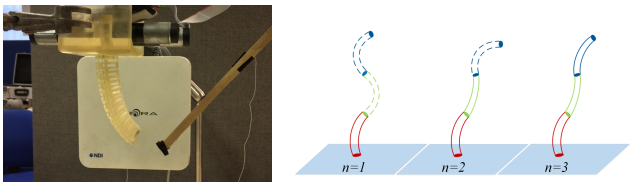
Real-Time Pose Estimation and Obstacle Avoidance for Multi-segment Continuum Manipulator in Dynamic Environments*

Ahmad Ataka[1], Peng Qi[2], Ali Shiva[1], Ali Shafti[1], Helge Wurdemann[3], Hongbin Liu[1], and Kaspar Althoefer[4]

***Abstract*— In this paper, we present a novel pose estimation and obstacle avoidance approach for tendon-driven multi-segment continuum manipulators moving in dynamic environments. A novel multi-stage implementation of an Extended Kalman Filter is used to estimate the pose of every point along the manipulator's body using only the position information of each segment tip. Combined with a potential field, the overall algorithm will guide the manipulator tip to a desired target location and, at the same time, keep the manipulator body safe from collisions with obstacles. The results show that the approach works well in a real-time simulation environment that contains moving obstacles in the vicinity of the manipulator.**



(a) (b)

Fig. 1. (a) A tendon-driven single-segment arms with NDI Aurora electro-magnetic based tracker for tip pose measurement. (b) Illustration of the proposed multi-stages pose estimation for three-segments manipulator. The solid line specifies active segments.

I. INTRODUCTION

The field of continuum robots has become very popular in recent years. Various continuum robot structures have been reported on in the last decade, including tendon-driven actuation [1], pneumatically actuated designs [2] and stiffness-controllable manipulators [3]. Kinematic and dynamic modelling, as well as the control of continuum manipulator have also been explored as reported in [4].

Very much inspired by nature, for example using the octopus arm [5] as role models, continuum manipulators possess higher flexibility and maneuverability in comparison with standard rigid-link manipulators. Exploiting those features, continuum arms can now be applied in areas in which they were previously prohibited from, such as surgery. However, this beneficial characteristic leads to a complexity in estimating their pose and modelling their highly flexible structure accurately. This in turn leads to new challenges when designing navigation strategies. Combining a model-based pose estimator and an obstacle avoidance algorithm for continuum manipulators is a suitable approach to achieve collision-free motion, especially in dynamic environments.

Pose estimation strategies have been implemented for various types of continuum manipulators, such as snake-like robots [6], [7], concentric tube [8], steerable catheter [9], as well as multi-section tendon-driven manipulators [10]. Most of the previous works make use of the Kalman-Filter [6], [8]. Several researchers rely on visual sensing [8], [9] which can be unreliable for dynamic environments. On the other hand, inertial sensors used, for example, by [7] suffer from an accumulated integration error. An interesting method as presented in [6] uses electro-magnetic trackers providing position and orientation information of a point along the body of manipulators. However, none of the works tries to use estimation results to improve obstacle avoidance.

In the obstacle avoidance side, real-time adaptive motion planning was implemented for multi-segment continuum manipulator in dynamic environments [11]. However, it assumes a planar workspace, limiting its applicability. Another study suggested an inverse-Jacobian-based planning which provides solutions in actuator space [12]. However, it is specifically designed for a manipulator moving inside a tubular environment. A classical potential field, like the one employed in [13], albeit suffers from local minima in complex environments, is reactive and thus suitable to be used for real-time applications in dynamic environments.

In this paper, we propose an approach based on real-time pose estimation and obstacle avoidance for tendon-driven multi-segment continuum manipulators, as an extension to the one shown in Figure 1a [1], moving in dynamic environments. A novel multi-stage implementation of a non-linear observer based on an Extended Kalman Filter is proposed to estimate the pose of every point along the manipulator's body with the tip's position serves as the only information

*Research is partially supported by 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.

[1]A. Ataka, A. Shiva, A. Shafti, and H. Liu are with The Centre for Robotics Research (CoRe), Department of Informatics, Kings College London, London WC2R 2LS, United Kingdom. Corresponding author e-mail: ahmad.ataka@kcl.ac.uk

[2]P. Qi is with the Department of Biomedical Engineering, Faculty of Engineering, National University of Singapore, Singapore 117583. (e-mail: peng.qi@nus.edu.sg)

[3]H. Wurdemann is with the Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom. (e-mail: h.wurdemann@ucl.ac.uk)

[4]K. Althoefer is with the School of Engineering and Material Science, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom. (e-mail: k.althoefer@qmul.ac.uk)

Corresponding author e-mail: ahmad.ataka@kcl.ac.uk

provided besides the kinematic model. To the best of our knowledge, this is the first work combining pose estimation and obstacle avoidance for collision-free navigation of tendon-driven multi-segment continuum manipulators, as an extension to our previous work [14]. The overall strategy will guide the manipulator's tip to a desired target location and keep the entire manipulator body avoid obstacles.

II. CONTINUUM MANIPULATOR MODELLING

A. Forward Kinematics and Jacobian Derivation

In this paper, each segment of the manipulator is assumed to be a circle with a constant radius of curvature. Therefore, each segment can be parameterized by configuration space variables $\mathbf{k}_i = \begin{bmatrix}\kappa_i & \phi_i & s_i\end{bmatrix}^T$, which stand for the curvature (κ_i), rotational deflection angle (ϕ_i), and arc length (s_i) of segment-i. The homogeneous transformation matrix describing the tip pose of segment-i with respect to its base, ${}^{i}_{i-1}\mathbf{T}(\mathbf{k}_i) \in SE(3)$, depends on $\mathbf{k}_i$ as presented in [4]. The pose of the end effector with respect to the world frame for N segments manipulator is described as follows

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

where $\mathbf{k} = \begin{bmatrix}\mathbf{k}_1 & \mathbf{k}_2 & \dots & \mathbf{k}_N\end{bmatrix}^T$. The base of the bottom segment is fixed.

For tendon-driven actuation, the actuator space variables of segment-i are written as $\mathbf{q}_i = \begin{bmatrix}l_{i1} & l_{i2} & l_{i3}\end{bmatrix}^T$ where the length of tendon-j in segment-i is given by $L + l_{ij}$ for a segment with normal length L. The complete actuator space variables for N segments manipulator can then be expressed as $\mathbf{q} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_N\end{bmatrix}^T$. The configuration space variables $\mathbf{k}$ can be expressed as function of the tendon length $L + l_{ij}$ and cross-section radius d as presented in [4].

A point along the segment of manipulator is specified by a scalar $\xi_i \in [0,1]$ from the base ($\xi_i = 0$) to the tip ($\xi_i = 1$) of segment-i. The set of scalars of all segments are then constructed into a scalar coefficient vector [15], $\xi = \begin{bmatrix}\xi_1 & \xi_2 & \dots & \xi_N\end{bmatrix}^T$. The value of the vector is defined as $\xi = \{\xi_r = 1 : \forall r < i, \quad \xi_i, \quad \xi_v = 0 : \forall v > i\}$.

So, the complete forward kinematic is given by

$$ {}^{N}_{0}\mathbf{T}(\mathbf{q},\xi) = \begin{bmatrix}\mathbf{R}(\mathbf{q},\xi) & \mathbf{p}(\mathbf{q},\xi) \\ \mathbf{0}_{1\times 3} & 1\end{bmatrix} \tag{2} $$

where $\mathbf{R}(\mathbf{q},\xi) \in SO(3)$ stands for the rotation matrix and $\mathbf{p}(\mathbf{q},\xi) \in \mathbb{R}^3$ stands for the position vector of the point along the body of manipulator. A Jacobian, defined as $\mathbf{J}(\mathbf{q},\xi) = \frac{\partial \mathbf{p}(\mathbf{q},\xi)}{\partial \mathbf{q}} \in \mathbb{R}^{3\times 3N}$, is used to map velocity in task space to configuration space as follows

$$ \dot{\mathbf{p}}(\mathbf{q},\xi) = \mathbf{J}(\mathbf{q},\xi)\dot{\mathbf{q}} \Leftrightarrow \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q},\xi)^{+}\dot{\mathbf{p}}(\mathbf{q},\xi), \tag{3} $$

where the $(^+)$ operation stands for the pseudo-inverse matrix operation defined as $\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$.

B. State-Space Representation

The kinematic model of the continuum manipulator can be rewritten in the form of the state space representation. Assume that we will only analyze $n \leq N$ segments. For the sake of simplicity, we call these n segments *active segments*. As we will show in the next section, the number of active segments can be adjusted over time. Using our model, the tendon's actuator space $\mathbf{q} \in \mathbb{R}^{3n}$ corresponds to the state variable $\mathbf{x}$ while the input $\mathbf{u}$ is represented by the DC motor's rotational speed, in which the tendon is connected, as an actuator. Hence, the input value is equivalent to the tendon length's rate of change $\dot{\mathbf{q}} \in \mathbb{R}^{3n}$. The state variables and input of the system can then be written as

$$ \mathbf{x} = \mathbf{q} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_n\end{bmatrix}^T, \tag{4} $$

$$ \mathbf{u} = \dot{\mathbf{q}} = \begin{bmatrix}\dot{\mathbf{q}}_1 & \dot{\mathbf{q}}_2 & \dots & \dot{\mathbf{q}}_n\end{bmatrix}^T, \tag{5} $$

where $\mathbf{q}_i = \begin{bmatrix}l_{i1} & l_{i2} & l_{i3}\end{bmatrix}^T$. The state equation is defined as

$$ \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k + \Delta t \mathbf{u}_k, \tag{6} $$

where Δt stands for the time sampling.

A 3-DOF electro-magnetic-based tracker is assumed to be embedded in the tip of each segment. This sensor will give the tip's position of every segment with respect to the manipulator base at a rate whose value is assumed to be equal to the sampling frequency $\frac{1}{\Delta t}$ so that there is always updated data available at every sampling step. Hence, the output value $\mathbf{y} \in \mathbb{R}^{3n}$ is represented by the tip position of each active segment,

$$ \mathbf{y}_k = g(\mathbf{x}_k) = \begin{bmatrix}\mathbf{p}(\mathbf{x}_k, \xi = \chi_1) & \dots & \mathbf{p}(\mathbf{x}_k, \xi = \chi_n)\end{bmatrix}^T, \tag{7} $$

where $\chi_i = \begin{bmatrix}\xi_1 = 1 & \dots & \xi_i = 1 & \xi_{i+1} = 0 & \dots & \xi_n = 0\end{bmatrix}^T$ and $\mathbf{p}(\mathbf{x}_k, \xi)$ is derived using (2).

III. POSE ESTIMATION AND OBSTACLE AVOIDANCE

A. Multi-Stages Pose Estimation

A non-linear observer, which does not rely on the information of the state's initial value, is needed to estimate the pose of a point along the manipulator's body. The well-known Extended Kalman Filter (EKF) [16] is used to estimate the state value. Given the state equation and output equation as expressed in (6)-(7), we can estimate the next tendon length value $\hat{\mathbf{x}}_{k+1|k+1}$ based on the current value of estimate $\hat{\mathbf{x}}_{k|k}$, input $\mathbf{u}_k$, and the measurement data $\mathbf{y}_k$. For n number of active segments, the EKF can be expressed as follows

$$ \begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= f(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k), \\ \mathbf{P}_{k+1|k} &= \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^T + \mathbf{Q}_k, \\ \mathbf{K}_k &= \mathbf{P}_{k+1|k}\mathbf{C}_k^T(\mathbf{C}_k \mathbf{P}_{k+1|k}\mathbf{C}_k^T + \mathbf{R}_k)^{-1}, \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_k(\mathbf{y}_k - g(\hat{\mathbf{x}}_{k+1|k})), \\ \mathbf{P}_{k+1|k+1} &= (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_{k+1|k}, \end{aligned} \tag{8} $$

where $\mathbf{P}_{k|k}$, $\mathbf{Q}_k$, $\mathbf{R}_k \in \mathbb{R}^{3n\times 3n}$ represent the estimation covariance, process noise variance and measurement noise variance matrix respectively, $\mathbf{K}_k \in \mathbb{R}$ represents the Kalman

gain, while the output $\mathbf{y}_k$ is computed from position sensor measurement data, such as an electro-magnetic tracker.

The matrix $\mathbf{A}_k \in \mathbb{R}^{3n\times 3n}$ and $\mathbf{C}_k \in \mathbb{R}^{3n\times 3n}$ are a local linearization of (6) and (7), defined as $\mathbf{A}_k = \frac{\partial f(\mathbf{x}_k,\mathbf{u}_k)}{\partial \mathbf{x}_k}$ and $\mathbf{C}_k = \frac{\partial g(\mathbf{x}_k)}{\partial \mathbf{x}_k}$. Using our state space equation model, $\mathbf{A}$ turns out to be an identity matrix while $\mathbf{C}$ turns out to be a matrix consisting of the Jacobian of each segment tip, as follows

$$\mathbf{A}_k = \mathbf{I} \in \mathbb{R}^{3n\times 3n}, \tag{9}$$

$$\mathbf{C}_k = \begin{bmatrix}\mathbf{J}(\mathbf{x}_k, \xi = \chi_1) & \dots & \mathbf{J}(\mathbf{x}_k, \xi = \chi_n)\end{bmatrix}^T. \tag{10}$$

The tip position estimation $\hat{\mathbf{y}}_{k|k}$ is derived by applying Eq. (7) to the state estimation $\hat{\mathbf{x}}_{k|k}$ as follows,

$$\hat{\mathbf{y}}_{k|k} = g(\hat{\mathbf{x}}_{k|k}). \tag{11}$$

However, as will be shown in Section IV, implementing a standard EKF for all segments simultaneously from the beginning (i.e., that the number of active segments chosen to be equal to the total number of segments ($n = N$) at all times), it is possible for the state estimate to arrive at a physically impossible state value, such as a negative tendon length. This is due to the fact that there can be more than one mathematically possible state value for a given segment tip position. To overcome this problem, a novel multi-stage implementation of a non-linear observer is proposed.

For a three segment manipulator, the estimation starts from the bottom segment only, followed by the bottom and middle segments jointly, and finally all three segments as one are considered, as illustrated in Figure 1b. This approach is motivated by the fact that the large estimation error of the base segment will disturb the performance of estimation in the more distal segments. By doing such a stage-by-stage estimation, a segment will be taken into consideration for the estimation process (i.e. by incrementing the number of active segments n) only after the tip position estimation error of the previous segment(s) is lower than a predefined threshold value δ. The proposed algorithm is formally defined in Algorithm 1.

It is important to be noted that the size of the following variables $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\mathbf{Q}_k$, $\mathbf{R}_k$, $\mathbf{u}_k$, $\mathbf{y}_k$, $\hat{\mathbf{y}}_{k|k}$, $\mathbf{A}_k$, $\mathbf{C}_k$ depend on the number of active segments n. In line 3 of Algorithm 1, we initialize a component of state vector $\mathbf{q}_n \in \mathbb{R}^3$. This vector is then added as a new component of $\hat{\mathbf{x}}_{k|k}$ in line 4. The function Initialize() in line 5 will initialize matrix $\mathbf{P}_{k|k}$, $\mathbf{Q}_k$, $\mathbf{R}_k \in \mathbb{R}^{3n\times 3n}$ every time the active segments is incremented. The routine GetInputSignal() and GetSensorData() in line 7 and 8 will return vector $\mathbf{u}_k \in \mathbb{R}^{3n}$ and $\mathbf{y}_k \in \mathbb{R}^{3n}$ retrieved from the input signal and sensor data respectively. In line 13, the error vector $\mathbf{e}$ is derived by subtracting the latest component of $\hat{\mathbf{y}}_{k|k}$ and $\mathbf{y}_k$, i.e. $\mathbf{e}$ stands for the tip's position estimation error of the last active segment.

B. The Complete Obstacle Avoidance Algotihm

A modified version of a reactive potential field method based on [13] is used. The field is considered as the velocity of the manipulator $\dot{\mathbf{p}}$ defined as the negative gradient of a potential function U, i.e. $\dot{\mathbf{p}} = -\nabla_{\mathbf{p}} U(\mathbf{p})$.

Algorithm 1 Multi-Stages Pose Estimation

1: $n \leftarrow 1$, $k \leftarrow 0$
2: **loop**
3: $\quad \mathbf{q}_n \leftarrow$ InitState()
4: $\quad \hat{\mathbf{x}}_{k|k} \leftarrow$ Add($\mathbf{q}_n$)
5: $\quad (\mathbf{P}_{k|k}, \mathbf{Q}_k, \mathbf{R}_k) \leftarrow$ Initialize(n)
6: $\quad$ **loop**
7: $\quad\quad \mathbf{u}_k \leftarrow$ GetInputSignal(n)
8: $\quad\quad \mathbf{y}_k \leftarrow$ GetSensorData(n)
9: $\quad\quad \mathbf{A}_k \leftarrow$ Equation (9)
10: $\quad\quad \mathbf{C}_k \leftarrow$ Equation (10)
11: $\quad\quad (\hat{\mathbf{x}}_{k+1|k+1}, \mathbf{P}_{k+1|k+1}) \leftarrow$ EKF($\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\mathbf{u}_k$, $\mathbf{y}_k$, $\mathbf{A}_k$, $\mathbf{C}_k$)
12: $\quad\quad \hat{\mathbf{y}}_{k|k} \leftarrow$ Equation (11)
13: $\quad\quad \mathbf{e} \leftarrow$ DetermineError($\hat{\mathbf{y}}_{k|k}$, $\mathbf{y}_k$)
14: $\quad\quad k \leftarrow k+1$
15: $\quad\quad$ **if** norm($\mathbf{e}$) $< \delta$ **and** $n < N$ **then**
16: $\quad\quad\quad$ **break**
17: $\quad\quad$ **end if**
18: $\quad$ **end loop**
19: $\quad n \leftarrow n+1$
20: **end loop**

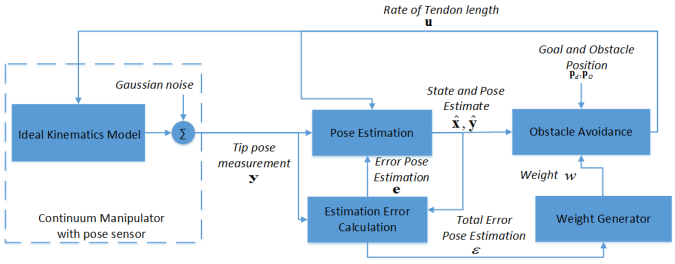


Fig. 2. The block diagram of the proposed algorithm.

The attractive potential field to attract the manipulator's tip to a desired position $\mathbf{p}_d$ can be expressed as

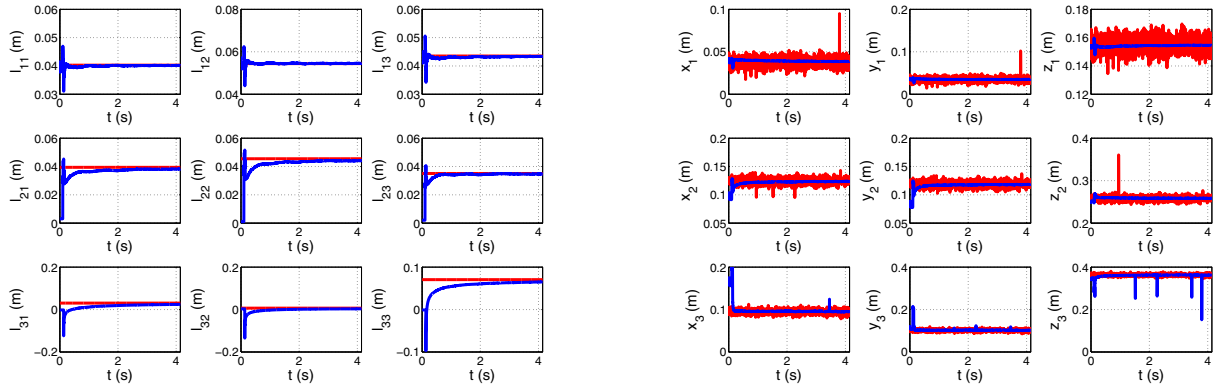
$$\dot{\mathbf{p}}_{\mathbf{p}_d} = -c(\mathbf{p} - \mathbf{p}_d). \tag{12}$$

where c is an attractor stiffness. The repulsive potential produced by an obstacle $\mathscr{O}$ can be expressed as follows

$$\dot{\mathbf{p}}_{\mathscr{O}} = \begin{cases} \eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2}\frac{\partial \rho}{\partial \mathbf{p}} & \text{if } \rho < \rho_0 \\ 0 & \text{if } \rho \geq \rho_0 \end{cases}. \tag{13}$$

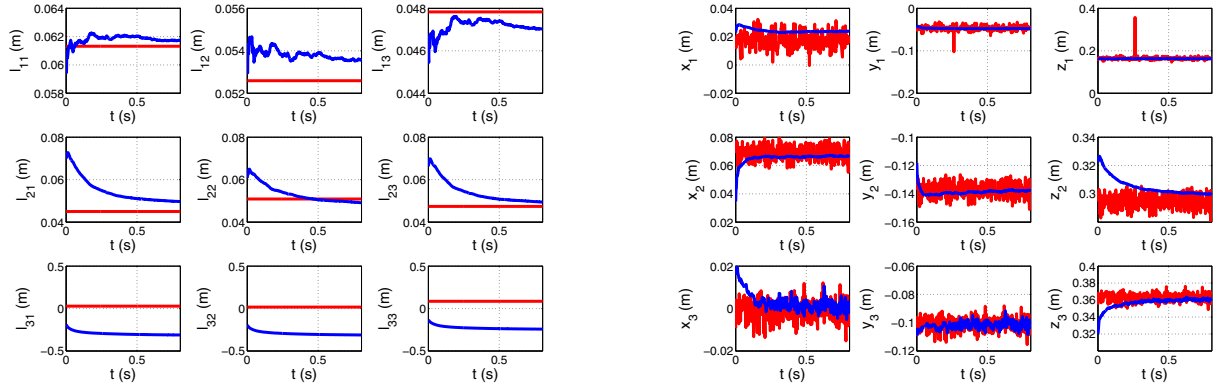
where $\rho = \sqrt{(\mathbf{p} - \mathbf{p}_{\mathscr{O}})^T(\mathbf{p} - \mathbf{p}_{\mathscr{O}})}$ is the closest distance between a point in the manipulator's body to an obstacle $\mathscr{O}$, η is positive repulsion constant, and ρ_0 is the range of the potential influence.

To make the whole body of the manipulator safe from collision, we choose a number of *point subjected to potentials* (PSP) along the body of the manipulator. The pose of a PSP in segment-i is expressed as $\mathbf{p}(\mathbf{q}, \xi = \lambda_i)$ where $\lambda_i = \begin{bmatrix}\xi_1 = 1 & \dots & \xi_i \in [0,1] & \xi_{i+1} = 0 & \dots & \xi_N = 0\end{bmatrix}^T$. The repulsive potential will be applied only to the PSP which is closest to the obstacles.



(a) (b)

Fig. 3. The pose estimation using an multi-stages EKF for the zero-input scenario shows (a) comparison between the true state (red line) and the estimated state (blue line) and (b) comparison between the true pose with added Gaussian noise (red line) and the estimated pose (blue line).



(a) (b)

Fig. 4. The pose estimation using a standard EKF for the zero-input scenario shows (a) comparison between the true state (red line) and the estimated state (blue line) and (b) comparison between the true pose with added Gaussian noise (red line) and the estimated pose (blue line).

Finally, we transform each spatial velocity, applied at the end-effector and the PSP, to the corresponding actuator space velocity as an input for the kinematic model and the EKF, $\mathbf{u}_k$. The total velocity in the actuator space is given by

$$\mathbf{u}_k = \dot{\mathbf{q}} = \mathbf{J}_e^{+}\dot{\mathbf{p}}_{\mathbf{p}_d} + \mathbf{J}_a^{+}\dot{\mathbf{p}}_{\mathscr{O}} \tag{14}$$

where $\mathbf{J}_e$ and $\mathbf{J}_a$ denote the Jacobian of the tip and the Jacobian of the PSP-a, i.e. the closest PSP to the obstacles.

The estimation and obstacle avoidance are combined together as shown in Figure 2. The pose of the end-effector $\mathbf{p}(\mathbf{x}_k, \xi = \chi_N)$ and the PSPs $\mathbf{p}(\mathbf{x}_k, \xi = \lambda_i)$ needed by the obstacle avoidance stage are all retrieved from the estimation stage. The estimated poses are determined by doing forward kinematics for the estimated state produced by EKF, i.e. $\hat{\mathbf{p}}_k(\xi) = \mathbf{p}(\hat{\mathbf{x}}_{k|k}, \xi)$. The pose information is used by the obstacle avoidance stage to produce input signal $\mathbf{u}_k$ as given in (14) and then feed this to the estimation stages as written in line 7 of Algorithm 1.

It is important to be noted that in the early phase, the estimated state and pose are not yet to be close to the real value. Thus, it becomes necessary to deactivate the obstacle avoidance stage at the beginning of the estimation process and activate it only when the sum of the tip position estimation errors for all segments, ε, is less than a predefined threshold value Ω. This is done by multiplying the value of the attractor stiffness c and repulsion constant η with a weight scalar w whose value is given as follows

$$w = \begin{cases} 0 & \text{if } \varepsilon \geq \Omega \\ 1 & \text{if } \varepsilon < \Omega \end{cases}, \tag{15}$$

where $\varepsilon = \sum_{i=1}^{N} \text{norm}(\hat{\mathbf{p}}_k(\xi = \chi_i) - \mathbf{p}(\mathbf{x}_k, \xi = \chi_i))$.

IV. RESULTS AND DISCUSSION

We implemented our proposed algorithm in ROS (Robot Operating System), a real-time simulation environment, running at a rate of $\frac{1}{\Delta t}$ =40 Hz for a three-segment manipulator model. Table I shows the properties of the manipulator and the algorithm. We have 5 number of PSPs distributed

TABLE I

PROPERTIES OF MANIPULATOR AND THE PROPOSED ALGORITHM

Param	Value	Param	Value
d	0.0134 m	δ	5×10^{-3} m
L	0.12 m	c	2
N	3	η	2×10^{-7}
$\mathbf{Q}_k$	$10^{-10}\mathbf{I}\in\mathbb{R}^{3n\times3n}$	ρ_0	0.07 m
$\mathbf{R}_k$	$2.5\times10^{-5}\mathbf{I}\in\mathbb{R}^{3n\times3n}$	Ω	0.06 m
Δt	2.5×10^{-2} s		

uniformly along the backbone of each segment from the tip to the point close to the base. A spherical obstacle with a radius of 1 cm is assumed to move at a constant speed.

The measurement data, i.e. the tip pose of each segment $\mathbf{y}_k$, is retrieved from an ideal kinematic model with added Gaussian noise to simulate the electromagnetic position sensor. The noise has a zero mean and a standard deviation of $\sigma = 5\times10^{-3}$, i.e. the variance will be $\mathbf{R}_k = \sigma^2\mathbf{I}\in\mathbb{R}^{3n\times3n}$. The kinematic model has the true state $\mathbf{x}_k$ updated at every iteration, which is unknown, so that the obstacle avoidance stage which relies on the estimated states $\hat{\mathbf{x}}_{k|k}$ from the EKF. It is important to be noted that the state value in this paper, l_{ij}, refers to the deviation from a normal length L, so that the tendon length is given by $L+l_{ij}$.

A. State and Pose Estimation

In the first pose estimation simulation, the input to the manipulator is assumed to be zero ($\mathbf{u}=\mathbf{0}\in\mathbb{R}^{3N}$) at all times. Starting from a random state value, we compare the performance of the proposed multi-stage EKF and standard EKF as shown in Figure 3 and 4 respectively. The resulting segment length estimate is then used to produce tip pose estimate via forward kinematics using (11).

We can see that the proposed multi-stage estimation is able to estimate the states value (Figure 3a) as well as the tip positions of all segments (Figure 3b) and, at the same time, suppress the noise in the tip's position measurement. Although capable of estimating the tip position (Figure 4b), the standard EKF fails to estimate the states value correctly (Figure 4a). As we can observe, some of the state values, for the third segment in this case, will produce a negative tendon length, which is physically not possible. Hence, even though the tip position estimation error converges to zero, we are not able to correctly estimate the position of every point along the segment of the manipulator except the tip using this approach. Thus, the proposed multi-stage pose estimation has better performance.

In the second simulation, we force the manipulator end effector to follow a circular path. The input signal $\mathbf{u}_k$ is derived from the inverse Jacobian equation in (3). The results (see Figures 5a and 5b) show that the proposed multi-stage EKF can cope with the variation in the input signal and tip pose measurement data.

B. Obstacle Avoidance

In this part, the pose estimator is combined with the obstacle avoidance approach. A dynamically moving spherical obstacle, drawn as a black sphere, moves close to the body of manipulator while the target position, a small red dot, is assumed to be constant.

In the first simulation, the obstacle moves at a height such that it is close to the first segment, as shown in Figure 6a. In the second simulation, the obstacle moves close to the middle segment, as shown in Figure 6b. Both figures show how the obstacle avoidance approach makes use of the pose estimation result to avoid collisions between the body of the manipulator and the moving obstacle while also continuously attracting the end effector to the desired target position. It is noted that the algorithm will avoid not only the tip of each segment, where the position sensors are attached, from collision, but also the whole body of the manipulator. This feature can not be achieved without combining the pose estimation and obstacle avoidance and thus, it becomes the main contribution of the proposed algorithm.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a real-time pose estimation and obstacle avoidance approach for a tendon-driven multi-segment continuum manipulator operating in dynamic environments. To the best of our knowledge, this is the first time pose estimation and obstacle avoidance have been combined to achieve collision-free motion for a multi-segment continuum manipulator. The proposed multi-stage implementation of the EKF outperforms the standard EKF and correctly estimates the state values. The overall algorithm works well in a real-time simulation environment.

In the future, this algorithm can be applied to real manipulators. Future work will also explore the application of our approach to a more advanced dynamic model of the continuum manipulator and a possible improvement on the motion planning side, by combining the local reactive obstacle avoidance with a global motion planning strategy when operating in more complex environments.

REFERENCES

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

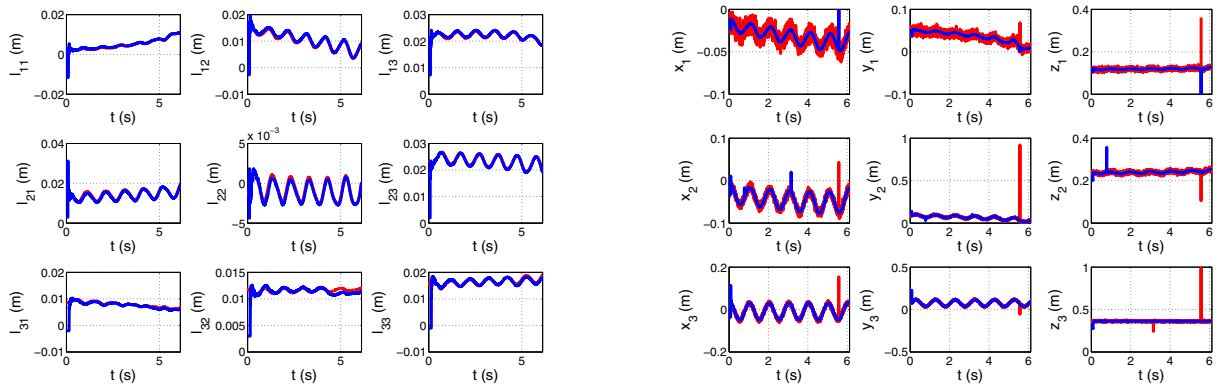
[2] T. Mahl, A. Hildebrandt, and O. Sawodny, "A Variable Curvature Continuum Kinematics for Kinematic Control of the Bionic Handling Assistant," *Robotics, IEEE Transactions on*, vol. 30, no. 4, pp. 935–949, Aug. 2014.

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

[4] R. J. Webster, III and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *Int. J. Rob. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010. [Online]. Available: http://dx.doi.org/10.1177/0278364910368147

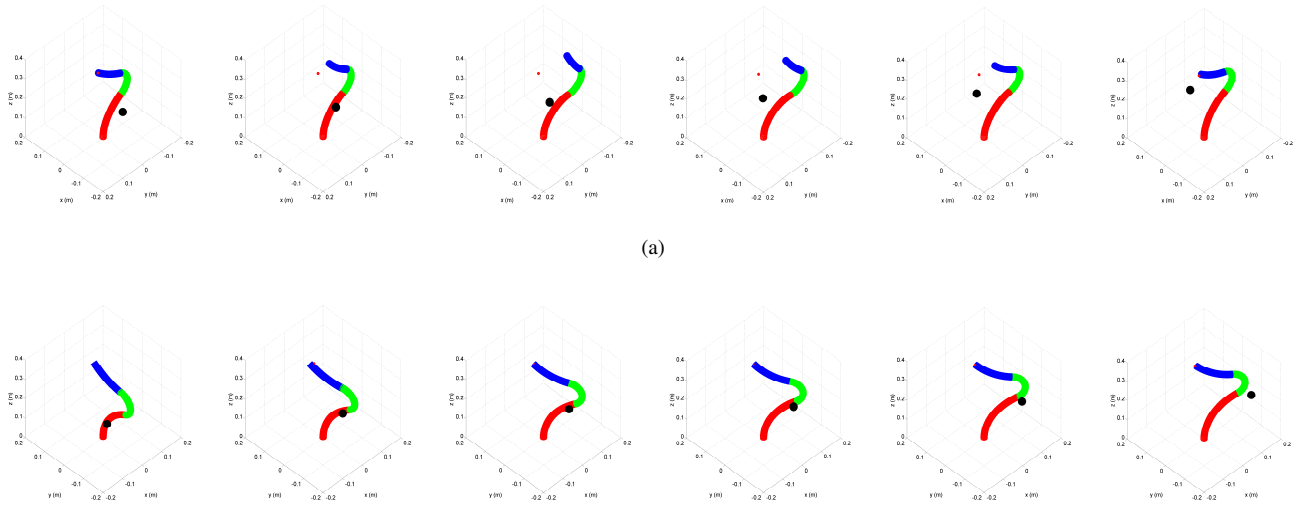
[5] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. Walker, B. Jones, M. Pritts, D. Dienno, M. Grissom, and C. Rahn, "Field trials and testing of the OctArm continuum manipulator," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2336–2341.

[6] R. Srivatsan, M. Travers, and H. Choset, "Using Lie algebra for shape estimation of medical snake robots," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept. 2014, pp. 3483–3488.



(a) (b)

Fig. 5. The pose estimation using an multi-stage EKF for the circular-end-effector-path scenario shows (a) comparison between the true state (red line) and the estimated state (blue line), (b) comparison between the true pose with added Gaussian noise (red line) and the estimated pose (blue line)



(a)

(b)

Fig. 6. A three-segments continuum manipulator with a static target (small red dot) when obstacle (black sphere) moves close to (a) the bottom segment and (b) middle segment. The order of movement is from left to right picture.

[7] Z. Zhang, J. Shang, C. Seneci, and G.-Z. Yang, "Snake robot shape sensing using micro-inertial sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov. 2013, pp. 831–836.

[8] E. Lobaton, J. Fu, L. Torres, and R. Alterovitz, "Continuous shape estimation of continuum robots using X-ray images," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 725–732.

[9] J. Borgstadt, M. Zinn, and N. Ferrier, "Multi-modal localization algorithm for catheter interventions," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 5350–5357.

[10] W. Rone and P. Ben-Tzvi, "Multi-segment continuum robot shape estimation using passive cable displacement," in *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*, Oct. 2013, pp. 37–42.

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

[12] G. Chen, M. T. Pham, and T. Redarce, "Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy," *Robotics and Autonomous Systems*, vol. 57, no. 67, pp. 712 – 722, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889008002042

[13] A. Ataka, P. Qi, H. Liu, and K. Althoefer, "Real-Time Planner for Multi-Segment Continuum Manipulator in Dynamic Environments," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, May 2016, pp. 4080–4085.

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

[15] I. Godage, D. Branson, E. Guglielmino, and D. Caldwell, "Path planning for multisection continuum arms," in *Mechatronics and Automation (ICMA), 2012 International Conference on*, Aug. 2012, pp. 1208–1213.

[16] A. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in Science and Engineering. Elsevier Science, 1970. [Online]. Available: https://books.google.co.id/books?id=nGlSNvKyY2MC