \epsilon}{\partial \mathbf{v}}, \quad (1.7)$$

where \mathbf{v} is the independent variable of the system, $\dot{\mathbf{v}}$ denotes the first-order time derivative of \mathbf{v} , and $\lambda \in \mathbb{R}^+$ is a convergence parameter. Finally, by expanding the design formula (1.7) with the energy function ϵ , the dynamic equation of a GND model for solving time-variant problems is established.

1.4 Chapter Summary

Taking the PCC assumption as an example, this chapter has introduced a simple modeling methods for continuum robots. Besides, a concise overview of the theoretical foundations of neurodynamic methods, including ZND and GND, is presented to help readers develop a clear understanding of these approaches.

