A Learning-Based Inverse Kinematics Solver for Two-Segment Continuum Robot Models

Jiewen Lai and Henry K. Chu

***Abstract*— In this brief work, we present a learning-based method to solve the inverse kinematics (IK) of a two-segment continuum robot manipulator using a simplified model featured with variable-length virtual links. This approach allows us to accurately estimate the overall continuum robot gesture only by its distal end-effector's status. A multilayer perceptron (MLP) is implemented for the learning process.**

I. The Method

Based on the piecewise constant curvature assumption [1], a continuum robot manipulator can be simplified as an articulated robot with several revolute joints. The robot configuration of single segment is shown in Fig. 1. Consider the n-th segment of the continuum manipulator with a length of L_n, the configuration of this segment can be defined by δ_n and θ_n, where δ_n represents the angular deviation in terms of bending direction of the n-th segment w.r.t. the base frame of segment n, and θ_n denotes the extent of bending in such direction. The extent of bending is defined as the complementary angle of the tangential intersection of two ends of the circular arc. By doing so, the configuration of a continuum segment can be interpreted as a 2-DOF articulated robot with two virtual rigid links with equal length of $\frac{L_n^\dagger}{2}$, which is given by $\frac{L_n^\dagger}{2} = \frac{L_n}{\theta_n} \tan \frac{\theta_n}{2}$, and $\lim_{\theta_n \to 0} \frac{L_n^\dagger}{2} = \frac{L_n}{2}$.

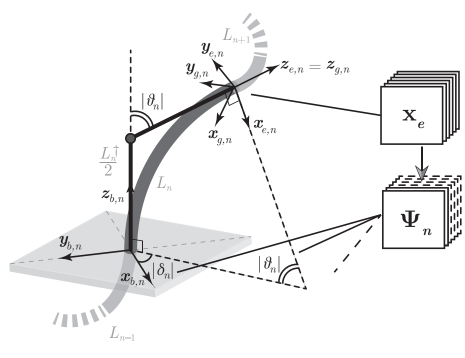

Fig. 1. Robot configuration at the n-th segment under the constant curvature approximation. Subscripts b, e, g represent the base frame, end-effector frame, and global frame, respectively.

Thus, the configuration and the end-effector status of an n-segment continuum robot can be described by

$$\mathbf{\Psi}_n = [\delta_1, \theta_1, ..., \delta_n, \theta_n]^\intercal \in \mathbb{R}^{2n} \tag{1}$$

$$\mathbf{x}_e = [x_e, y_e, z_e, \alpha_e, \beta_e, \gamma_e]^\intercal \in \mathbb{R}^6 \tag{2}$$

The authors are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong. e-mail: henry.chu@polyu.edu.hk

The task space, which comprises the position and orientation of the end-effector of the n-th segment, is represented by $\mathbf{x}_e = [x_e, y_e, z_e, \alpha_e, \beta_e, \gamma_e]^\intercal \in \mathbb{R}^6$, in which $\alpha_e, \beta_e, \gamma_e$ are the $\mathbf{ZYX}$ eulerian angles representing the tip orientation. The robot-independent mapping from the configuration space $\mathbf{\Psi}_n$ to task space $\mathbf{x}_i$ is given by $\mathbf{x}_i = f(\mathbf{\Psi}_n)$, where $f(\cdot)$ signifies the forward kinematics, and $\mathbf{x}_i$ represents the i-th Cartesian coordinate. Vise versa, the inverse kinematics is given by $\mathbf{\Psi}_n = f^{-1}(\mathbf{x}_i)$.

Using D-H convention, the forward kinematics of the simplified robot model can be attained by multiplying homogeneous transformation matrices as follows:

$$\mathbf{T}_i^0 = \prod_{i=1}^{i} \begin{bmatrix} \mathbf{R}_i^{i-1} & \mathbf{p}_i^{i-1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{3}$$

where i is the number of D-H frames, $\mathbf{p}_i^{i-1} = [x_e, y_e, z_e]^\intercal$.

The proposed IK solver is trained to attain the nonlinear relationship between $\mathbf{x}_e$ and $\mathbf{\Psi}_n$ based on the forward kinematics using MLP, which consists of l layers. For layer l, assuming that the total output is $\mathbf{V}^{(l)}$, in which $\mathbf{c}_j^{(l)}$ and $\mathbf{v}_j^{(l)}$ denote the input and output of node j. The connection between layer l and layer $(l-1)$ relies on a weighting matrix $\mathbf{W}^{(l)}$. The weights between the k-th node of layer $(l-1)$ and the j-th node of layer l is given by $\mathbf{w}_{jk}^{(l)}$. The general expression of the output of layer l can be concluded as

$$\begin{cases} \mathbf{v}_j^{(l)} = \varphi\left(\mathbf{c}_j^{(l)}\right), \\ \mathbf{c}_j^{(l)} = \sum\limits_{l-1} \mathbf{w}_{jk}^{(l)} \mathbf{u}_k^{(l-1)} + \mathbf{b}_j^{(l)}, \\ \mathbf{V}^{(l)} = \varphi\left(\mathbf{c}^{(l)}\right) = \varphi\left(\mathbf{W}^{(l)}\mathbf{U}^{(l-1)} + \mathbf{B}^{(l)}\right). \end{cases} \tag{4}$$

Making use of the back-propagation algorithm, the network can solve the mapping from input data to output data. Levenberg-Marquardt (LM) optimization is employed in backpropagation for its fast convergence in non-linear fitting problems. The topology of MLP - the number of hidden layers N_H and the number of neurons N_N in each hidden layer - is determined based on the performance (mean squared error). Without loss of generality, a two-segment continuum manipulator is used to evaluate the method and verify the learning-based IK solver via simulation.

References

[1] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.