Efficient Force Estimation for Continuum Robot

Qingyu Xiao¹, and Yue Chen¹

Abstract—External contact force is one of the most significant information for the robots to model, control, and safely interact with external objects. For continuum robots, it is possible to estimate the contact force based on the measurements of robot configurations, which addresses the difficulty of implementing the force sensor feedback on the robot body with strict dimension constraints. In this paper, we use local curvatures measured from fiber Bragg grating sensors (FBGS) to estimate the magnitude and location of single or multiple external contact forces. A simplified mechanics model is derived from Cosserat rod theory to compute continuum robot curvatures. Least-square optimization is utilized to estimate the forces by minimizing errors between computed curvatures and measured curvatures. The results show that the proposed method is able to accurately estimate the contact force magnitude (error: 5.25% – 12.87%) and locations (error: 1.02% – 2.19%). The calculation speed of the proposed method is validated in MATLAB. The results indicate that our approach is 29.0 – 101.6 times faster than the conventional methods. These results indicate that the proposed method is accurate and efficient for contact force estimations.

Index Terms—Force Estimation, Continuum Robots

I. INTRODUCTION

CONTINUUM robots could easily change their configurations due to external contact, which will affect the robot tip motion and eventually alter the control efforts. Many work has been done to estimate the external contact forces. Contact force can be directly measured by embedding miniature force sensors on the soft robot body [1], [2]. However, these methods are only effective when the contact occurs at the sensor location, for instance at the tip of the robot. Also, the strict dimension constraints also limit the wide application of these approaches, especially in the minimally invasive surgical tools where the tool tip often equipped with multiple treatment or diagnosis units [3]. Alternatively, continuum robot contact force can be estimated based on actuator space measurement. For example, Bajo et al. [4], [5] proposed a method for tendon-driven continuum robot contact force sensing, but this requires dedicated force cell at the actuation unit. Also, the actuation-space force cells may not be available for other continuum devices such as concentric tubes, steerable needles, catheters, or guidewires.

Alternatively, force can be estimated implicitly based on the measurements of continuum robot configuration change. This is applicable since the intrinsic compliant nature of continuum robots allows the robot configuration to be changed with the contact forces [6]. Numerous models have been published to predict the continuum robot configurations subjected to external loads, including beam-based method [7], calibration method [8], variable curvature method [9], and Cosserat rod theory [10]. Contact forces, in turn, can be estimated by matching the measured configuration with modelled data.

Many shape-based methods were proposed to estimate the contact forces. Prior research used shape-based method to estimate the tip force [11]–[15], but they were unable to estimate the forces applied along the body of continuum robot. Recent publications have shown the feasibility of using shape-based method to estimate the contact force along the body of the continuum robot. Qiao [16] proposed a shape-based method which can detect the point forces (mean error of 15.4%) and locations (mean error of 6.51%). Aloi [17] proposed a shape-based method based on Fourier transformation. This method will estimate the force in a distributed fashion and can detect the force location by observing the "peaks" of the distributed force profile. However, it cannot provide useful information to the exact force magnitude and location if it is point contact.

Aside from measuring the shape of continuum robot, curvature, which is invariant under rotation, can provide more robust information for force estimation. This is because the curvature will not change for any rotations the continuum robot undergoes. Thus, the curvature-based methods are still valid even if the robot orientation is not well-calibrated. The curvatures along the continuum robot can be measured using fiber Bragg grating sensors (FBGS) [18]–[22]. Several curvature-based methods have been investigated for force estimation. Qiao [23] utilized FBGS to measure the curvatures along the manipulator and calculates the force (mean error of 9.8% error) and locations (mean error of 3.6% error) directly using constitutive law. Al-Ahmad [24] utilized unscented Kalman filter to the measured curvatures, which achieved 7.3% error of force magnitude and 4.6% error of location.

In this paper, we present a new curvature-based force estimation method by representing the Cosserat rod theory in local frame to enable faster curvatures computation, which avoid the time-consuming integration of rotation matrix. Based on the simplified model, least squares optimization is adopted to estimate the contact forces in a significantly efficient approach.

The contributions of this paper include:

1) Analyzed the conditions for accurate force estimation via curvatures.
2) Derived a simplified model for fast curvature computation and force estimation.
3) Experimentally validated the proposed force estimation method on single, double and triple force cases.

The structure of the paper is arranged as follows. Section II describes the derivation of a simplified model from Cosserat rod theory to compute curvatures. Section III discusses the conditions for pure curvature-based force estimation method...
and the estimation procedures. Section IV shows the experimental results of the force estimation regarding accuracy and computation time. Finally, Section V is the conclusion of this paper.

II. CURVATURE CALCULATION
A. Review of Cosserat Rod Model

Cosserat rod model describes the equilibrium state of a small segment of thin rod subjected to internal and external distributed forces as well as internal and external distributed moments. Previous works [25] have bridged the differential geometry of a thin rod with Cosserat rod theory so that the deformation of the rod can be estimated by solving ODEs in [1] - [4] given external forces \( \mathbf{F}_e \) and external moments \( \mathbf{L}_e \). To distinguish variables from different frames, we use lowercase letters for variables in local frame and uppercase letters for variables in global frame.

\[
\dot{\mathbf{P}} = \mathbf{Rv} \quad (1)
\]

\[
\dot{\mathbf{R}} = \mathbf{R\ddot{u}} \quad (2)
\]

\[
\mathbf{N} = -\mathbf{F}_e \quad (3)
\]

\[
\mathbf{M} = -\dot{\mathbf{P}} \times \mathbf{N} - \mathbf{L}_e \quad (4)
\]

where the dot symbol in \( \dot{x} \) represents the derivative of \( x \) respect to the arc length \( s \), hat symbol in \( \hat{x} \) reconstruct vector \( \mathbf{x} \) to a 3 by 3 skew symmetric matrix, \( \mathbf{P} \) is the shape in global frame, \( \mathbf{R} \) is the rotation matrix of local frame relative to the global frame, \( \mathbf{N} \) and \( \mathbf{M} \) are internal force and moment in global frame which obey the constitutive law

\[
\mathbf{N} = \mathbf{R}\mathbf{K}_{SE}(\mathbf{v} - \mathbf{v}^*) \quad (5)
\]

\[
\mathbf{M} = \mathbf{R}\mathbf{K}_{BT}(\mathbf{u} - \mathbf{u}^*) \quad (6)
\]

\( \mathbf{v} \) and \( \mathbf{u} \) are differential geometry parameters of a rod, \( \mathbf{v}^* \) and \( \mathbf{u}^* \) are differential geometry parameters of a rod with no external loads. \( \mathbf{K}_{SE} \) and \( \mathbf{K}_{BT} \) are the stiffness matrices. The boundary conditions for solving the ODEs (1) - (4) are described as

\[
\mathbf{N}(\text{loc}_{tip}) = \mathbf{F}_{tip} \quad (7)
\]

\[
\mathbf{M}(\text{loc}_{tip}) = \mathbf{T}_{tip} \quad (8)
\]

where \( \mathbf{F}_{tip} \) and \( \mathbf{T}_{tip} \) are external force and torque applied at the tip. \( \text{loc}_{tip} \) is the tip position in arc length. Thus, the rod shape \( \mathbf{P} \) can be computed by solving the boundary value problem (BVP).

B. ODEs For Curvature Calculation

The widely accepted Cosserat rod model in [1] - [4] are expressed in global frame. But it is beneficial to use the model in local frame to calculate the curvatures. We firstly take the derivative of \( \mathbf{M} \) in (6) and combine with (4)

\[
\dot{\mathbf{R}}\mathbf{K}_{BT}(\mathbf{u} - \mathbf{u}^*) + \mathbf{R}\mathbf{K}_{BT}\mathbf{u} = -\dot{\mathbf{P}} \times \mathbf{N} - \mathbf{L}_e \quad (9)
\]

Then, using the relation \( \dot{\mathbf{P}} \times \mathbf{N} = \dot{\mathbf{P}}\mathbf{N}, \dot{\mathbf{P}} = \mathbf{Rv}, \) and \( (\dot{\mathbf{R}}v) = \dot{\mathbf{R}}v\mathbf{R}^T \) (this is true when \( \mathbf{R} \in \text{SO}(3) \)), we can obtain

\[
\dot{\mathbf{R}}\mathbf{K}_{BT}(\mathbf{u} - \mathbf{u}^*) + \mathbf{R}\mathbf{K}_{BT}\mathbf{u} = -\dot{\mathbf{R}}v\mathbf{R}^T \times \mathbf{N} - \mathbf{L}_e \quad (10)
\]

Multiplying \( \mathbf{R}^T \) on both side, and solve for \( \mathbf{u} \), we can have

\[
\dot{\mathbf{u}} = -K_{BT}^{-1}(\dot{\mathbf{u}}\mathbf{K}_{BT}(\mathbf{u} - \mathbf{u}^*) + \dot{\mathbf{v}}\mathbf{R}^T \times \mathbf{N} + \mathbf{R}^T \mathbf{L}_e) \quad (11)
\]

We define local variables \( \mathbf{n} = \mathbf{R}^T \mathbf{N}, \) and \( \mathbf{l}_e = \mathbf{R}^T \mathbf{L}_e \). The meaning of the variables \( \mathbf{n} \) and \( \mathbf{l}_e \) are actually the variables \( \mathbf{N} \) and \( \mathbf{L}_e \) expressed in local frame, respectively. Hence, (11) becomes

\[
\dot{\mathbf{u}} = -K_{BT}^{-1}(\dot{\mathbf{u}}\mathbf{K}_{BT}(\mathbf{u} - \mathbf{u}^*) + \dot{\mathbf{v}}\mathbf{n} + \mathbf{l}_e) \quad (12)
\]

In order to solve \( \mathbf{n} \), we use (3) and multiply \( \mathbf{R}^T \) on the two sides and add \( R^T \mathbf{N} \), which gives

\[
\mathbf{R}^T \mathbf{N} + \mathbf{R}^T \mathbf{N} = -\mathbf{R}^T \mathbf{F}_e + \mathbf{R}^T \mathbf{N} \quad (13)
\]

Again, we define \( \mathbf{R}^T \mathbf{F}_e = \mathbf{f}_e \). Notice that the left hand side in (13) is actually the derivative of \( \mathbf{R}^T \mathbf{N} \), which is \( \dot{\mathbf{n}} \). \( \mathbf{R} \) can be replaced by (2) on the right hand side. Thus we have

\[
\dot{\mathbf{n}} = -\mathbf{f}_e + \dot{\mathbf{u}}\mathbf{n} \quad (14)
\]

To sum up, the curvatures of a general rod can be calculated by

\[
\begin{align*}
\dot{\mathbf{u}} &= -K_{BT}^{-1}(\dot{\mathbf{u}}\mathbf{K}_{BT}(\mathbf{u} - \mathbf{u}^*) + \dot{\mathbf{v}}\mathbf{n} + \mathbf{l}_e) \\
\dot{\mathbf{n}} &= -\mathbf{f}_e + \dot{\mathbf{u}}\mathbf{n}
\end{align*}
\]

where \( \dot{\mathbf{loc}}_{tip} \) and \( \dot{\mathbf{f}}_{tip} \) are the location and applied external force at the tip, respectively. (15) and (16) can be simplified in a special case where the rod is straight (\( \mathbf{u}^* = \mathbf{0} \) and \( \mathbf{v}^* = [0 \ 0 \ 1]^T \)), with circular cross-section (\( K_{11} = K_{22} = K_{33}/2 \)), non-shear and inextensible (\( \mathbf{v} = [0 \ 0 \ 1]^T \)), and no external moment (\( \mathbf{l}_e = \mathbf{0} \)). The simplified ODEs can be written as

\[
\begin{align*}
\dot{u}_x &= -n_x/K_{BT,11} \\
\dot{u}_y &= n_x/K_{BT,22} \\
\dot{n}_x &= -f_x + u_t n_z \\
\dot{n}_y &= -f_y + u_t n_z \\
\dot{n}_z &= -f_z + u_x n_x + u_y n_y
\end{align*}
\]

where \( x, y, \) and \( z \) are the first, second, and third component of a vector, respectively. \( K_{BT,ij} \) means the entry of \( \mathbf{K}_{BT} \) at row \( i \) and column \( j \). Appendix A shows the detailed procedures to obtain the ODEs (19) - (22). And (21) - (23) is the same as (16). Note that both the integral variables and the boundary conditions are defined in local frame, the curvature \( u_t \) and \( u_y \) can be solved by simple backward integration from the distal point (\( s = L \)) to the proximal point (\( s = 0 \)), which requires no iterative computation to solve a standard BVP. Therefore, the computational complexity is only determined by the number of nodes we divide along the arc length. Moreover, solving (19) - (23) will lead to the same result as solving the model in (1) - (4), but the former method will have faster performance for curvature calculation.
III. Force Estimation Approach

A. Conditions For Accurate Force Estimation

While computing the configuration of a rod under external forces and moments can be straightforward, its inverse mechanics is ill-conditioned [26]. Force estimation based on variables in configuration space can become inaccurate even with perfect data measurement. In this subsection, we analyze the conditions where force can be estimated accurately. We assume the mapping between the measured curvature and estimated force is

\[ \mathcal{F}(\hat{u}) = \hat{f}_e \]  

(24)

where \( \mathcal{F} \) is a Cosserat model-based method to estimate the force \( \hat{f}_e \) from measured curvature \( \hat{u} \). The error of the estimation can be defined as

\[ E_G(\hat{u}) = \| \hat{f}_e - f^g_e \| \]  

(25)

where \( E_G(\hat{u}) \) is the 2-norm error of method \( \mathcal{F} \) with measured \( \hat{u} \). \( f^g_e \) is the ground true force. If the estimation method \( \mathcal{F} \) is sufficiently accurate, we can write

\[ E_G(u) \leq E_G(\hat{u}) \]  

(26)

where the \( u \) is the ideal curvature measurement with no noise. Ideal measurements on curvature or points of shape cannot guarantee that \( E_G(u) = 0 \). Additional information has to be assumed, or measured in order to achieve accurate results. To derive what else information is necessary for force estimation, we assume the ideal measurements of \( u \) is known. As shown in Fig. 1, variables \( P, R, M \) and their derivatives can be directly computed from ideal measurements using Cosserat rod model. However, because of the cross product in (4), unique solution of \( F_e \) can not be achieved with only curvature measurements. Thus, we list two special cases in Fig. 1 that can theoretically complete the information for force estimation.

Case 1 indicates the internal force \( N \) need to be measured in additional to curvatures for force estimation. The application of case 1 requires multiple force sensors mounted along the length of the manipulator. Case 2 requires the knowledge of \( L_e \) and the direction of \( F_e \). But assumptions can be made on these two variables:

1) No external moments (\( L_e = 0 \)).
2) External forces are always perpendicular to the surface of the manipulator \( F_{e,z} = 0 \).

One advantage of these assumptions is that force can be estimated without any additional mounted sensors. However, these assumptions also limit the application range of force estimation. The surface of the manipulator has to be smooth, and the manipulator has to work in an environment where friction is trivial to the shape. Compared with case 1, case 2 with the aforementioned two assumptions allows force estimation without additional sensors. The goal of this paper is to estimate force only from curvature data. Thus, we will use these assumptions throughout the following paper.

B. Point Forces For Cosserat Model

The force defined in Cosserat rod theory is distributed force whose unit is N/m. However, the objective of this paper is to estimate the magnitude and location of point force whose unit is N. Conversion has to be performed between point force and distributed force. Firstly, we define the applied point forces in a new force vector which contains the locations and magnitude components of all forces.

\[ f_{vec} = [s_1^1, f_{pt,z}^1, f_{pt,y}^1, s_2^2, f_{pt,z}^2, f_{pt,y}^2, \ldots, s_h^h, f_{pt,z}^h, f_{pt,y}^h]^T \]  

(27)

where \( s^i \) is the location of the \( i^{th} \) point force, \( f_{pt,z}^i \) and \( f_{pt,y}^i \) are the components of the \( i^{th} \) point force, and \( h \) is the total number of forces acting on the continuum robot. The arc length of the manipulator can be divided into \( q - 1 \) segments, and this gives

\[ Loc = [loc_1, loc_2, \ldots, loc_q] \]  

(28)

where \( loc_i \) is the location of the \( i^{th} \) node, \( Loc \) is the whole list of the nodes. The conversion from point forces to distributed forces is the process to distribute \( f_{vec} \) on the nodes \( Loc \). Fig. 2 shows an example to distribute the point force \( f_{pt}^j \) to two adjacent nodes \( j-1 \) and \( j \), the point force is distributed linearly according to the arc length distance between the two nodes. Therefore, the distributed forces can be calculated by

\[ f^{j-1} = \frac{f_{pt}^j}{(loc_j - loc_{j-1})^2} (loc_j - loc_{j-1}) \]  

(29)

\[ f^j = \frac{f_{pt}^j}{(loc_{j-1} - loc_j)^2} (loc_{j-1} - loc_j) \]  

(30)

In the case where multiple point forces are close and distributed to the same node, the forces on that node will be superposed. After knowing the distributed forces along the rod. The mechanics model is complete and curvature can then be calculated through the integration of (19) - (25).

C. Force Estimation From Observed Curvature

A robust method for force estimation is to minimize the least square loss between the calculated curvature with measured curvatures.

\[ loss_u(f_{vec}) = \| u_{cal} - \hat{u} \|^2 \]  

(31)
where $u_{cal}$ is the curvature calculated from $f_{vec}$, $\hat{u}$ is the curvature measured by sensors.

Similar to minimizing the loss of curvature, the loss of shape $\text{loss}_p(f_{vec})$ could also be used for the objective function of the optimization algorithm. Fig. 5 shows an simulation example of the loss of shape $\text{loss}_p(f_{vec})$ and loss of curvature $\text{loss}_u(f_{vec})$ with various force magnitudes and locations. The loss map is calculated by following steps: 1) Choosing $f_{vec} = [200 \ 0.3 \ 0]^T$ as the ground truth, and calculate the curvature $\hat{u}$ and shape $\hat{p}$ using Cosserat rod theory. The $\hat{u}$ and $\hat{p}$ can be assumed as the measured data from sensors. 2) Calculate $u_{cal}$ and $P_{cal}$ with the same method, but use different $f_{vec}$ whose first component ranges from 100 - 290 mm and the second component ranges from 0 - 0.5 N. 3) use (31) to compute the loss for each pair of force magnitude and location. As illustrated in Fig. 5, the loss of curvature (b) shows better convex property than that of shape coordinates (a), and thus more robust results can be calculated [27]. Therefore, we use curvature based method to estimate the force.

![Fig. 2. Distribution of point force to adjacent nodes.](image)

![Fig. 3. Loss maps computed by (a) shape (b) curvature.](image)

**IV. EXPERIMENTS AND RESULTS**

A. Experiment Setup

As illustrated in Fig. 4, the experiments were performed by adding multiple forces to a straight Nitinol tube. ATI force sensor (ATI Industrial Automation, United States) with 3D printed probe was mounted on a fixed vertical board. The probes are designed with different height in order to contact with the Nitinol tube (Fig. 4c). For every single probe that mounted on the force sensor, a counterpart probe (Fig. 4a and Fig. 4b) is also designed to keep the same distance from the probe head to the wall. This allows the force sensor with probe to be interchangeable with its counterpart such that the shape of Nitinol tube remains unchanged. For example, the Nitinol tube in Fig. 4a and Fig. 4b has the same shape, but the location of the force sensor swapped. Thus, we can use one force sensor to measure multiple external forces. Fiber bragg gratings sensors (FBGS International NV, Belgium) was inserted inside the Nitinol tube to measure the curvature.

![Fig. 4. Experiment setup for force estimation. The ATI force sensor with a probe is mounted on a fixed vertical wall. The FBGS is inserted inside Nitinol tube for curvature measurement. Counterpart of each force sensor with probe are designed to keep the shape of Nitinol in (a) and (b) the same, and the force sensor can swap the location. Different height design of the probe in (c) allows for 3D shape deformation.](image)

B. Model Calibration

The calibration of the experiment was conducted in two step: 1) location calibration 2) stiffness calibration. Since the FBGS fiber is transparent, it is hard to figure out the locations of gratings inside the fiber. But the relative locations (spacing between adjacent gratings) are specified on the user manual (20 mm). The objective of location calibration is to ensure the curvatures we measured are aligned with the positions on the Nitinol. Three cases of single force at different locations were recorded, and the bias of the location can be minimized by adding an offset to the results.

$$s_{est} = s_{cal} + s_{bias}$$  \hspace{1cm} (32)

where $s_{est}$ is the estimated location of the external force, $s_{cal}$ is the result of the model, and $s_{bias}$ is the bias to offset the error. After location calibration, we conducted stiffness calibration to match the calculated force magnitude with the measured force magnitude. Notice that the stiffness change has trivial impact on the location estimation, therefore the location calibration is still valid after completing the stiffness calibration. The result of the calibration is listed in TABLE I.

![TABLE I

| Name            | Variable | Value | Unit |
|-----------------|----------|-------|------|
| Nitinol tube length | loctip   | 290   | mm   |
| Inner tube diameter   | $d_{in}$ | 1.118 | mm   |
| Outer tube diameter   | $d_{out}$ | 1.397 | mm   |
| Young’s modulus       | $E_{ni}$ | 67    | GPa  |
| Location bias         | $s_{bias}$ | -3.12 | mm   |

C. Force Number Estimation

The output of the force estimation method is a force vector $f_{vec}$ defined in (27), which specifies the number of force $h$. 

The ideal estimation result should assign the force number $h$ in $f_{vec}$ exactly the same as the real force number before running the force estimation method. In practice, knowing the number of forces on a manipulator in advance is hard through other measurements. Mismatching the model force number with real force number will reduce the estimation accuracy, even though the magnitude of the redundant forces are small. However, as shown in Fig. 5, the mismatch of force number can be detected by setting a threshold for curvature loss $\text{loss}_u(f_{vec})$.

For the situation where $h$ is less than the real force number, it is obvious that a large curvature loss occurs. This indicates that $h$ needs to increased in the model for better estimation. For the situation where the $h$ is larger than the real force number, the curvature loss value drops significantly. This indicates that a smaller $h$ will be assigned in the model for better estimation. In this paper, we start with $h = 1$ and gradually increase the $h$ until the curvature loss drops below a specified threshold (3.0, in our case) at the first time. This is also valid for single force estimation, the curvature loss of which is below the threshold at the beginning ($h = 1$).

**D. Force Estimation Results**

The error analysis of the force estimation result is grouped into 3 categories: single force estimation analysis, double forces estimation analysis, and triple forces estimation analysis. For each category, the experiment was performed 13 times by applying varying contact force (0.27 - 1.96 N) at 13 different locations on the Nitinol tube. Intuitively, the number of nodes $q$ can also have impact on the estimation results. As demonstrated in Fig. 8, the RMSE of both force magnitude and location decrease as the number of nodes $q$ increases. However, if the $q$ is sufficiently large, RMSE can barely decrease any more.

For the force magnitude estimation with $q = 250$ as shown in Fig. 8b, the RMSE of single-location estimation, double-location estimation, and triple-location estimation is $2.95 \pm 2.11$ mm, $3.15 \pm 2.79$ mm, and $5.1873 \pm 5.64$ mm, respectively. This indicates that the larger location error will occur when the number of external forces increases. The worst prediction in triple force location error is 10.59 mm, which is 3.65% of the total length ($loc_q = 290$ mm).

Fig. 6 and Fig. 7 summarize the the force estimation results ($q = 250$) for single force and multiple forces, respectively. The blue dots are the measured result and the red dots are the predicted results. Most of the cases in Fig. 6 can be predicted accurately except the case 1 (magnitude error of 8.72%). This error occurs because the curvature can only be measured by the first few gratings (the spacing of adjacent gratings is 20 mm), while the rest gratings will read 0 because no internal moment exists after the location where the force applied. In Fig. 7, each case has two or three values, which refer to the two or three forces, respectively. The red dots and the blue dots are aligned well with each other, which demonstrate the accuracy of the proposed estimation method.

**E. Computational Speed**

In this section, we aim to evaluate the feasibility for realtime force estimation by analyzing the force estimation speed.
The laptop for the speed test has a CORE-i7 Intel CPU with 6 cores. The force estimation method was implemented in MATLAB, and the optimization problem is solved using \textit{fmincon} function with interior-point algorithm. The number of nodes \( q \) as well as the number of inputs (forces and locations) also have significant impact on the calculation speed. We tested the method with \( q \) in the range of 50 and 250. For \( q \) larger than 250, the accuracy of this method will not improve, as shown in Fig. 8. Since our main motivation to increase \( q \) is to ensure the accuracy of the estimation, \( q \) larger than 250 will be excluded in the test.

The consumed time for single, double, and triple forces estimation is shown in Fig. 9. For best accuracy (\( q = 250 \)), single force estimation can be completed in 0.315 ± 0.080 s, double force estimation can be completed in 0.857 ± 0.191 s, and the triple force estimation can be completed in 1.890 ± 0.078 s. For better performance, smaller \( q \) which does not significantly compromise the accuracy is preferred to achieve faster force estimation. For the case \( q = 90 \), single force estimation takes 0.134 ± 0.038 s, double force estimation takes 0.344 ± 0.088 s, and triple force estimation takes 0.730 ± 0.045 s.

Notice that we can calculate the same curvature by solving BVP proposed in [25] using Levenberg-Marquardt (LM) method, or employ the derivative propagation (DP) method proposed in [28] to speed up the iterations. We compare the computation time for the three approaches for various \( q \). TABLE II lists out the computation time for single force estimation. Our method shows much faster computation speed than the other two methods. Typically, when \( q = 100 \), our method is 29.0 times faster than derivative propagation method, and 101.6 times faster than Levenberg-Marquardt method. The computational speed can be further improved by implement the method in C++ language, or use better optimization algorithm if the Jacobian of the curvature loss can be computed in a faster method.

V. CONCLUSIONS

In this article, we analyzed the mechanics model and come up with the conditions on which the force can be estimated with only curvature measurements. After having those conditions (no external moment and no axis force such as friction), a simplified Cosserat rod theory was derived to compute the curvature in a fast speed. Least squares optimization was used to minimize the loss between computed curvatures and measured curvatures to find the optimal force vector \( \mathbf{f}_{sec} \), which contains the estimated force magnitudes and locations.

The proposed method was validated on a straight 290 mm Nitinol tube, with single or multiple forces acting at different locations. The results showed that the model can estimate both the force magnitude and location accurately. The RMSE of single force estimation, double force estimation, and triple force estimation are 0.084 ± 0.073 N, 0.115 ± 0.102 N, and 0.1090 ± 0.1173 N, respectively. The RMSE of the single-location estimation, double-location estimation, and triple-location estimation is 2.95 ± 2.11 mm, 3.15 ± 2.79 mm, and 5.1873 ± 5.64 mm, respectively. Moreover, the computation time is tested, showing that the single force estimation, double force estimation, and triple force estimation can be completed in 0.134 s, 0.344 s, and 0.730 s in MATLAB R2021a. This speed is 29.0 time faster than solve BVP with DP, and 101.6 times faster than solving BVP with LM.
A. Deriving Curvature ODEs

In this section $K = K_{RT}$ is used for convenience. We firstly use conditions $u^t = 0, I_0 = 0$ to simplify (15):

$$Ku = \dot{u}Ku + \dot{v}n$$

(33)

Expand the term $\dot{u}Ku$ in (33),

$$\dot{u}Ku = \begin{bmatrix} 0 & -u_z & u_z \\ u_z & 0 & -u_z \\ -u_z & u_z & 0 \end{bmatrix} \begin{bmatrix} K_{11} & 0 & 0 \\ 0 & K_{22} & 0 \\ 0 & 0 & K_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_yK_{11} \\ u_xK_{22} \\ u_zK_{33} \end{bmatrix}$$

(34)

$$\dot{u}u_t(K_{33} - K_{22})$$

(35)

$$u_tu_t(K_{33} - K_{22})$$

(36)

For the term $\dot{v}n$ in (33), we use Kirchhoff assumption $v = [0 \ 0 \ 1]^T$ and multiply all the elements

$$\dot{v}n = \begin{bmatrix} -n_y \\ n_x \\ 0 \end{bmatrix}$$

(37)

For the term $Ku$ in (33), we have

$$Ku = \begin{bmatrix} K_{11}u_x \\ K_{22}u_y \\ K_{33}u_z \end{bmatrix}$$

(38)

Now, consider the circular cross sectional condition ($K_{11} = K_{22} = K_{33}/2$) and substitute in (36) will have

$$\dot{u}Ku = \begin{bmatrix} u_yu_z/2 \\ -u_xu_z/2 \\ 0 \end{bmatrix}$$

(39)

Combine the third row in (37) - (39) will have

$$u_z = 0$$

(40)

This means the curvature at z-direction will not change for this special case. Recall the boundary conditions specified in [17], we can conclude

$$u_z = 0$$

(41)

Substitute this conclusion back to (39) will obtain

$$\dot{u}Ku = 0$$

(42)

Therefore, substitute (37), (38), and (42) into (33) will simplify the ODEs

$$\dot{u}_x = -n_y/K_{11}$$

(43)

$$\dot{u}_y = n_x/K_{22}$$

(44)

Notice that $u_z$ is always zero and has no impact on the other variables, therefore it is excluded from the ODEs.
[21] T. Li, L. Qiu, and H. Ren, “Distributed curvature sensing and shape reconstruction for soft manipulators with irregular cross sections based on parallel dual-fbg arrays,” IEEE/ASME Transactions on Mechatronics, vol. 25, no. 1, pp. 406–417, 2019.

[22] D. Barrera, I. Gasulla, and S. Sales, “Multipoint two-dimensional curvature optical fiber sensor based on a nontwisted homogeneous four-core fiber,” Journal of Lightwave Technology, vol. 33, no. 12, pp. 2445–2450, 2015.

[23] Q. Qiao, D. Willems, G. Borghesan, M. Ourak, J. De Schutter, and E. Vander Poorten, “Estimating and localizing external forces applied on flexible instruments by shape sensing,” in 2019 19th International Conference on Advanced Robotics (ICAR). IEEE, 2019, pp. 227–233.

[24] O. Al-Ahmad, M. Ourak, J. Vlekken, and E. Vander Poorten, “Fbg-based estimation of external forces along flexible instrument bodies,” Frontiers in Robotics and AI, vol. 8, 2021.

[25] B. A. Jones, R. L. Gray, and K. Turlapati, “Three dimensional statics for continuum robotics,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009, pp. 2659–2664.

[26] S. I. Kabanikhin, “Definitions and examples of inverse and ill-posed problems,” 2008.

[27] S. Bubeck, “Convex optimization: Algorithms and complexity,” arXiv preprint arXiv:1405.4980, 2014.

[28] D. C. Rucker and R. J. Webster, “Computing jacobians and compliance matrices for externally loaded continuum robots,” in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 945–950.