# 11

# Reactive Navigation for Continuum Manipulator in Unknown Environments

**Ahmad Ataka[1], Ali Shiva[1] and Kaspar Althoefer[2]**

[1]King's College London, London, United Kingdom
[2]Queen Mary University of London, London, United Kingdom

## Abstract

Navigating continuum robots, such as STIFF-FLOP manipulator [7], to reach a certain configuration without colliding with its surrounding environments tends to be an ongoing research area as the inherent complexity of this class of robots poses new challenges in this field compared to rigid-link robots. In this chapter, we describe the reactive navigation algorithm for guiding the tip of continuum robot toward the goal while, at the same time, avoiding the robot body from collision with the surrounding environment. We limit the navigation problem in this chapter to the case where the information of the environment is unknown before movement execution and the robot can rely only on the most updated sensory information. Two navigation algorithms, both inspired by the physical phenomena of electromagnetism, are implemented in the kinematic model of multisegment continuum manipulator. The pose estimation strategy needed to estimate the position of the manipulator body is also presented.

## 11.1 Introduction

Inspiration from certain species in nature such as the octopus [1] has sparked a new trend in robotics which aims at enhancing robotic maneuverability and dexterity compared to the conventional rigid-link robots, giving birth to the new breed of continuum robots. Many continuum robotic systems have been developed during the recent years [2–4] using a variety of designs both for the structure and for the actuation methods, such as tendon driven systems [5],

pneumatically actuated robots [6], or stiffness-controllable manipulators [4]. A recent example of this new type of robot has been presented in [7], where the STIFF-FLOP manipulator, built upon soft material, is designed to be used for a surgical purpose. Although developments in this field show potential improvement in the features which aim to resemble biology, yet the structural flexibility of this new class of robots poses new challenges in areas such as modeling, control, pose estimation, and navigation.

The subject of robot navigation continues to be the focus of many research works. Yet, with regard to continuum manipulators, research in this area is still in its early stages and therefore not fully conclusive. A common drawback witnessed in a majority of works based on optimization-based planning [1, 3, 4, 6, 8] and sampling-based planning [9–13] is the lack of capability in dealing with unknown obstacles. In [14], an inverse kinematic method is described for a steerable needle. However, due to the assumption of a static and well-defined environment, this method falls short of dealing with unknown obstacles. Some other studies which aim to tackle dynamic and unknown environments, although provide interesting solutions, are yet constrained to specific geometries and/or applications. For instance, an adaptive motion planning algorithm is discussed in [12] for the OctArm manipulator, but is limited to a planar scenario. The inverse-Jacobian method proposed in [15] specifically addresses guidance inside a tubular environment. In [16], the null space of a redundant continuum manipulator is exploited for navigating the robot. However, the method is not computationally cost efficient due to successive matrix multiplications.

One of the prominent trends in robot navigation is taking inspiration from natural physical phenomena such as the electric potential field [17]. Being relatively simple and straightforward, this method has been extensively applied for mobile robots, rigid manipulators, and even continuum manipulators [18, 19]. The global convergence is not guaranteed in this algorithm. However, since this method does not rely on the global map of the environment, it can be used in unknown and dynamic environments, and hence applied in a reactive manner.

Other physics-inspired works applied the properties of an artificially induced magnetic field for navigating rigid-link manipulators, such as [20, 21]. Although this method claims to resolve the problem of entrapment in local minima observed in the electric potential field case, this method still requires the geometry and location of obstacles to be known beforehand thus being impractical for unknown and/or dynamic obstacles. The effort to make the magnetic-field-inspired approach more reactive is set forward in [22]

by defining the concept of circular fields which can be applied to partially unknown environments. This method, however, requires a prior knowledge of the location of the center of the obstacles, which would not be possible to implement in real-world scenarios with unknown obstacles.

In this article, we present our recent works on continuum manipulator navigation which is inspired by the physical phenomena of electromagnetism. The use of electric field to keep manipulator body from collision is based on our works in [18]. Another method presented here is the use of magnetic field to avoid the collision, in which the backbone of the manipulator is considered as the means for current flow, resembling an electrical wire which induces a field on nearby objects. It assumes an artificial current flowing continuously along the segment of the continuum manipulator which induces "shadow" current on the surface of the nearby obstacle in such a way that the system behavior will resemble the physical behavior of the two current-carrying wires. Combined with attractive potential function at the tip and the pose estimation strategy to estimate the pose of any point along the body of the manipulator as presented in [19, 23], both methods will guide the manipulator's tip at a desired target location while at the same time prevent the manipulator body from colliding with unknown environment in a reactive manner.

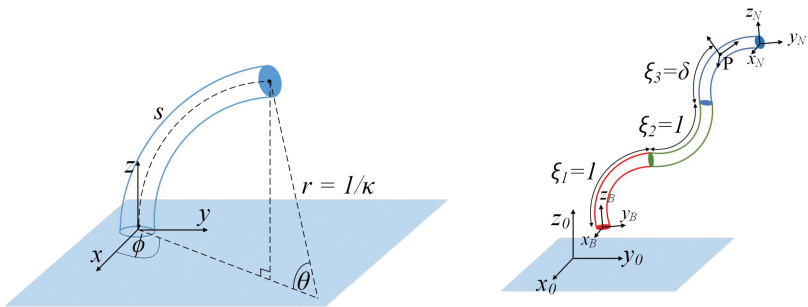
## 11.2 Modeling and Pose Estimation

### 11.2.1 Kinematic Model

The constant curvature kinematic model is used in this research. The underlying assumption is that each manipulator's segment can be mathematically described by the equation of circular arc with a constant radius of curvature. Therefore, every segment can be fully characterized by configuration space variables $\mathbf{k}_i = \begin{bmatrix} \kappa_i & \phi_i & s_i \end{bmatrix}^T$, denoting the curvature ($\kappa_i$), rotational deflection angle ($\phi_i$), and arc length ($s_i$) of segment-$i$, respectively, as illustrated in Figure 11.1a. The tip of segment-$i$ with respect to the base, ${}^{i}_{i-1}\mathbf{T}(\mathbf{k}_i) \in SE(3)$, will depend solely on variable $\mathbf{k}_i$ as explained at length in [11]. In this chapter, we assume that the manipulator's base is fixed. Hence, the pose of the end effector with respect to the world frame attached to the base of the first segment for manipulator with $N$ segments is described as

$$ {}^{N}_{0}\mathbf{T}(\mathbf{k}) = \prod_{i=1}^{N} {}^{i}_{i-1}\mathbf{T}(\mathbf{k}_i), \tag{11.1} $$

where $\mathbf{k} = \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \dots & \mathbf{k}_N \end{bmatrix}^T$.




(a) (b)

**Figure 11.1** (a) The frame definition attached to each individual segment. (b) Illustration of a three -segment continuum manipulator with a movable base.

These configuration space variables $\mathbf{k}$ are normally different from the real controllable parameters of the actuator and generally very robot-specific due to the wide range of choices in design and actuating mechanism. For a tendon-driven robot like the one used in [18], the actuator space variables are the tendon length and can be written as $\mathbf{q}_i = \begin{bmatrix} l_{i1} & l_{i2} & l_{i3} \end{bmatrix}^T$ in which $l_{ij}$ denotes the length deviation of tendon-$j$ in segment-$i$ from normal length $L$. For $N$-segment manipulator, the whole actuator space is expressed as $\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_N \end{bmatrix}^T$. The mapping between configuration space variables of segment-$i$, $\mathbf{k}_i$, and tendon length in segment-$i$, $\mathbf{q}_i$, depends on the manipulator's cross-section radius $d$ as has been derived in [11].

To help expressing the pose of any point along the body of the manipulator, the scalar coefficient vector is employed as illustrated in Figure 11.1b. To specify a point in segment-$i$, a continuous scalar $\xi_i \in [0,1]$ is used which covers all the point from the base ($\xi_i = 0$) to the tip ($\xi_i = 1$) of the segment. The set of scalars from all segments constitutes a vector, $\xi = \begin{bmatrix} \xi_1 & \xi_2 & \dots & \xi_N \end{bmatrix}^T$ defined as $\xi = \{\xi_r = 1 : \forall r < i, \xi_i, \xi_r = 0 : \forall r > i\}$ [16].

Using the above formulation, the forward kinematic is written as

$$ {}^{N}_{0}\mathbf{T}(\mathbf{q},\xi) = \begin{bmatrix} \mathbf{R}(\mathbf{q},\xi) & \mathbf{p}(\mathbf{q},\xi) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{11.2} $$

where $\mathbf{R}(\mathbf{q},\xi) \in SO(3)$ and $\mathbf{p}(\mathbf{q},\xi) \in \mathbb{R}^3$, respectively, express the orientation and position of any point along the body of the manipulator.

A Jacobian matrix, $\mathbf{J}(\mathbf{q},\xi) \in \mathbb{R}^{3\times(3N)} = \frac{\partial \mathbf{p}(\mathbf{q},\xi)}{\partial \mathbf{q}}$, transforms the velocity vector as follows

$$\dot{\mathbf{p}}(\mathbf{q},\xi) = \mathbf{J}(\mathbf{q},\xi)\dot{\mathbf{q}} \Leftrightarrow \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q},\xi)^{-1}\dot{\mathbf{p}}(\mathbf{q},\xi). \tag{11.3}$$

Another way of describing the kinematic model of continuum manipulator is by using the state space representation described by the following equations

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k,\mathbf{u}_k), \tag{11.4}$$

$$\mathbf{y}_k = g(\mathbf{x}_k), \tag{11.5}$$

where $\mathbf{x}_k \in X$, $\mathbf{u}_k \in U$, and $\mathbf{y}_k \in Y$, respectively, denote the state variable, input, and output value at iteration-$k$, $X$, $U$, and $Y$, respectively, denote the state space, input space, and output space, while $f : X \times U \to X$ and $g : X \to Y$, respectively, map the current state and input value to the next state and the current state to the output value. Here we define the term *active segments* to denote $n \leq N$ number of segments which will be analyzed using state space representation.

To transform the kinematic model into a state space equation, the tendon's actuator space $\mathbf{q} \in \mathbb{R}^{3n}$ is chosen as state variables $\mathbf{x}$ while the tendon length's rate of change $\dot{\mathbf{q}} \in \mathbb{R}^{3n}$, proportional to the rotational speed of the motor to which the tendon is connected, is the input $\mathbf{u}$. The state and input variables are expressed as

$$\mathbf{x} = \mathbf{q} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_n\end{bmatrix}^T, \tag{11.6}$$

$$\mathbf{u} = \dot{\mathbf{q}} = \begin{bmatrix}\dot{\mathbf{q}}_1 & \dot{\mathbf{q}}_2 & \dots & \dot{\mathbf{q}}_n\end{bmatrix}^T, \tag{11.7}$$

in which $\mathbf{q}_i = \begin{bmatrix}l_{i1} & l_{i2} & l_{i3}\end{bmatrix}^T$. Therefore, using this definition, the state equation in Equation (11.4) describing the kinematic model of continuum manipulator is written as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k,\mathbf{u}_k) = \mathbf{x}_k + \Delta t\,\mathbf{u}_k, \tag{11.8}$$

where the time sampling is denoted by $\Delta t$.

The output variable $\mathbf{y} \in \mathbb{R}^{3n}$ in Equation (11.5) is represented by the tip position of each active segment retrieved from a 3-DOF position sensor embedded in the tip of each segment.

$$\mathbf{y}_k = g(\mathbf{x}_k) = \begin{bmatrix}\mathbf{p}(\mathbf{x}_k,\xi=\chi_1) & \dots & \mathbf{p}(\mathbf{x}_k,\xi=\chi_n)\end{bmatrix}^T, \tag{11.9}$$

in which $\chi_i = \begin{bmatrix}\xi_1 = 1 & \dots & \xi_i = 1 & \xi_{i+1} = 0 & \dots & \xi_n = 0\end{bmatrix}^T$ while $\mathbf{p}(\mathbf{x}_k,\xi)$ is taken from Equation (11.2).

### 11.2.2 Pose Estimation

In order to avoid the manipulator body from collision, the pose of any point along the body of the robot needs to be known at every iteration. This is not directly possible without the sensing capability of each segment's tendon length. The state space representation of the continuum robot model described in the previous subsection is beneficial to estimate the pose of any point along the body of the manipulator with only position sensor available embedded in the tip of each segment.

The well-known extended Kalman filter (EKF) is modified to estimate the tendon's length of each segment using the kinematic model and the measured position data of the tip of each segment. The state space equations expressed in Equations (11.8) and (11.9) enable us to estimate the next tendon length value $\hat{\mathbf{x}}_{k+1|k+1}$ based on the current estimate $\hat{\mathbf{x}}_{k|k}$, input $\mathbf{u}_k$, and sensor data $\mathbf{y}_k$ using the information of the estimation covariance $\mathbf{P}_{k|k}$, process noise variance $\mathbf{Q}_k$, measurement noise variance $\mathbf{R}_k$, and linearized model of state Equations (11.8) and (11.9) characterized by matrix $\mathbf{A}_k = \frac{\partial f(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k}$ and $\mathbf{C}_k = \frac{\partial g(\mathbf{x}_k)}{\partial \mathbf{x}_k}$.

As explained in more detail in our previous work [23], implementing the standard EKF could make the state estimate arrives at physically impossible state values, for instance, a negative tendon length, caused by the multiple mathematically possible states for a certain segment tip position. Therefore, a multistage implementation of EKF, as illustrated in Figure 11.2, is employed. The underlying principle of this approach is by taking a segment into the estimation process only when the tip pose estimation error of the previous segment is smaller than a threshold $\delta$. The full description of the algorithm

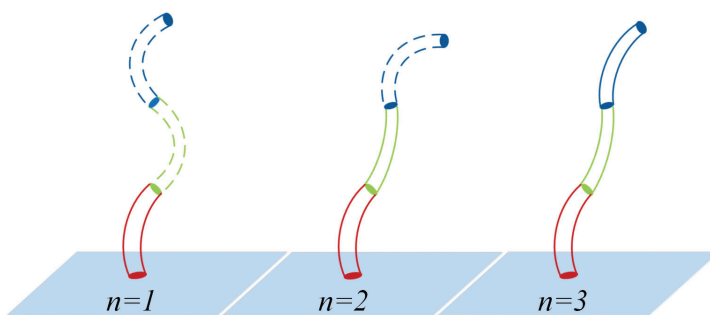



**Figure 11.2** Illustration of the multistage extended Kalman filter for the case of three-segment continuum manipulator.

is shown in Algorithm 1 and described at length in [23]. The state estimate which represents the estimated tendon length $\hat{\mathbf{x}}$ produced by EKF can then be used in Equation (11.2) to determine the estimated pose of every point along the manipulator body needed by the navigation algorithm which will be explained in the next section.

## 11.3 Reactive Navigation

In this section, we present the reactive navigation algorithm inspired by the phenomena of electromagnetism. The underlying principle of this approach is that the manipulator moves in space filled with an artificial vector field $\mathbf{F}(\mathbf{q})$ designed to make the manipulator behave in an intended way to avoid collision with surrounding obstacle. Besides that, the end effector of the manipulator will also be influenced by an artificial attractive vector field toward the intended goal position.

**Algorithm 1** Multistage pose estimation.

$n \leftarrow 1, k \leftarrow 0$
**loop**
  $\mathbf{q}_n \leftarrow$ InitState()
  $\hat{\mathbf{x}}_{k|k} \leftarrow$ Add($\mathbf{q}_n$)
  $(\mathbf{P}_{k|k}, \mathbf{Q}, \mathbf{R}) \leftarrow$ Initialize($n$)
  **loop**
    $\mathbf{u}_k \leftarrow$ GetInputSignal($n$)
    $\mathbf{y}_k \leftarrow$ GetSensorData($n$)
    $\mathbf{A}_k \leftarrow \frac{\partial f(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k}$
    $\mathbf{C}_k \leftarrow \frac{\partial g(\mathbf{x}_k)}{\partial \mathbf{x}_k}$
    $(\hat{\mathbf{x}}_{k+1|k+1}, \mathbf{P}_{k+1|k+1}) \leftarrow$ EKF($\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}, \mathbf{u}_k, \mathbf{y}_k, \mathbf{A}_k, \mathbf{C}_k$)
    $\hat{\mathbf{y}}_{k|k} \leftarrow$ Equation (11.9)
    $\mathbf{e} \leftarrow$ DetermineError($\hat{\mathbf{y}}_{k|k}, \mathbf{y}_k$)
    $k \leftarrow k + 1$
    **if** norm($\mathbf{e}$) $< \delta$ **and** $n < N$ **then**
      **break**
    **end if**
  **end loop**
  $n \leftarrow n + 1$
**end loop**

A quadratic potential analogous with the spring-mass system is used to guide the end effector to a desired configuration $\mathbf{q}_d$ as follows

$$U_d(\mathbf{q}) = \frac{1}{2}k(\mathbf{q}-\mathbf{q}_d)^T(\mathbf{q}-\mathbf{q}_d), \tag{11.10}$$

where $k$ stands for a positive constant. The use of the kinematic model of continuum manipulator leads to the modification of a generalized vector field $\mathbf{F}$ into the task-space velocity of the manipulator $\dot{\mathbf{p}}$ as follows

$$\dot{\mathbf{p}}_{\mathbf{p}_d} = -\nabla U_d(\mathbf{p}) = -k(\mathbf{p}-\mathbf{p}_d). \tag{11.11}$$

This task-space velocity $\dot{\mathbf{p}}$ can then be mapped into actuator-space velocity $\dot{\mathbf{q}}$ via Equation (11.3). The vector field used to ensure the manipulator body safe from collision will be explained in more detail.

### 11.3.1 Electric-field-based Navigation

In the first part of this section, we present the modified version of the classical electric-field-based navigation based on our previous works in [18] to do obstacle avoidance. Each obstacle will produce a repulsive vector field as follows

$$\dot{\mathbf{p}}_{\mathscr{O}} = \begin{cases} \eta\left(\frac{1}{\rho}-\frac{1}{\rho_0}\right)\frac{1}{\rho^2}\frac{\partial\rho}{\partial\mathbf{p}} & \text{if } \rho < \rho_0 \\ 0 & \text{if } \rho \geq \rho_0 \end{cases}. \tag{11.12}$$

$\rho = \sqrt{(\mathbf{p}-\mathbf{p}_{\mathscr{O}})^T(\mathbf{p}-\mathbf{p}_{\mathscr{O}})}$ represents the closest distance between the point on manipulator's body and the obstacle surface, while $\eta$ and $\rho_0$ stand for positive constant and the limit distance of influence, respectively.

Several points along the manipulator body, called "point subjected to potentials" (PSP), are chosen to be a working point of the proposed vector field to ensure that the body of the manipulator is safe from collision. The pose of a PSP in segment-$i$ is given by $\mathbf{p}(\mathbf{q}, \xi = \lambda_i)$, where

$$\lambda_i = \begin{bmatrix} \xi_1 = 1 & \dots & \xi_i \in [0,1] & \xi_{i+1} = 0 & \dots & \xi_N = 0 \end{bmatrix}^T. \tag{11.13}$$

It is noted that only the closest pair of obstacle-PSP will contribute to the motion at a time. Only manipulator's end effector will be influenced by the combined attractive and repulsive vector fields while the rest of the body is influenced only by the obstacle repulsion. The task-space velocities of the end effector and the closest PSP to the obstacle are then transformed to the

actuator space, manifested as an input signal for the kinematic model and the pose estimation, $\mathbf{u}_k$.

$$\mathbf{u}_k = \dot{\mathbf{q}} = \mathbf{J}_e^{+}\dot{\mathbf{p}}_{\mathbf{p}_d} + \mathbf{J}_a^{+}\dot{\mathbf{p}}_{\mathscr{O}}. \tag{11.14}$$

Here the terms $\mathbf{J}_e$ and $\mathbf{J}_a$, respectively, represent the end-effector's Jacobian and the PSP Jacobian, $\dot{\mathbf{p}}_{\mathscr{O}}$ is a repulsive vector field generated by the closest obstacle to the PSP, while the $(^{+})$ operation is defined as $\mathbf{J}^{+} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1}$ [4].

### 11.3.2 Magnetic-field-based Navigation

One of the fundamental formulas on the magnetic field generation is the Biot-Savart equation describing the magnetic field generated by a current-carrying wire. An electrical current $i_o$ flowing on the wire of infinitesimal length $\mathbf{dl}_o$ as illustrated in Figure 11.3.2 will generate an infinitesimal magnetic field $\mathbf{dB}$ given by the following equation [25]

$$\mathbf{dB} = \frac{\mu_0}{4\pi}\frac{i_o \mathbf{dl}_o \times \mathbf{r}}{|\mathbf{r}|^3}. \tag{11.15}$$

Here, $\mathbf{r}$, $\mu_0$, and $\times$, respectively, stand for the position of a point with respect to the wire, the permeability constant, and the vector cross-product operation. This magnetic field will produce a force $\mathbf{dF}$ on any other current-carrying wire with an infinitesimal length $\mathbf{dl}_a$ and current $i_a$, as depicted in Figure 11.3.2, as described by the following equation

$$\mathbf{dF} = i_a \mathbf{dl}_a \times \mathbf{B}. \tag{11.16}$$

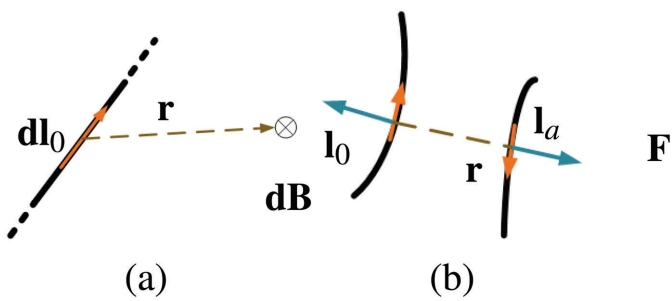



**Figure 11.3** (a) An illustration of current-carrying wire with electrical current flows in the direction of $\mathbf{dl}_o$ inducing magnetic field $\mathbf{dB}$ which points inside toward the paper. (b) Two current-carrying wires with currents flow opposite to each other, $\mathbf{l}_a$ and $\mathbf{l}_o$, will produce repulsion force $\mathbf{F}$ on both wires.

From the previous equations, we can then derive the force interaction between two current-carrying wires. After some simplification and dropping all the infinitesimal notation, we can write down the interaction force as follows

$$\mathbf{F} = \frac{\mu_0 i_a i_o}{4\pi} \ \frac{\mathbf{l}_a \times (\mathbf{l}_o \times \mathbf{r})}{|\mathbf{r}|^3}. \tag{11.17}$$

Inspired by this physical phenomena, we can see continuum manipulator as a current-carrying wire with current direction $\mathbf{l}_a$ which will induce artificial current vector $\mathbf{l}_o$ on the obstacle surface located at position $\mathbf{r}_o$ with respect to the manipulator. This induced current on the obstacle will produce magnetic field $\mathbf{B}$ which will produce force on the robot current in such a way that the force $\mathbf{F}$ will avoid the robot body from colliding with the obstacle. Using the fact that $\mathbf{r}_o = -\mathbf{r}$ by definition, Equation (11.17) can be expressed as follows

$$\mathbf{F}(\mathbf{r}_o) = c \ \ \mathbf{l}_a \times (\mathbf{r}_o \times \mathbf{l}_o) \ \ f(|\mathbf{r}_o|). \tag{11.18}$$

Here, $c > 0$ and $f(|\mathbf{r}_o|) \geq 0$ denote a scalar constant and positive scalar function, respectively. To simplify the cross-product operation, we introduce a skew-symmetric matrix $\hat{\mathbf{l}}$ as a substitution to the vector cross-product operation $\mathbf{l}\times$ of vector $\mathbf{l} = \begin{bmatrix} l_x & l_y & l_z \end{bmatrix}^T$ defined as

$$\hat{\mathbf{l}} = \begin{bmatrix} 0 & -l_z & l_y \\ l_z & 0 & -l_x \\ -l_y & l_x & 0 \end{bmatrix}. \tag{11.19}$$

The magnetic-field-based navigation extends the formulation presented in Equation (11.18). An artificial current $i_a$ is assumed to flow continuously along the body of the continuum manipulators from the base to the tip. At every point on the robot's body, the current direction, specified by $\mathbf{l}_a$, is always tangential to the robot's curvature as illustrated in Figure 11.4. Once the parts of the robot body are close enough to the obstacle, the closest obstacle point to the robot will induce a shadow current $i_o$ with the current direction $\mathbf{l}_o$. We designed this current direction $\mathbf{l}_o$ in such a way that the interaction force results in a repulsive behavior. For that purpose, we take inspiration from the repulsion phenomena observed in the case of two parallel wires carrying current in the opposite direction as shown in Figure 11.3.2. Hence, the current direction on the obstacle is defined as

$$\mathbf{l}_o = -\mathbf{l}_a. \tag{11.20}$$

The characteristic of the produced force will be described further in the following lemmas.

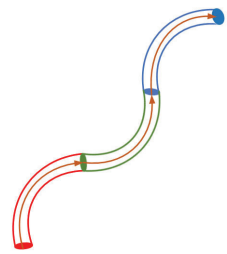


**Figure 11.4** The definition of the current direction $\mathbf{l}_a$, drew as orange arrows, on the manipulator's body.

**Lemma 1.** *For every obstacle point with current direction* $\mathbf{l}_o$ *as defined in Equation (11.20) and located at position* $\mathbf{r}_o$ *with respect to the robot, vector* $\mathbf{l}_o$, $\mathbf{l}_a$, *and* $\mathbf{r}_o$ *are coplanar.*

*Proof.* Three vectors are coplanar if and only if they are not linearly independent, i.e., we can write down one vector as a linear combination of two other vectors. We can write down

$$\mathbf{l}_o = a\ \mathbf{l}_a + b\ \mathbf{r}_o, \tag{11.21}$$

where $a$ and $b$ are scalar. If we choose $a = -1$ and $b = 0$, the above equation will simply become Equation (11.20), thus proofing the lemma. □

**Lemma 2.** *For every obstacle point with current direction* $\mathbf{l}_o$ *as defined in Equation (11.20) and located at position* $\mathbf{r}_o$ *with respect to the robot, the produced force defined in Equation (11.18) will always be repulsive except when* $\mathbf{l}_o$, $\mathbf{l}_a$, *and* $\mathbf{r}_o$ *lie on the same line.*

*Proof.* Lemma 1 concludes that $\mathbf{l}_o$, $\mathbf{l}_a$, and $\mathbf{r}_o$ are coplanar; hence, each vector can be described using a pair of unit vectors of this plane: $\mathbf{l}_a = \begin{bmatrix} l_{ax} & l_{ay} \end{bmatrix}^T$, $\mathbf{r}_o = \begin{bmatrix} r_{ox} & r_{oy} \end{bmatrix}^T$. Assuming $cf(|\mathbf{r}_o|) = 1$, the produced force on the robot $\mathbf{F}$ can be derived from Equations (11.18) to (11.20) as

$$\mathbf{F} = \begin{bmatrix} -l_{ay}^2 r_{ox} + l_{ax} l_{ay} r_{oy} \\ -l_{ax}^2 r_{oy} + l_{ax} l_{ay} r_{ox} \end{bmatrix}. \tag{11.22}$$

A dot-product between the force $\mathbf{F}$ and position vector of obstacle $\mathbf{r}_o$ can be simplified as follows

$$\mathbf{F}^T \mathbf{r}_o = 2 l_{ax} l_{ay} r_{ox} r_{oy} - l_{ay}^2 r_{ox}^2 - l_{ax}^2 r_{oy}^2 = -(l_{ay} r_{ox} - l_{ax} r_{oy})^2. \tag{11.23}$$

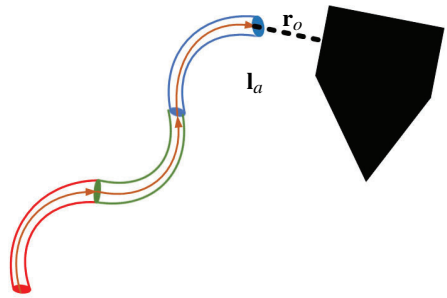



**Figure 11.5** Configuration where the standard continuous current algorithm fails, i.e., when the current $\mathbf{l}_a$ and vector $\mathbf{r}_o$ toward polygonal obstacle illustrated as a black object lie on the same line.

From Equation (11.23), it can be concluded that $\mathbf{F}^T\mathbf{r}_o \leq 0$. This implies that in the case of $\mathbf{F}^T\mathbf{r}_o < 0$, the force $\mathbf{F}$ will always have component in the opposite direction of $\mathbf{r}_o$, thus, producing a repulsive behavior (away from the obstacle). The exception occurs when $\mathbf{F}^T\mathbf{r}_o = -l_{ay}r_{ox} + l_{ax}r_{oy} = 0$, which is the implication of $\hat{\mathbf{l}}_a\mathbf{r}_o = \mathbf{0}$. It means that $\mathbf{r}_o$ is in the same or opposite direction of $\mathbf{l}_a$, which will occur only when $\mathbf{l}_a$ (and consequently $\mathbf{l}_o$) lies on the same line as $\mathbf{r}_o$. □

**Lemma 3.** *The produced force* $\mathbf{F}$ *in Equation (11.18) has no component along the line tangential to the continuum manipulator's curvature.*

*Proof.* Suppose we define $\mathbf{B}$ as follows

$$\mathbf{B} = c(\mathbf{r}_o \times \mathbf{l}_o) \;\; f(|\mathbf{r}_o|), \tag{11.24}$$

the dot product between $\mathbf{l}_a$ and $\mathbf{F}$ can be written as

$$\mathbf{l}_a^T\mathbf{F} = \mathbf{l}_a^T\hat{\mathbf{l}}_a\mathbf{B}. \tag{11.25}$$

From the definition of the skew-symmetric matrix in Equation (11.19), it can be shown that $\mathbf{l}_a^T\hat{\mathbf{l}} = \mathbf{0}^T$ and thus it produces a zero product between $\mathbf{l}_a$ and $\mathbf{F}$, meaning that the two vectors are perpendicular. By choosing $\mathbf{l}_a$ to flow along the curvature of the continuum manipulator, it can be concluded that the force $\mathbf{F}$ will have no component in this direction. □

This proposed algorithm is advantageous for the continuum manipulator due to the fact that it has no force component in the direction tangential to the manipulators curvature, as has been proven in Lemma 3. Hence, it is suitable

for manipulators with more dominant bending than extension ability, like the case for most of the current designs [10]. Compared to the works proposed in [18], where several points along the backbone of the manipulator are used as subject to the repulsive field, this method is superior since it will have little contribution to the manipulator's extension, and thus, help avoiding length constraint in such a manipulator.

However, this method as it also has a drawback. As mentioned in Lemma 2, it has zero force for the case when the robot current $\mathbf{l}_a$ and the obstacle relative position vector $\mathbf{r}_o$ lie on the same line. This will happen when an obstacle is located at exactly above the distal tip, as illustrated in Figure 11.5. Although this practically never happens in reality, in order to make sure that it is solved, we can add a special condition when $\hat{\mathbf{l}}_a \mathbf{r}_o = \mathbf{0}$. In this case, the current direction $\mathbf{l}_o$ can be chosen as any $90^\circ$ rotation of robot current direction $\mathbf{l}_a$.

Finally, the scalar function $f(|\mathbf{r}_o|)$ can be chosen to have the following form

$$f(|\mathbf{r}_o|) = \begin{cases} \left(\dfrac{1}{|\mathbf{r}_o|} - \dfrac{1}{r_b}\right) \dfrac{1}{|\mathbf{r}_o|^2} & \text{if } |\mathbf{r}_o| < r_b \\ 0 & \text{if } |\mathbf{r}_o| \geq r_b \end{cases}, \tag{11.26}$$

where $r_b$ specifies a limit distance of the potential influence. Hence, the proposed method has been fully described.

### 11.3.3 The Complete Algorithm

The pose estimation stage is combined to the navigation stage as illustrated in Figure 11.6. Here, the pose estimation stage provides information on the end-effector pose estimate $\mathbf{p}(\mathbf{x}_k, \xi = \chi_N)$ along with the pose estimate of closest point on the robot body to the obstacle $\mathbf{p}(\mathbf{x}_k, \xi = \lambda_i)$ at every iteration for the navigation purpose. This is done by applying the forward kinematics equation to the state estimate from the EKF as follows

$$\hat{\mathbf{p}}_k(\xi) = \mathbf{p}(\hat{\mathbf{x}}_{k|k}, \xi). \tag{11.27}$$

The produced control signal $\mathbf{u}_k$ from the navigation stage is then used as an input to the estimation stage to estimate the state value for the next iteration. Lastly, as explained in more detail in our previous work [23], due to the early error of the estimation stage, the navigation stage will be activated only when a certain condition relating to the tip position estimation errors for all segments is fulfilled. In short, we have combined the pose estimation and

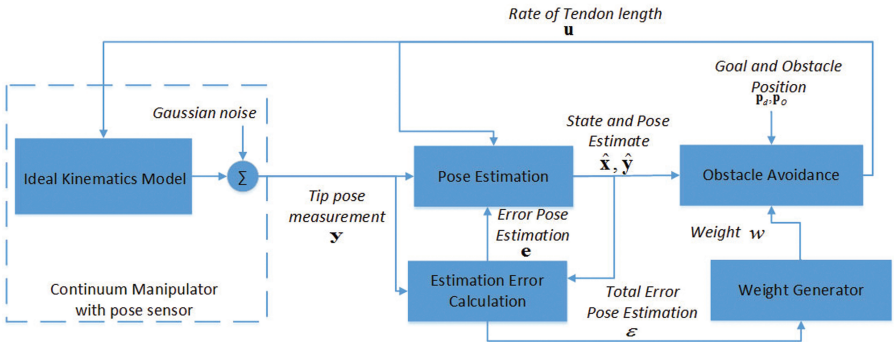



**Figure 11.6** The way the pose estimator is combined with the obstacle avoidance algorithm.

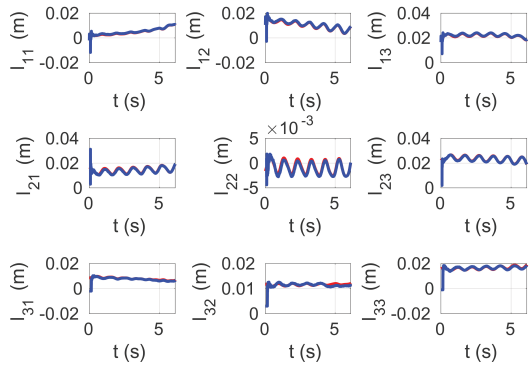
navigation algorithm to ensure collision-free movement of the manipulator's body.

## 11.4 Results and Discussion

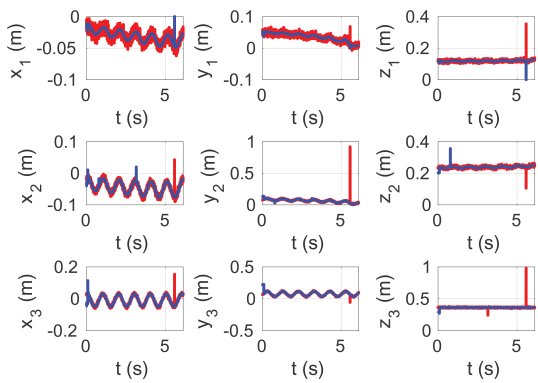
Some simulation results to confirm the performance of the algorithm are presented in this section. All presented simulations use the Robot Operating System (ROS) framework. The model of a three-segment continuum manipulator was used to test the algorithm. Five points per segment, distributed uniformly along the backbone of each segment, are chosen to be the working point of the navigation field. A spherical obstacle with varying dimension and behavior is placed in the vicinity of the manipulator to test the proposed method. For the pose estimation stage, the measurement data of the tip pose $\mathbf{y}_k$ of each segment are generated via ideal kinematic model added with Gaussian noise to replace a 3-DOF position tracker assumed to be embedded in the tip of each segment. A more detail description on the value of parameters used in the pose estimation and electric-field-based navigation can be referred back to our work in [23] and [18].

The first part of simulation shows the performance of the pose estimation stage in the absence of obstacle. In this case, a circular trajectory is used as a reference path for the end effector. We can see from Figures 11.7a and 11.7b how the multistage EKF is able to cope with the varying input signal and sensory data and successfully track the real state value.

In the second part of simulation, a moving spherical obstacle with varying size is assumed to move in the proximity of the manipulator body. For all




(a)




(b)

**Figure 11.7** The performance of the pose estimation when the manipulator's end effector is given a circular trajectory shows (a) the true state (red line) and the estimated state (blue line) comparison and (b) the true pose with Gaussian noise (red line) and the estimated pose (blue line) comparison, respectively.

figures, the order of the subfigures is from the upper left picture to the upper right picture and then the lower left picture to the lower right picture.

We first implemented the electric-field-based navigation. Here, we assume two cases: the one in which the obstacle is close to the bottom segment drawn in red as shown in Figure 11.8 and the one in which the obstacle is close to the middle segment drawn in green as shown in Figure 11.9.

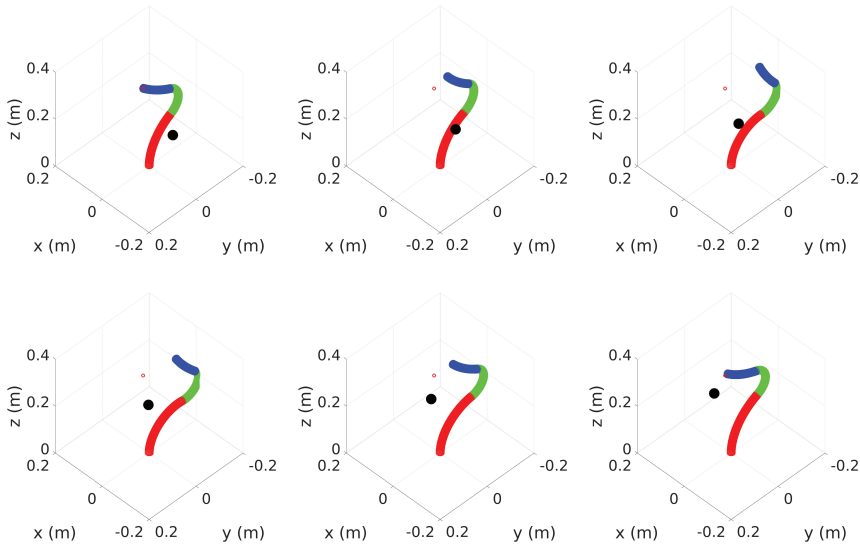



**Figure 11.8** The movement of a three-segment continuum manipulator avoiding moving obstacle drawn in black in the proximity of the bottom segment drawn in red. The order of subfigures is from the upper left picture to the upper right picture and then the lower left picture to the lower right picture.

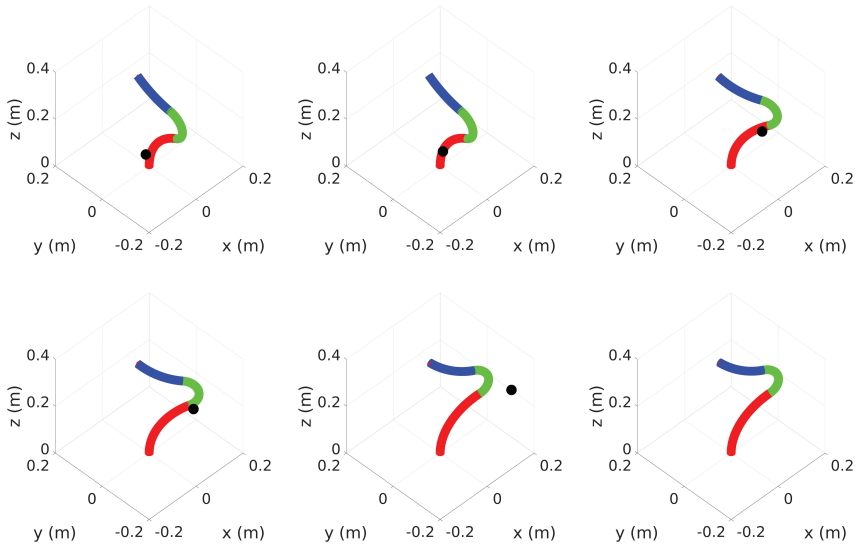



**Figure 11.9** The movement of a three-segment continuum manipulator avoiding moving obstacle drawn in black in the proximity of the middle segment drawn in green. The order of subfigures is from the upper left picture to the upper right picture, and then the lower left picture to the lower right picture.

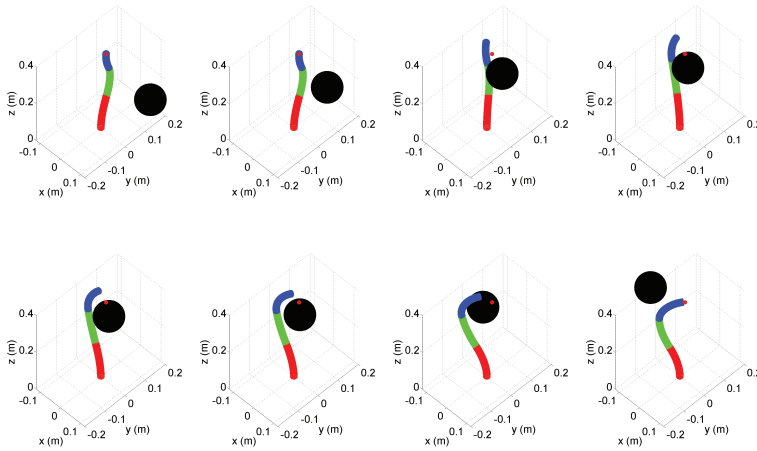



**Figure 11.10** Simulation of the magnetic-field-based navigation working to avoid moving obstacle drew in black sphere. The order of movement is from the upper left to the upper right and then the lower left to the lower right.

The results demonstrate how the combined pose estimation and electric-field-based navigation is able to make the body of the manipulator safe from colliding with unknown and even dynamic spherical obstacle.

We then implement the magnetic-field-based navigation to the robot model. In the first scenario, shown in Figure 11.10, the desired end-effector position is fixed while the obstacle, with 5 cm radius, is able to move in the vicinity of the manipulator's body. While, in the second scenario, shown in Figure 11.11, the obstacle, with 3 cm radius, is made to be fixed while the end effector starts from a position located at some distance from the target point. After the end effector reaches the target, the target's position is moved immediately so that the manipulator starts to move to the new target position.

In Figure 11.10, we can see that the spherical obstacle drew in black moves toward the manipulator's body. The force produced by the magnetic interaction of current flowing along the backbone of the manipulator and shadow current flowing on the surface of the obstacle produces a repulsive behavior which makes the manipulator body moves away from the approaching obstacle. We can see that the closest point on the manipulator to the obstacle tends to move in the direction opposite of the vector connecting the robot and the obstacle, as has been proven in Lemma 2. The figure also shows how the manipulator behaves in such a way that the bending

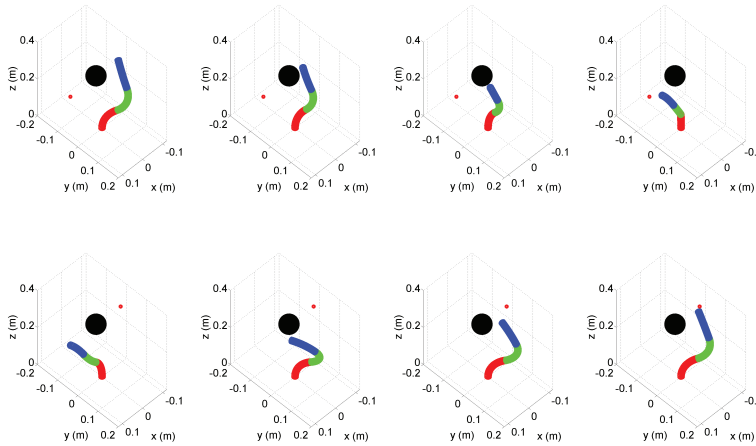



**Figure 11.11** Simulation of the magnetic-field-based navigation working to avoid static obstacle (black sphere) while moving toward the goal (red dot). The order of movement is from the upper left to the upper right and then the lower left to the lower right.

movement is more dominant than the extension, as can be observed that the segments' lengths do not change a lot during the whole movement. This has also been proven in Lemma 3 and becomes a major advantage of this method to be implemented for the continuum manipulator compared to the standard potential field. Figure 11.11 shows how the manipulator can avoid the static obstacle on the way toward the target (shown in the upper figures). This is also the case when the target is moved back to the end-effector's initial position (shown in the lower figures).

### 11.4.1 Discussion

From the simulation results, it can be seen that in general both the electric-field-based navigation and the magnetic-field-based navigation work successfully in avoiding the obstacle. The implementation of the multistage EKF as a pose estimator is useful to estimate the pose of any point along the manipulator's body such that collision with surrounding the obstacle can be avoided. The reactive nature of the navigation algorithm also enables the manipulator to avoid collision with dynamically moving and unknown obstacle without prior information of the surrounding environment. The magnetic-field-based navigation is also superior due to the fact that it is designed to comply with the continuum manipulator's limitation especially on its extension capability.

However, the repulsive nature of the algorithm makes it still susceptible to local minima when the manipulator moves in a complex environment. This problem is left to be investigated further in future works.

## 11.5 Conclusion

In this article, we present our recent works on reactive navigation for the case of the continuum manipulator. The navigation problem we are trying to deal with in this chapter is that in which the information of the environment is unknown prior to the robot movement, which makes the robot have to rely on local sensory information and response in a reactive manner. The two presented algorithms, one inspired by the electric field phenomena while the other inspired by the magnetic field, are shown to successfully avoid the continuum manipulator body from colliding with the environment. The magnetic-field-based navigation is shown to be more suitable for navigating the class of the continuum manipulator robot due to the fact that it mainly works to bend, rather than extend, the segment of manipulator. In the future, the practical implementation of the algorithm will be explored along with the effort to overcome the current limitations. Implementation of the method on other continuum manipulators such as a snake-like robot, or a more complex dynamic model of the continuum manipulator will also be investigated.

## Acknowledgment

Research is partially supported by King's College London, the STIFF-FLOP project grant from the European Commission Seventh Framework Programme under grant agreement 287728, the Four By Three grant from the European Framework Programme for Research and Innovation Horizon 2020 under grant agreement no 637095, and the Indonesia Endowment Fund for Education, Ministry of Finance Republic of Indonesia.

## References

[1] McMahan, W., Chitrakaran, V., Csencsits, M., Dawson, D., Walker, I. D., Jones, B. A., et al. (2006). "Field trials and testing of the OctArm continuum manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2336–2341.

[2] Kim, Y.-J., Cheng, S., Kim, S., and Iagnemma, K. (2013). "A novel layer jamming mechanism with tunable stiffness capability for minimally invasive surgery," in *Proceedings of the IEEE Transactions on Robotics*, 29, 1031–1042.

[3] Lyons, L. A., Webster, R. J., and Alterovitz, R. (2009). "Motion planning for active cannulas," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 801–806.

[4] Maghooa, F., Stilli, A., Noh, Y., Althoefer, K., and Wurdemann, H. A. (2015). "Tendon and pressure actuation for a bio-inspired manipulator based on an antagonistic principle," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA),* 2556–2561.

[5] Qi, P., Qiu, C., Liu, H., Dai, J. S., Seneviratne, L., and Althoefer, K. (2014). "A novel continuum-style robot with multilayer compliant modules," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3175–3180.

[6] Mahl, T., Hildebrandt, A., and Sawodny, O. (2014). "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," in *Proceedings of the IEEE Transactions on Robotics*, 30, 935–949.

[7] Cianchetti, M., Ranzani, T., Gerboni, G., De Falco, I., Laschi, C., and A. Menciassi. (2013). "STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3576–3581.

[8] Niu, G., Zheng, Z., and Gao, Q. (2014). "Collision free path planning based on region clipping for aircraft fuel tank inspection robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 3227–3233.

[9] Torres, L. G. and Alterovitz, R. (2011). "Motion planning for concentric tube robots using mechanics-based models," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5153–5159.

[10] Walker, I. D., Carreras, C., McDonnell, R., and Grimes, G. (2006). Extension versus bending for continuum robots. *Int. J. Adv. Robot. Syst.* 3, 171–178.

[11] Webster, R. J., III, and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *Int. J. Rob. Res.* 29, 1661–1683.

[12] Xiao, J., and Vatcha, R. (2010). "Real-time adaptive motion planning for a continuum manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5919–5926.

[13] Xu, J., Duindam, V., Alterovitz, R., and Goldberg, K. (2008). "Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring random trees and backchaining," in *Proceedings of the IEEE international conference automation science engineering*, 41–46.

[14] Duindam, V., Alterovitz, R., Sastry, S., and Goldberg, K. (2008). "Screw-based motion planning for bevel-tip flexible needles in 3D environments with obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2483–2488.

[15] Chen, G., Pham, M. T., and Redarce, T. (2009). Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy. *Robot. Auton. Syst.* 57, 712–722.

[16] Godage, I. S., Branson, D. T., Guglielmino, E., and Caldwell, D. G. (2012). "Path planning for multisection continuum arms," in *Proceedings of the International Conference on Mechatronics and Automation (ICMA)*, 1208–1213.

[17] Khatib, O. (1985). "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2, 500–505.

[18] Ataka, A., Qi, P., Liu, H., and Althoefer, K. (2016). "Real-time planner for multi-segment continuum manipulator in dynamic environments," in *Proceedings of the IEEE International Conference Robotics and Automation,* 4080–4085.

[19] Ataka, A., Qi, P., Shiva, A., Shafti, A., Wurdemann, H., Dasgupta, P., et al. (2016). "Towards safer obstacle avoidance for continuum-style manipulator in dynamic environments," in *Proceedings of the 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 600–605.

[20] Singh, L., Stephanou, H., and Wen, J. (1996). "Real-time robot motion control with circulatory fields," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 3, 2737–2742.

[21] Singh, L., Wen, J., and Stephanou, H. (1997). "Motion planning and dynamic control of a linked manipulator using modified magnetic fields," in *Proceedings of the 2004 IEEE International Conference on Control Applications*, 9–15.

[22] Haddadin, S., Belder, R., and Albu-Schäffer, A. (2011). "Dynamic motion planning for robots in partially unknown environments," in *Proceedings of the IFAC World Congress*, 18.

[23] Ataka, A., Qi, P., Shiva, A., Shafti, A., Wurdemann, H., Dasgupta, P., et al. (2016). "Real-time Pose estimation and obstacle avoidance for multi-segment continuum manipulator in dynamic environments," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2827–2832.

[24] Siciliano, B. (1990). Kinematic control of redundant robot manipulators: a tutorial. *J. Intell. Robot. Syst.* 3, 201–212.

[25] Jackson, J. D. (1999). *Classical electrodynamics*, 3rd edition. (New York, NY: Wiley).

[26] Bergeles, C., and Dupont, P. E. (2013). "Planning stable paths for concentric tube robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3077–3082.

[27] Palmer D., Cobos-Guzman, S., and Axinte, D. (2014). Real-time method for tip following navigation of continuum snake arm robots. *Robot. Auton. Syst.* 62, 1478–1485.