Modeling Kinematic Uncertainty of Tendon-Driven Continuum Robots via Mixture Density Networks

Jordan Thompson, Brian Y. Cho, Daniel S. Brown, and Alan Kuntz

***Abstract*— Tendon-driven continuum robot kinematic models are frequently computationally expensive, inaccurate due to unmodeled effects, or both. In particular, unmodeled effects produce uncertainties that arise during the robot's operation that lead to variability in the resulting geometry. We propose a novel solution to these issues through the development of a Gaussian mixture kinematic model. We train a mixture density network to output a Gaussian mixture model representation of the robot geometry given the current tendon displacements. This model computes a probability distribution that is more representative of the true distribution of geometries at a given configuration than a model that outputs a single geometry, while also reducing the computation time. We demonstrate one use of this model through a trajectory optimization method that explicitly reasons about the workspace uncertainty to minimize the probability of collision.**

I. Introduction

Much work is being done currently to develop tendon-driven continuum robots for minimally-invasive surgery [1]–[3]. However, application of these robots are bottlenecked in part by an inability to accurately and efficiently compute their kinematics. Stochastic and unmodeled effects on the kinematics of tendon-driven continuum robots may lead to these robots being unsafe to use during surgical tasks as small deviations from the expected geometry could cause unwanted collisions with fragile structures within the body. Further, as the complexity of the kinematic models grow with the introduction of these effects, so too does the required computation time. There is a need for a model that is capable of reasoning over these kinematic uncertainties while also minimizing the necessary computation time. We propose a novel approach to kinematic modeling for tendon-driven surgical manipulators through learned Gaussian mixture models, enabling direct estimation of learned kinematic uncertainty.

Current state-of-the-art physics-based models attempt to explicitly model the different forces acting on the robot [4]–[8]. While in theory, these models should accurately compute the geometry of the robot, they tend to either be too restrictive in their assumptions or computationally too slow for real-time surgical applications. As the complexity of the physics being modeled increases, these models increase in accuracy. However, they suffer in computation time and in the presence of additional unmodeled or stochastic effects. Data-driven approaches attempt to solve this problem by implicitly learning how to model the forces that impact the resulting robot geometry [9], [10]. These approaches decrease the computation time; however, they do not currently possess the ability to handle the inherent uncertainties associated with the robot's kinematics. Any unmodeled effects on the kinematics lead to geometric uncertainties that impact the safe use of these robots. In this paper, we provide results that demonstrate how these unmodeled effects manifest for tendon-driven continuum robots. The issues that arise from the current state-of-the-art models make using them in surgical applications potentially unsafe and/or too slow to perform real-time computations.

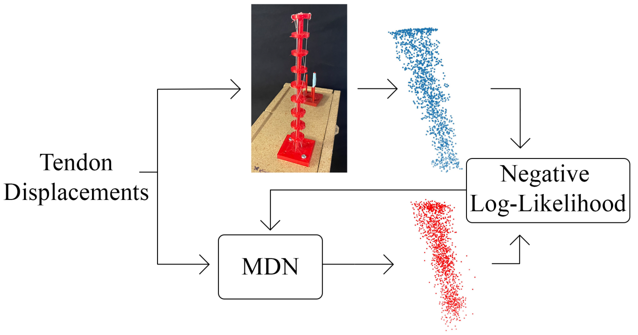

Fig. 1: A Mixture Density Network (MDN) is trained to produce a Gaussian mixture model representation of the robot's geometry given the tendon displacements. The negative log-likelihood is computed between the MDN's output and ground-truth point clouds collected from the physical robot. The MDN is then trained to minimize the negative log-likelihood.

We present a solution to modeling the kinematics of tendon-driven continuum robots that can reason implicitly over modeling uncertainty. We train a mixture density network that outputs Gaussian mixture models of the robot's geometry at a given configuration of tendon displacements. Contrary to standard kinematic models that explicitly output the robot's geometry, our method enables reasoning over the probability that the robot will occupy different regions of its workspace. We compare the accuracy of our model across a range of different numbers of mixture components in the model outputs, and we demonstrate a decrease in computation time when compared to a current state-of-the-art Cosserat rod model.

We apply our model to an optimization-based motion planner that explicitly minimizes the probability of collision with obstacles along a nominal trajectory. We perform Bayesian

The authors are with the Robotics Center and the Kahlert School of Computing at the University of Utah, Salt Lake City, UT 84112, USA; email: {jordan.thompson, brian.cho, daniel.brown, alan.kuntz}@utah.edu.

This material is based upon work supported in part by the National Science Foundation under grant number 2133027. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

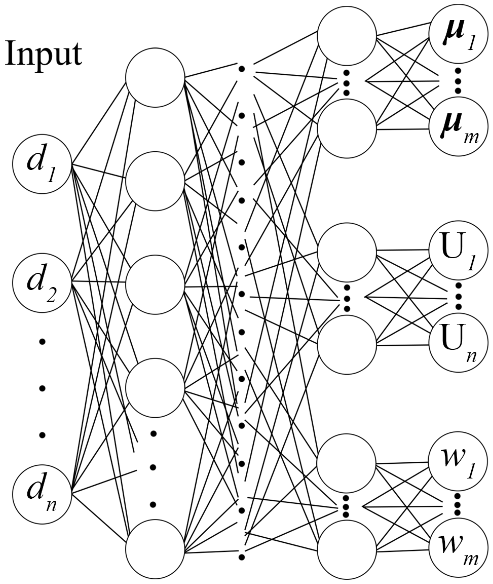

Fig. 2: An example mixture density network architecture that takes the list of tendon displacements $d = (d_1, d_2, ..., d_n)$ as input and outputs $\boldsymbol{\mu}_i$, U_i, and w_i independently for each component of the resulting Gaussian mixture model.

optimization with respect to the probability that the robot will collide with the environment. We estimate this probability by integrating the outputs of our model over the obstacles in the environment. We show that using our model enables a reduction in the probability of collision within a simulated chest cavity environment.

II. Related Work

A. Physics-Based Models

Tendon driven continuum robots are subject to many uncertainties, nonlinearities, and disturbances that negatively impact modeling accuracy. Much work has been done to handle these issues through improvements to physics-based kinematic models. Constant Curvature (CC) is one such model that assumes the robot consists of a finite set of curved links where each link can be defined by a set of arc parameters [4]. This assumption greatly simplifies the kinematics of the robot at the cost of accuracy. Pseudo-rigid body models (PRBM) have also been used to model continuum robot kinematics by assuming the robot consists of a set of rigid links connected by a torsional spring [5]. PRBM models provide higher accuracy than CC models at the cost of increased computation time.

As opposed to CC, Variable Curvature (VC) methods such as Cosserat rod models are used to more exactly represent the backbone structure of the robot [6]. The Cosserat rod model assumes the backbone to be a 1-dimensional rod and models the bending, stretching, twisting, and shearing forces applied to the robot. The result is a set of nonlinear ordinary differential equations for the kinematics and a set of nonlinear partial differential equations for the dynamics [7]. While VC models provide a more complete representation of continuum robot geometry than CC models, VC models are generally computationally more expensive. Recent work has provided significant computation time reductions for kinematic modeling and motion planning using the Cosserat rod model [8].

Rao, et al., perform a comparison of the different physics-based models and provide a set of guidelines for determining which model to use based on the structure of the robot assuming parallel routed tendons [11]. They conclude that when the structure of the robot contains a small number of separator disks holding the tendons, PRBM provides the best accuracy with respect to its computation time; however, as the number of disks increases, it becomes more useful to use VC models. We compare our work in this paper to the state-of-the-art Cosserat rod model described in [8].

B. Data-Driven Models

Data-driven models have also been proposed to model both the kinematics and dynamics for continuum manipulators. Nonlinear auto-regressive models with exogeneous inputs have been shown to perform well in predicting the end effector position for continuum robots [9]. While this model is able to implicitly learn how to model the forces on the robot, it does not reason over the uncertainties of continuum robots' geometries.

Reinhart and Steil compare several data-driven and hybrid modeling approaches for the forward-kinematics of soft robots [10]. They concluded that a non-linear Extreme Learning Machine (ELM) achieved the best result for minimizing the error with respect to the end effector position.

Work has also been done to learn the full geometry for concentric tube robots as a parametric curve [12]. Their model is capable of achieving high accuracy for concentric tube robots but does not allow for reasoning over the geometric uncertainties associated with continuum robots.

Others propose to use Gaussian Process Regression (GPR) for the forward kinematics of tendon-driven continuum robots [13]. They show a low position and orientation error post-training; however, this method does not allow for reconstruction of the full robot geometry. Gaussian Mixture Regression (GMR) has also been used for the inverse kinematics [14]. They find that Gaussian mixtures do not perform as well as other methods for computing the inverse kinematics of tendon-driven continuum robots. By contrast, we find in this paper that Gaussian mixtures perform well with respect to forward kinematics for representing the full geometry of the robot.

The current state of data-driven approaches to forward kinematic models primarily focuses on learning how to predict the position of the end effector of the robot while implicitly modeling the forces acting on the robot. The approach proposed in this paper extends these ideas to modeling the entire geometry of the robot as a Gaussian mixture model. Unlike prior work, this new model is capable of modeling the geometric uncertainties of tendon-driven continuum robots. In our experiments, we show that we are capable of explicitly

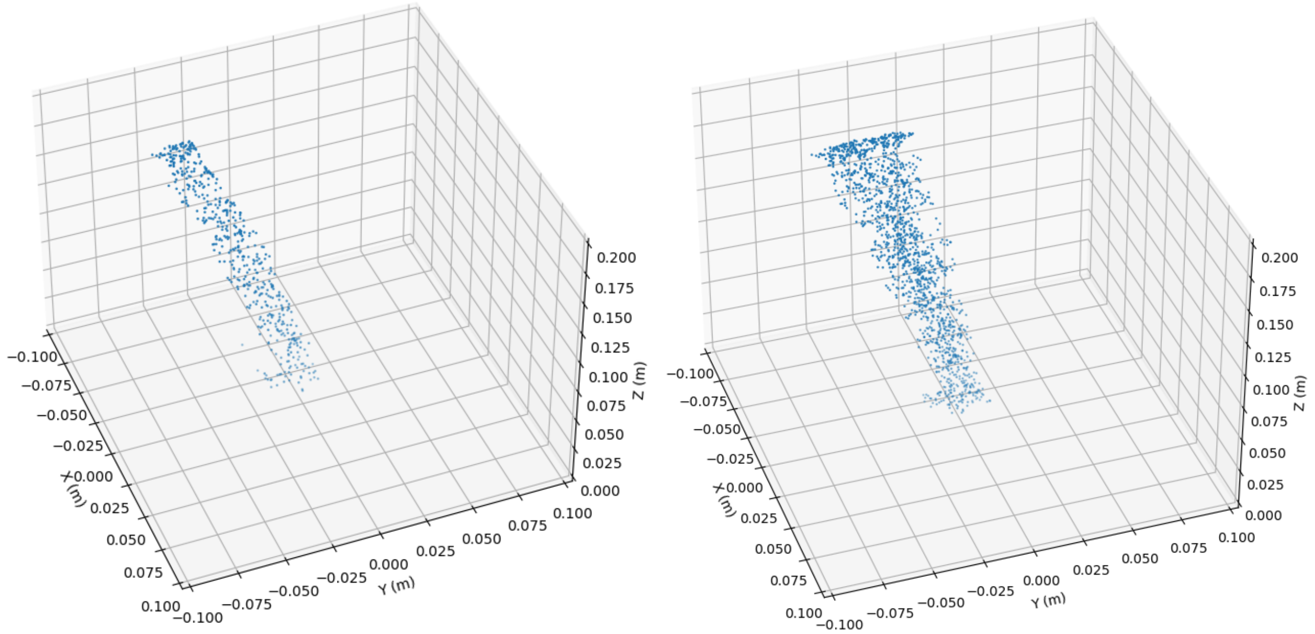

Fig. 3: An example configuration in the training data set. (Left) One particular instance of the configuration. (Right) The concatenated distribution of all point clouds gathered during data collection illustrating the inherent uncertainty associated with the geometry of the robot at a given configuration.

reasoning over these geometric uncertainties to improve the safety of tendon-driven continuum robot operation.

III. Kinematic Modeling

We propose a novel learned, data-driven approach to modeling the kinematics for tendon-driven continuum robots. As opposed to standard kinematic models that directly compute the position and orientation of each joint axis of the robot, our model computes a Gaussian mixture model (GMM) representation of the robot's geometry in the workspace. A GMM is a weighted combination of Gaussian probability distributions. The probability density function of a GMM for a random variable x is formulated as

$$\begin{aligned} p(x) &= \sum_{i=1}^{n} w_i p_i(x) \\ p_i(x) &= \frac{\exp\left(-\frac{1}{2}(x-\boldsymbol{\mu}_i)^T \Sigma_i^{-1} (x-\boldsymbol{\mu}_i)\right)}{\sqrt{(2\pi)^n |\Sigma_i|}} \end{aligned} \tag{1}$$

for n Gaussian components where $w_i \geq 0$ and $\sum_{i=1}^{n} w_i = 1$, and $\boldsymbol{\mu}_i$, Σ_i, and w_i are the mean, covariance matrix, and weight for component i respectively.

Given $d = (d_1, d_2, ..., d_m)$, where d_j is the displacement of tendon j, the model computes the GMM through a mixture density network (MDN). MDNs are a type of neural network composed of an initial feedforward fully connected network followed by separate feedforward fully connected networks to compute the means, covariance matrices, and weights for each component of the GMM. Figure 2 shows the architecture used to model the kinematics for the tendon-driven continuum surgical manipulator. For the purposes of modeling the workspace geometry of the robot, the MDN produces a three dimensional GMM for each configuration of tendon displacements.

We note that the model computes the matrices U_i as opposed to the covariance matrices Σ_i. Following the derivation from [15], U_i is used to reconstruct the precision matrix (inverse covariance matrix) for component i.

$$(\bar{\mathrm{U}}_i)_{jk} = \begin{cases} (\mathrm{U}_i)_{jk}, & \text{if } j \neq k \\ \exp(\mathrm{U}_i)_{jk}, & \text{otherwise} \end{cases} \tag{2}$$

where $\bar{\mathrm{U}}_i$ is the upper triangular matrix of the Cholesky decomposition of the precision matrix for component i, and indices j, k are the row and column for the corresponding matrix respectively. Thus, we have

$$\Sigma_i^{-1} = \bar{\mathrm{U}}_i^T \bar{\mathrm{U}}_i \tag{3}$$

This enables the outputs of the MDN to be unconstrained while maintaining that the covariance matrix for each component is positive-definite. The negative log-likelihood function can also be efficiently computed using U_i which reduces the required training time. This results from the ability to compute the log of the square root of the determinant of the precision matrix as the sum of the diagonal elements of U_i [15]. Following from equation 3,

$$|\Sigma_i^{-1}| = \left(\prod_{j=1}^{3} \operatorname{diag}(\bar{\mathrm{U}}_i)_j \right)^2. \tag{4}$$

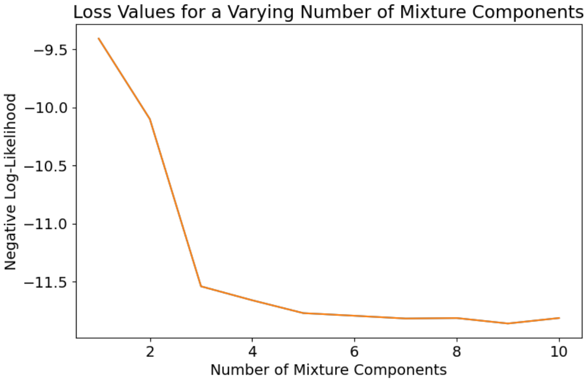

Fig. 4: Loss function values at different numbers of mixture components on the test data set. The results suggest that 5 mixture components are the minimum number of components necessary to sufficiently minimize the negative log-likelihood.

Taking the square root gives

$$|\Sigma_i^{-1}|^{\frac{1}{2}} = \prod_{j=1}^{3} \mathrm{diag}(\bar{\mathrm{U}}_i)_j. \tag{5}$$

We can then take the log to result in

$$\log |\Sigma_i^{-1}|^{\frac{1}{2}} = \sum_{j=1}^{3} \mathrm{diag}(\mathrm{U}_i)_j. \tag{6}$$

The MDN is trained to minimize the negative log-likelihood of ground-truth data collected from the robot. To gather the ground-truth distributions, we collect point clouds of the robot's geometry at various configurations using two RealSense cameras on opposing sides of the robot. We chose configurations for the data set by discretizing the robot's configuration space and collecting data for all configurations in this discretization. For each configuration in the data set, we randomly sample multiple starting configurations of the robot, then move the robot along a linear interpolation in the configuration space between the starting and desired configuration. A point cloud is then collected once the robot has arrived at the desired configuration. The concatenated point clouds across the different starting configurations gives a distribution of potential geometries the robot can have at the desired configuration. Figure 3 shows an example of one configuration in the training set. The right graph of figure 3 illustrates the variability that arises in the robot's resulting geometry despite being in the same configuration. The training and testing data sets contain 2,277, and 253 configurations respectively.

Given the training set of point clouds X and corresponding tendon displacement configurations Y, we produce the associated GMM for each configuration $y \in Y$ using the MDN.

$$\begin{aligned} MDN_\theta(y) = w_{i|\theta}, \boldsymbol{\mu}_{i|\theta}, \mathrm{U}_{i|\theta} \\ i \in \{1, 2, ..., n\} \end{aligned} \tag{7}$$

where θ are the network weights of the MDN.

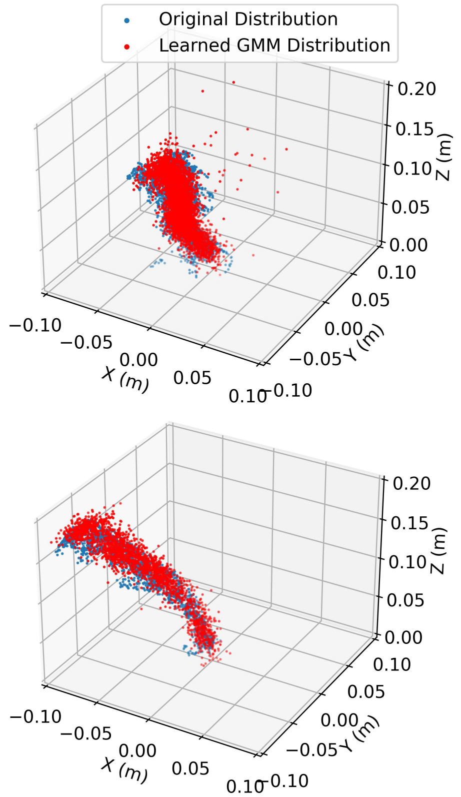

Fig. 5: An example set of outputs from the mixture density network after training with five mixture components compared with the ground-truth geometric distribution. The red points are sampled from the Gaussian mixture model output from the mixture density network. The blue points are the points collected from the robot during data collection. The sampled points demonstrate how the learned Gaussian mixture model is capable of representing the full distribution of geometries as a result of the kinematic uncertainty.

Using the computed GMMs, the negative log-likelihood of the training set point clouds is computed using the loss function derived in [15].

$$\begin{aligned} \mathcal{L}(\theta) &= \mathbb{E}_{x \in X, y \in Y}[-\log p_\theta(x|y)] \\ &\propto \mathbb{E}_{x \in X, y \in Y}\Bigg[-\log \sum_{i=1}^{n} w_{i|\theta} \cdot \exp\Bigg(\sum_{j=1}^{3} \mathrm{diag}(\mathrm{U}_{i|\theta})_j \\ &\quad - \tfrac{1}{2}\|\bar{\mathrm{U}}_{i|\theta}(x - \boldsymbol{\mu}_{i|\theta})\|_2^2\Bigg)\Bigg] \end{aligned}$$

The network parameters θ are optimized during training to minimize $\mathcal{L}(\theta)$. Figure 1 shows the training process for the MDN.

IV. Analysis

We trained our model on a tendon-driven continuum robot with nine separator disks. The robot is 0.2 m in length and is controlled via three parallel routed tendons and one helical tendon.

The model was trained with the number of mixture components (Gaussians) ranging from one to ten. Figure 4 shows the negative log-likelihood values on the test data set on models with different numbers of mixture components. We

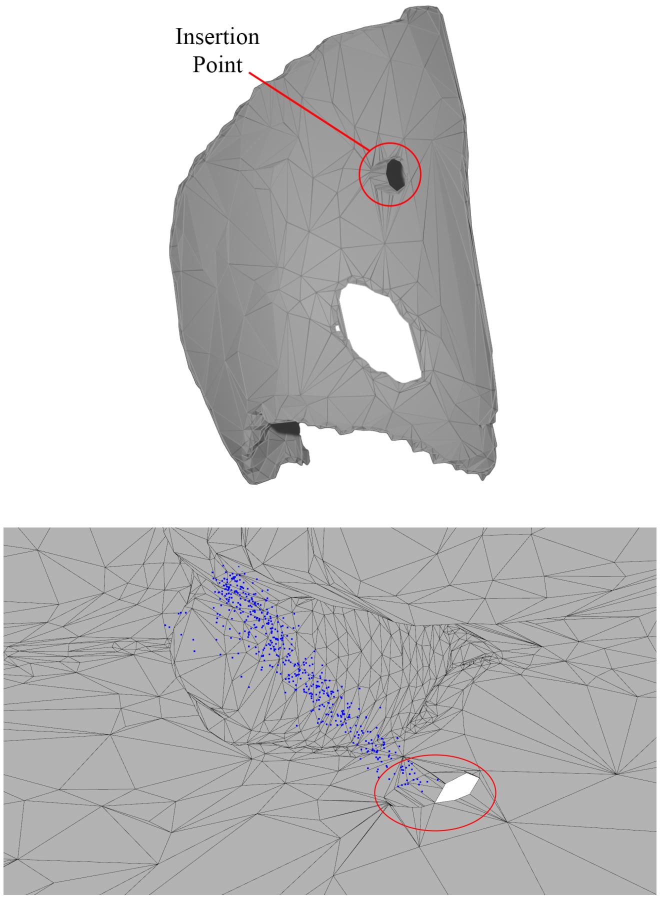

Fig. 6: The simulated chest cavity mesh environment used during motion planning, segmented from a patient undergoing a procedure to treat pleural effusion. (Top) The exterior of the volume. The red circle shows the insertion point for the robot. (Bottom) The interior of the volume. The red circle shows the insertion point for the robot. The blue points show a point cloud representation of the robot's shape inside the chest cavity in its home position (zero displacement for all tendons).

find that five mixture components minimize the negative log-likelihood the most while avoiding mode collapse. Mode collapse occurs when the MDN begins to output one or multiple component mixture weights arbitrarily close to zero. This implies that the model could have been trained with fewer mixture components while still achieving the same loss. Having a number of mixture components greater than five leads to this mode collapse phenomenon.

We note that the optimal number of mixture components is most likely robot specific. For different structures of the robot with different tendon routing patterns, it will be necessary to experimentally determine the appropriate number of mixture components for the GMM. We leverage the model with five components for the rest of the evaluation.

Figure 5 shows the model's output at two different randomly sampled configurations of the robot compared with the ground-truth geometric distribution at those configurations. The points sampled from the GMM accurately represent the expected geometry of the robot with a high density of points being sampled within the ground-truth distribution and a low density of points being sampled everywhere else.

We compare the computation time of the trained MDN with the Cosserat rod model presented in [8]. Timing our model over 10,000 randomly sampled configurations results in a mean computation time of 0.33 ms. The results in [8] show a mean computation time of 0.39 ms for the Cosserat rod model. This demonstrates a 15% reduction in computation time from the Cosserat rod model to our model.

We note that while standard comparisons of different kinematic models for tendon-driven continuum robots directly compare the accuracy of each model with respect to the predicted endpoint, we cannot provide this comparison for our model. Because our model learns distributions over geometries, there is no inherent notion of a predicted endpoint which poses a fundamental difference between the proposed model and standard kinematic models.

V. Planning with Gaussian Kinematics

One application for this kinematic model is safe motion planning. Using the learned kinematic model enables explicit reasoning over the probability of collision with obstacles as a result of the geometric uncertainty of the robot. Other motion planners that consider the probability of collision primarily consider the uncertainty as a result of motion and sensing uncertainty [16]–[18]. Our model enables an extension of these motion planners to reason explicitly over the geometric uncertainty associated with the kinematics of tendon-driven continuum surgical manipulators. Here we present a motion planner that uses our learned kinematic model to explicitly minimize the probability of collision for a given nominal trajectory of tendon displacements.

We first construct a nominal trajectory using the Probabilistic Roadmap (PRM) [19] motion planning algorithm, a widely used algorithm in robotics and motion planning to find feasible paths for robots operating in complex environments. It does so by constructing a roadmap of the given environment via random sampling in the robot's configuration space, enabling the robot to efficiently navigate to a goal while avoiding obstacles. However, any planning algorithm could be used to generate a nominal trajectory of configurations. In the generation of the initial roadmap (and subsequently the nominal trajectory), we check for collisions by testing that the means of each component of the GMM are not in collision with the obstacles in the environment. This condition enables us to later efficiently compute the probability of collision with the environment.

To improve the safety of the motion plan, we locally optimize the nominal trajectory itself with respect to the probability of collision as determined by our model. The probability of collision along the whole trajectory is minimized using Bayesian Optimization. To efficiently compute the probability of collision, we extend the method in [20] to reason over the probability of collision between a GMM and a given environment. To do so, for each component in the predicted GMM, we transform the environment's geometry such that the mean of the mixture component is at the origin with a covariance matrix equal to the identity matrix. We translate the environment by the negative of the mean and multiply by the Cholesky decomposition of the precision matrix. The component distribution in the new environment is the unit sphere centered at the origin. The new environment is constructed via

$$\hat{\mathrm{E}}_i = \bar{\mathrm{U}}_i(\mathrm{E} - \boldsymbol{\mu}_i) \tag{8}$$

where E is the set of all vertices in a mesh representation of the environment and $\hat{\mathrm{E}}_i$ is the set of all mesh vertices in the transformed environment for mixture component i.

A locally convex region of free space is then defined over this transformed environment using the method presented in [20] as a set of linear constraints $\boldsymbol{a}_{ij}^T\boldsymbol{p} \leq b_{ij}$, where $\boldsymbol{a}_{ij}, \boldsymbol{p} \in \mathbb{R}^3$, $b_{ij} \in \mathbb{R}$, i is the mixture component index, j is the constraint index, and $\boldsymbol{p}$ is a point in the robot's workspace. A conservative estimate for the probability that a configuration $\boldsymbol{x}$ is not in free space X_f is computed by

$$p(\boldsymbol{x} \notin \mathrm{X}_f) \leq \sum_{i=1}^{N}\left(w_i \sum_{j=1}^{K_i}(1 - \mathtt{cdf}(b_{ij}))\right) \tag{9}$$

for N mixture components and where K_i is the number of linear constraints for component i. The `cdf` function here refers to the cumulative distribution function for the standard normal Gaussian $\mathcal{N}[0,1]$. The probability of collision along a trajectory τ is estimated by taking the complement of the product of the probability that each state in the trajectory is not in collision.

$$p(\exists \boldsymbol{x} \in \tau : x \notin \mathrm{X}_f) \leq 1 - \prod_{\boldsymbol{x} \in \tau}(1 - p(\boldsymbol{x} \notin \mathrm{X}_f)) \tag{10}$$

Using Bayesian Optimization [21], the upper bound on $p(\exists \boldsymbol{x} \in \tau : \boldsymbol{x} \notin \mathrm{X}_f)$ is iteratively minimized for the given nominal trajectory. During each iteration of the optimization, we choose one intermediate configuration along the trajectory to optimize. New potential configurations are randomly sampled in a neighborhood about the chosen configuration. We evaluate each random sample using the probability of improvement acquisition function [22]. The probability of collision is then recomputed using the sample with the highest likelihood of reducing the probability of collision. If the sampled trajectory has a lower probability of collision than the previous best trajectory, the best trajectory is updated to the sample. This process is repeated for a predefined number of iterations with a predefined number of samples at each iteration.

We tested the motion planner in a simulated chest cavity mesh environment shown in figure 6. The nominal trajectory through the chest had a 15% chance of colliding after planning using the PRM. After 50 iterations of optimization, the probability of collision had been reduced to under 7%. Figure 7 shows the change in probability of collision over the optimization iterations. Using our learned kinematic model, we more than halved the probability of collision.

VI. Conclusion

We present a novel approach to the modeling of tendon-driven continuum robot kinematics. Our method explicitly models the geometric uncertainties associated with the robot kinematics through a mixture density network trained to output Gaussian mixture models that accurately reflect the distribution over geometries. We find that our model accurately represents the ground-truth distributions over geometries and is computationally faster than a current state-of-the-art Cosserat rod model. We also demonstrate the capabilities of our model in the downstream task of motion planning. We plan nominal trajectories and use the trained mixture density network to explicitly minimize the probability of collision using Bayesian Optimization.

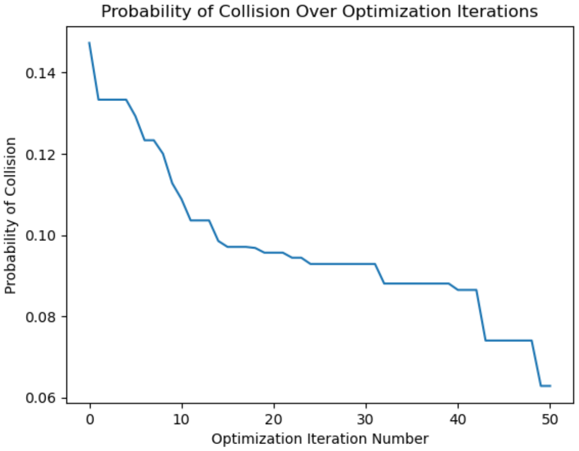

Fig. 7: Results from the optimization of a nominal trajectory in the simulated chest cavity mesh environment. We ran 50 iterations of Bayesian Optimization with respect to the probability of collision. The nominal trajectory had approximately a 15% chance of collision prior to optimization. After optimization, the probability of collision had been reduced to under 7%.

In the future we intend to apply the model to other continuum robot types. We also intend to integrate the optimization into the sampling process of a motion planner that integrates sampling with optimization for motion planning beyond the optimization of nominal trajectories, similar to as in [23], [24] but with our uncertainty model and metric.

We believe that the GMM-based kinematic model presented in this work has the potential to improve safe planning and control of tendon-driven continuum robots in medicine.

References

[1] F. Wang, H. Wang, J. Luo, X. Kang, H. Yu, H. Lu, Y. Dong, and X. Jia, "Fiora: A flexible tendon-driven continuum manipulator for laparoscopic surgery," *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 1166–1173, 2021.

[2] T. Kato, I. Okumura, S.-E. Song, A. J. Golby, and N. Hata, "Tendon-driven continuum robot for endoscopic surgery: Preclinical development and validation of a tension propagation model," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2252–2263, 2014.

[3] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[4] R. J. Webster III 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.

[5] S. Huang, D. Meng, X. Wang, B. Liang, and W. Lu, "A 3d static modeling method and experimental verification of continuum robots based on pseudo-rigid body theory," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4672–4677.

[6] D. C. Rucker and R. J. Webster III, "Statics and dynamics of continuum robots with general tendon routing and external loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, 2011.

[7] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robotics*, vol. 6, no. 6, pp. 790–811, 2019.

[8] M. Bentley, C. Rucker, and A. Kuntz, "Interactive-rate supervisory control for arbitrarily-routed multitendon robots via motion planning," *IEEE Access*, vol. 10, pp. 80 999–81 019, 2022.

[9] A. Parvaresh and S. A. A. Moosavian, "Dynamics and path tracking of continuum robotic arms using data-driven identification tools," *Robotica*, vol. 40, no. 4, pp. 1098–1124, 2022.

[10] R. F. Reinhart and J. J. Steil, "Hybrid mechanical and data-driven modeling improves inverse kinematic control of a soft robot," *Procedia Technology*, vol. 26, pp. 12–19, 2016.

[11] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance," *Frontiers in Robotics and AI*, vol. 7, p. 630245, 2021.

[12] A. Kuntz, A. Sethi, R. J. Webster, and R. Alterovitz, "Learning the complete shape of concentric tube robots," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 140–147, 2020.

[13] W. Shen, G. Yang, T. Zheng, Y. Wang, K. Yang, and Z. Fang, "An accuracy enhancement method for a cable-driven continuum robot with a flexible backbone," *IEEE Access*, vol. 8, pp. 37 474–37 481, 2020.

[14] J. Chen and H. Y. Lau, "Learning the inverse kinematics of tendon-driven soft manipulators with k-nearest neighbors regression and gaussian mixture regression," in *International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2016, pp. 103–107.

[15] J. Kruse, "Technical report: Training mixture density networks with full covariance matrices," *arXiv preprint arXiv:2003.05739*, 2020.

[16] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[17] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.

[18] W. Sun, L. G. Torres, J. Van Den Berg, and R. Alterovitz, "Safe motion planning for imprecise robotic manipulators by minimizing probability of collision," in *Robotics Research: The 16th International Symposium ISRR*. Springer, 2016, pp. 685–701.

[19] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[20] S. Patil, J. Van Den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3238–3244.

[21] M. Pelikan, D. E. Goldberg, E. Cantú-Paz *et al.*, "Boa: The bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, vol. 1, no. 1999, 1999.

[22] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, pp. 345–383, 2001.

[23] A. Kuntz, C. Bowen, and R. Alterovitz, "Fast anytime motion planning in point clouds by interleaving sampling and interior point optimization," in *Robotics Research*, vol. 10, Jan. 2020, pp. 929–945.

[24] A. Kuntz, M. Fu, and R. Alterovitz, "Planning high-quality motions for concentric tube robots in point clouds via parallel sampling and optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 2205–2212.