Real-time Adaptive Kinematic Model Estimation of Concentric Tube Robots

Chunwoo Kim, Seok Chang Ryu and Pierre E. Dupont

Abstract—Kinematic models of concentric tube robots have matured from considering only tube bending to considering tube twisting as well as external loading. While these models have been demonstrated to approximate actual behavior, modeling error can be significant for medical applications that often call for positioning accuracy of 1-2mm. As an alternative to moving to more complex models, this paper proposes using sensing to adaptively update model parameters during robot operation. Advantages of this method are that the model is constantly tuning itself to provide high accuracy in the region of the workspace where it is currently operating. It also adapts automatically to changes in robot shape and compliance associated with the insertion and removal of tools through its lumen. As an initial exploration of this approach, a recursive on-line estimator is proposed and evaluated experimentally.

I. INTRODUCTION

Concentric tube robots are a promising continuum robot technology that are being applied to a broad variety of interventions in such fields as neurosurgery [1], [2] and cardiac surgery [3], [4]. Comprised of a telescoping combination of precurved superelastic tubes, the shape of these robots is controlled by rotating and translating each tube with respect to the others at their base. Many aspects of this technology have been investigated including kinematic modeling [5]–[7], design and workspace evaluation [8]–[10], stability [11]–[13] and motion planning [14].

A continuing challenge, however, is that the most frequently used kinematic model [5], [6], while formulated from principles of mechanics, is a function of a small number of imprecisely known parameters. Furthermore, as designers of these robots attempt increasingly challenging applications, simplifying assumptions inherent in the model are being violated. For example, while the assumption of linear elasticity typically holds up to about 1% strain for NiTi [15], designs fabricated by our group often use significantly higher strains. Additional factors such as cross section ovalization and friction also contribute to model error.

Furthermore, the kinematic model can be time varying. Many procedures require the insertion and removal of tools and implants through the lumen of the robot. See, e.g., [3], [4]. As shown in Fig. 1(a), these devices are mounted at the tips of tubes that are inserted from the base of the robot. While designed to be compliant, these tubes change both the curvature and the compliance of the robot when inserted or removed. An additional source of model variation with time corresponds to relaxation of tube precurvature with use.

Modeling error manifests as steady-state position or force error during closed-loop control. For example, most current implementations of tip position control for concentric tube robots are kinematic-model-based feedforward controllers [5]–[7]. During teleoperation, the operator manually closes a feedback loop by observing slave error and then moving the master beyond the desired position and orientation in order to drive the slave to the desired configuration. This can result in a confusing mismatch of configuration between master and slave as shown in Fig. 1(b).

The addition of a sensor-based feedback loop can improve...
performance [16], [17], but is currently challenging to implement in practice since existing sensors, e.g., electromagnetic, tend to fill the entire robot lumen leaving no room for tools. As better sensors become available, the smoothest, most accurate motions will result from the combination of an accurate model-based feedforward controller with sensor-based feedback.

Several approaches to reducing model error are possible. The first is to derive more sophisticated models that incorporate some combination of neglected phenomena. This approach will help the community sort through the relative importance of the various effects. This will, however, add to the complexity of solving the equations in real time. Furthermore, the insertion and removal of additional tubes for tool delivery is an important case. And, while this can be certainly modeled analytically, it is impossible to predict what tube or tube shape a clinician may attempt to insert through a robot in an actual procedure.

These observations motivate an alternative approach, which is to use existing models, but to adaptively update the model parameters. This approach, while requiring sensing, provides several advantages. For example, the model can automatically adapt to changes associated with the insertion and removal of tool tubes. Furthermore, the parameters of a kinematic model are typically calibrated to minimize error over the entire workspace of a robot. Many surgical procedures, however, are confined to small regions of the workspace. Adaptive tuning of model parameters within these regions at the start of a procedure can enable substantial local error reduction. It is always possible to save both the global parameter values as well as parameters tuned for different workspace regions in order to facilitate transitioning outside the current region.

Considering tip position control, the sensing required for model adaptation could be provided by electromagnetic tracking [16], [17], FBG curvature sensing [18] or by imaging [19], [20]. Most of these sensors have a separate coordinate system that must be registered to the robot coordinate system. Another advantage of model adaptation is that the registration transformation can be incorporated into the kinematic model and included in the online adaptation.

On-line model estimation and adaptive control have been considered extensively for rigid robots. See, e.g., [21]–[23]. Much of this work exploits the fact that the kinematic and dynamic equations of rigid manipulators are linear in terms of the link parameters, which is not generally the case for continuum robots.

In the context of tendon-based continuum robots, a few papers have considered adaptive modeling [24], [25]. For example, in [24], recursive linear estimation is used to update friction and backlash parameters for actuation compensation. In [25], the authors propose a technique called model-less control in which the kinematic Jacobian is continuously estimated based on commanded differential motions of the actuators and sensed tip location - without using a nominal underlying kinematic model. In an alternative approach, Gaussian Process Regression was used to learn an unknown function that modifies the control input to compensate for the imprecise internal kinematics model of a cable-driven surgical assistant robot [26].

The contribution of this paper is to propose, develop and validate a method for on-line adaptive kinematic modeling of concentric tube robots. This approach can provide some of the robustness to disturbances addressed by model-less control while still providing the real-time error reduction associated with accurate feedforward control. Section II summarizes the current mechanics-based model as well as the functional approximation described in [5]. This section also describes a recursive approach to parameter estimation. This section also describes a recursive approach to parameter estimation. Validation experiments appear in Section III and conclusions in Section IV.

II. KINEMATIC MODELING

Mechanics models of concentric tube robots have been developed [5]–[7] under certain assumptions: tubes can bend and twist elastically, but cross section shear and axial elongation are neglected. Furthermore, tube stiffness in bending and torsion is assumed to be linear elastic.

While this model and parameter set are very useful for designing a tube set with a desired workspace [12], it ignores phenomena such as friction [27] as well as nonlinear elastic material behavior. Thus, it can be anticipated that when the parameters are calibrated over the entire workspace, it will be impossible to eliminate error due to these unmodeled phenomena.

To avoid the real-time computational burden of solving this split boundary value problem, an algebraic functional approximation was previously proposed [5]. In this approach, the mechanics model was solved off line for a dense set of tip configurations over the workspace. The input-output data pairs of actuator variables (tube rotations and extensions) and tip configurations were then modeled using a product of truncated Fourier series. For example, a positional coordinate, \( p_i(s_{\text{max}}) \) was represented by

\[
p_i(s_{\text{max}}) = \left( \prod_{j=2}^{n} H(\alpha_j, q) \right) \left( \prod_{k=2}^{n} H(L_k / \lambda_k, q) \right) \tag{1}
\]

\[
H(\beta_j, q) = \sum_{k=-q}^{q} c_k e^{i(k\beta_j)} \tag{2}
\]

Here, \( H(\beta_j, q) \) is a scalar Fourier series of order \( q \) involving the actuator variable \( \beta_j \). The first product term in (1) involves the rotational inputs while the second term involves tube extension variables, \( L_k \), nondimensionalized using wavelength parameters \( \lambda_k \).

Using this representation, the forward kinematic solution reduces to an algebraic function evaluation while the inverse kinematic solution was implemented as a root finding problem. The order of the Fourier series, \( q \), determines the number of free parameters of the model and can be selected to achieve a desired maximum error over the workspace. Assuming the same order series for all actuator variables and
that each tube is rotated and translated independently, each
tip coordinate is modeled by \((2n - 1)(2q + 1)\) parameters.
While the model was originally validated based on its ability
closely match the input/output data of the mechanics-based
model, an additional unintended benefit of this approach is that the adjustable number of model parameters
may enhance the ability of the model to match the actual
kinematics.
An additional advantage is that the model is linear in its
parameters and so, for example, positional coordinate
\(p_i(s_{\text{max}})\) can be written as
\[
p_i(s_{\text{max}}) = \tilde{c}_p^T(s_{\text{max}}) \tilde{\phi}(\alpha_2, \ldots, \alpha_0; L_2, \ldots, L_m)
\]  
where \(\tilde{c}_p(s_{\text{max}}) \in \mathbb{R}^{(2n-1)(2q+1)}\) is a vector obtained by
appropriately stacking the Fourier coefficients in (1) and \(\tilde{\phi}\) is the
vector of corresponding trigonometric terms. Thus, it is
possible to refine parameter values on line using recursive
least squares as described in the subsection below.

A. Recursive Algebraic Model Estimation

Recursive least square (RLS) [28] is an algorithm to
recursively update the parameters of a linear model from
the observed output. This algorithm was chosen as it is
straightforward to implement on the functional approxi-
mation kinematics model and is linear in its parameters.
The updated parameter set is optimal in the sense that it
minimizes the weighted sum of squares of the error between
the observed and predicted output. In terms of the parameter
\(\tilde{c}\) and measured position coordinates \(p\) of (3), the estimated
parameter \(\tilde{c}_{k+1}\) from the \(k\) observation is given by
\[
\tilde{c}_{k+1} = \arg\min_{\tilde{c}} \sum_{i=1}^{k} \kappa^{k-i} |\tilde{c}_i^T \hat{\phi}_i - p_i|^2
\]  
The weighting factor \(\kappa \leq 1\) is a called a “forgetting fac-
tor” since it assigns less weight to earlier errors. Instead
of solving the optimization problem of (4), \(\tilde{c}_{k+1}\) can be recursively updated from its value \(\tilde{c}_k\) at the previous step and
the \(k^{th}\) measured position coordinates, \(p_k\), by the following
equations.
\[
e_k = p_k - \tilde{c}_k^T \hat{\phi}_k
\]  
\[
M_{k+1} = \frac{1}{\kappa} \left( M_k - \frac{1}{\kappa + \phi_k^T \hat{\phi}_k} M_k \hat{\phi}_k \hat{\phi}_k^T M_k \right)
\]  
\[
\tilde{c}_{k+1} = \tilde{c}_k + e_k M_{k+1} \tilde{c}_k
\]  
There are two free parameters that can be tuned and influ-
ence the performance of the on-line parameter estimation.
One is the initial value of the matrix \(M_0\) and the other
is the forgetting factor \(\kappa\). The matrix \(M\) is related to the
covariance of the kinematics input \(\hat{\phi}\). Its initial value, \(M_0\), can be estimated from the sampled kinematics inputs collected
prior to adaptation or it can be initialized to an identity matrix
multiplied by constant, \(M_0 = \sigma I\) [28]. The initial value can
significantly impact the rate of convergence.

Since it has not been shown that the actual kinematic
model lies in the set of models corresponding to the al-
gebraic model, it is likely that the best fit parameters of the
algebraic model will vary for different regions of the
workspace. Consequently, selection of the forgetting factor,
which controls the contribution of prior prediction errors
to the parameter update, may have significant implications
when moving between regions of the workspace. The choice
of forgetting factor is a compromise between local tracking
accuracy, speed of response to perturbations and stability
[29].

III. EXPERIMENTS

The proposed adaptive estimation method was evaluated
using the teleoperated concentric tube robot depicted in Fig.
2a. Due to space limitations, only adaptation of the position
coordinates is considered here. The robot is comprised of
three tubes with parameters listed in Table I where regions
of constant curvature are denoted by section number. The two
outer tubes form a variable curvature section and are rotated
independently, but translated together. The innermost tube is
significantly more compliant that the outer two tubes and is
translated and rotated independently of the other tubes. Thus,
the kinematic variables are \(\{\alpha_2, \alpha_3, L_3\}\). Including translation
and rotation of the tube assembly, the robot possesses five
degrees of freedom. The missing degree of freedom is the
roll orientation at the tip.

The robot controller was implemented on a PC in C++
using a CAN bus to communicate velocity commands to
each motor amplifier. Desired tip position and orientation are
read from the encoders of a 6 DOF haptic interface (Phantom
Omn) and converted to motor commands by computing the
inverse kinematic solution using a second-order model based
on (2). The model has 125 parameters per tip coordinate
which were initialized by pre-computing 40 x 40 x 40 grid
points over the workspace using measured values of tube
stiffness, curvature and length.

The controller block diagram is shown in Fig. 3. A motor-
control loop operates via the CAN bus at 1Mbps, reading
motor counts and sending velocity commands. The teleop-
eration loop operates at 1kHz reading commands from the
haptic device, computing the inverse kinematics using root
finding and updating the set point for the motor controller.
The adaptive model estimator runs at 240Hz reading the
position of the robot tip from an electromagnetic (EM) sensor
and updating the kinematic model.

An EM tracker sensor (TrackStar, NDI, Waterloo, Canada)
was mounted at the tip of the robot to track tip position and
orientation and was integrated with an adaptive parameter
estimator as shown in Fig. 3. The robot was registered to
the sensor coordinate system by rotating the entire tube set
around the its axis (Z in Fig. 2b) and then fitting a circle
to the measured position of the tip. When the robot is in
its initial configuration, the initial position of the tip in the
robot coordinate system is \((r, 0, d)\), as shown in Fig. 2b. Then,
when the tube set is rotated, the measured tip position forms
a circle of radius \(r\) on a plane \(Z=d\) with center located at

3216
TABLE I
CONCENTRIC TUBE PARAMETERS.

<table>
<thead>
<tr>
<th>Section 1</th>
<th>Section 1</th>
<th>Section 1</th>
<th>Section 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tube 1</td>
<td>Tube 2</td>
<td>Tube 3</td>
<td></td>
</tr>
<tr>
<td>Length (mm)</td>
<td>150</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Curvature (m⁻¹)</td>
<td>4.348</td>
<td>4.348</td>
<td>0.0</td>
</tr>
<tr>
<td>Relative Stiffness</td>
<td>1</td>
<td>1</td>
<td>0.286</td>
</tr>
</tbody>
</table>

(0, 0, d) in the robot coordinate system. Fitting a circle to the measured tip positions identifies the plane and the center of the circle. The Z and X axes of the robot coordinate system comprise the normal of the circle and the line from the circle center to the initial tip position, respectively. The origin of the robot coordinate system is located dmm away from the center along Z axis. The value of the parameter d is measured from the tube design.

The value of σ influenced the convergence speed of the adaptation and the errors during the adaptation. This value was found to provide fast convergence and was used for all experiments. The forgetting factor κ was set to unity leading to equal weighting of the entire measurement history. This is a reasonable choice for trajectories located in one region and represents the conservative limit of what can be achieved through adaptation.

A. Trajectory-based Model Adaptation

To evaluate adaptation in a specific region of the workspace, a periodic trajectory was generated corresponding to the robot tip tracing a square in space. The square was 40mm on a side, was perpendicular to the z axis and was centered at {0, 0, 210}. The desired tip orientation was specified as the constant value of {0, 0, 1}. The robot was commanded to trace out the square at a speed of 10mm/s.

Initially, model-adaptation was turned off and the error shown in Fig. 4(a) reflects the difference between actual robot kinematics and the algebraic model whose parameters have been fit to the mechanics-based model. Note that registration error could also be present here and would correspond to a rigid body displacement.

The adaptive estimator was then turned on and allowed to update the model as the square trajectory was traversed five times. The five traversals are plotted in Fig. 4(b). Notice that convergence to a steady-state path occurs almost immediately. To quantitatively examine convergence, the path error magnitude as a function of time is plotted in Fig. 5 with and without adaptation. Without adaptation, the error is a constant periodic function with a maximum error of 8mm, but with adaptation, maximum error over the perimeter never exceeds 3mm.

To examine the error from a geometric perspective, its in-plane and out-of-plane components are plotted in Fig. 6 as a function of perimeter coordinate. Again, notice that error reduction occurs so quickly that the transient is not visible.

B. Robustness to Tool Insertion and Removal

To evaluate the ability of the method to adapt to model perturbations caused, e.g., by the insertion and removal of tools through the robot lumen, additional experiments were
performed. To simulate tool delivery, a 0.8mm NiTi wire was inserted inside a 1.45mm outer diameter polyimide tube was used. This combination is similar to the system used to delivery and actuate the forceps shown in Fig. 1(a).

For these experiments, the robot tip was commanded to trace out a 40mm diameter circle at a velocity of 14mm/sec (40°/s). The circle was normal to the x axis and centered at \(-35,0,180\). The desired tip orientation was specified as the constant value of \((-\sin 60^\circ,0,\cos 60^\circ)\). In initial tests with this trajectory, it was observed that minimum error was achieved in 2 traversals of the circle. Consequently, results reported below correspond to those achieved after 2 adaptive traversals of the curve. Error distance is reported using a parameterization of in-plane radial distance, \(e_r\), and out-of-plane distance, \(e_p\).

The ability to adapt to a model change due to tool insertion and removal is illustrated Fig.7. Fig.7(a) illustrates initial model error prior to any adaptive tuning and prior to tool insertion (maximum error: \(e_r = 6.42\)mm, \(e_p = 5.41\)mm). The adaptive algorithm is then turned on and run for 2 traversals resulting in the converged path of Fig.7(b) (maximum error: \(e_r = 1.37\)mm, \(e_p = 1.66\)mm).

The adaptive algorithm was then turned off and the mock tool was next inserted in the robot lumen. The effect of inserting the straight tube and wire is to straighten the robot causing the path shown in Fig.7(c) to rise roughly vertically above the commanded path (maximum error: \(e_r = 2.94\)mm, \(e_p = 3.93\)mm). Allowing the adaptive algorithm to run for 2 traversals, however, eliminates much of this perturbation as shown in Fig.7(d) (maximum error: \(e_r = 1.61\)mm, \(e_p = 1.34\)mm).

When adaptation is turned off and the straight tube and wire are removed, robot curvature increases causing the path to shift downward as shown in Fig.7(e) (maximum error: \(e_r = 2.04\)mm, \(e_p = 3.49\)mm). Finally, adaptation is turned back on and the model converges again to the desired circle as shown in Fig.7(f) (maximum error: \(e_r = 1.52\)mm, \(e_p = 1.97\)mm).
This paper proposes a new direction in the control of concentric tube robots, namely, adaptive on-line model estimation. There are many reasons why such algorithms could prove to be an important component of any commercial concentric tube robot control system. For example, small variations in the geometric and mechanical properties of tube sets comprising any robot design will likely necessitate initial calibration of every tube set. Furthermore, significant modeling error can occur during operation due to both neglected mechanical phenomena (e.g., nonlinear elasticity) and perturbations, such as tool insertion and removal.

The approach described here exploits the linearity with respect to model parameters of the kinematic functional approximation proposed in [5]. This enables model adaptation to be implemented using recursive least squares. By selecting the order of Fourier series used in the functional approximation, this approach also provides a direct means to control the number of free parameters of the kinematic model.

**References**


