Adaptive Neural Network Control of a Compact Bionic Handling Arm

Achille Melingui, Othman Lakhal, Boubaker Daachi, Jean Bosco Mbede, and Rochdi Merzouki

***Abstract*—In this paper, autonomous control problem of a class of bionic continuum robots named "Compact Bionic Handling Arm" (CBHA) is addressed. These robots can reproduce biological behaviors of trunks, tentacles, or snakes. The modeling problem associated with continuum robots includes nonlinearities, structured and unstructured uncertainties, and the hyperredundancy. In addition to these problems, the CBHA comprises the hysteresis behavior of its actuators and a memory phenomenon related to its structure made of polyamide materials. These undesirable effects make it difficult to design a control system based on quantitative models of the CBHA. Thus, two subcontrollers are proposed in this paper. One, encapsulated in the other, and both implemented in real time allow controlling of the CBHA's end-effector position. The first subcontroller controls the CBHA's kinematics based on a distal supervised learning scheme. The second subcontroller controls the CBHA's kinetics based on an adaptive neural control. These subcontrollers allow a better assessment of the stability of the control architecture while ensuring the convergence of Cartesian errors. The obtained experimental results using a CBHA robot show an accurate tracking of the CBHA's end-effector position.**

***Index Terms*—Adaptive neural networks control, bionic manipulators, continuum robots.**

I. Introduction

NOWADAYS, different types of robotic systems are in use or under development. One of the main goals is to use these robots in the home or industry. For this reason, it is important to develop arms capable of helping the elderly or persons with reduced mobility in household tasks. For such applications, it is necessary to develop a class of robotic arms with average performance with respect to industrial rigid arms, but it should be cost effective.

Development of lightweight robot arms should incorporate in their design: Mechanical structure made by rapid manufacturing, electric and pneumatic actuators, compliant gripper, intelligent control, etc. Continuum manipulators are increasingly used in scientific research [1], and in a wide range of applications including medical [2], military [3], manipulation [4], etc.

Because of the flexibility of their structure, it is necessary to find a compromise in performance obtained in terms of stability, precision, and rapidity. It is necessary to develop accurate models (often kinematic) in order to improve the closed-loop control of these flexible arms. Moreover, it is important to consider dynamic models of these flexible arms in case of dynamic interactions with the external environment. Generally, modeling of such systems focuses on the kinematic behaviors than the dynamic because of their soft behaviors.

In the literature, two types of modeling approaches exist. Both are based on information available to describe the behavior of the system (kinematic and dynamic). Qualitative approach is to divide the parameter space into several classes according to the operation modes and to determine mathematical relationships between the effects (observation experts, sensor measurements, and statistics) and causes (actuators inputs) using learning techniques. Qualitative approaches can give accurate models when the learning data cover all the regions of the robot's workspace. Regarding quantitative approach, the existing methods [1]–[6] are also known as models-based methods. The first step is to formulate mathematically the relationship between causes and effects using the kinematic and dynamic fundamental equations, which are often differential or complex equations depending on the type of system. Quantitative models are usually less accurate, due to the unmodeled uncertainties and nonlinear behaviors of the continuum structures. To validate the developed models, we need to express the difference between the information provided by the system and those obtained from the model in normal operating conditions (no disturbance fault).

Regarding qualitative approaches, neural networks (NN) were used for the modeling and control of continuum robots [7]–[9]. Thanks to their universal approximation property, they have been widely used for the identification and control of nonlinear systems [10], [11]. Most contributions focused on the control of rigid robots [12]–[15]. However, NN were also used for the modeling and control of continuum robots. Giorelli *et al.* [8] used feedforward NN to approximate the inverse kinematic model (IKM) of a soft manipulator, which was controlled using three cables. Braganza *et al.* [7] implemented a low-level joint controller on a soft extensible manipulator by using NN to compensate for the dynamic uncertainties. However, one limitation of continuum manipulators related to its morphology is that even small changes in the actuated lengths can cause significant changes in the end-effector position. An interesting approach was proposed by Rolf and Steil [16], where a goal babbling

Manuscript received July 13, 2014; revised September 25, 2014 and November 30, 2014; accepted January 14, 2015. Date of publication March 10, 2015; date of current version October 21, 2015. Recommended by Technical Editor M. Basin. This work was supported by the European Project InTraDE 019C, from Interreg IVB NWE program.

A. Melingui and J. B. Mbede are with the Electrical and Telecommunications Engineering Department, Ecole Nationale Superieure Polytechnique, University of Yaounde I, 8390 Yaounde, Cameroon (e-mail: achillemelingui@gmail.com; mbede@neuf.fr).

O. Lakhal and R. Merzouki are with CRIStAL UMR-CNRS, Ecole Polytechnique de Lille, 59655 Villeneuve d'Ascq, France (e-mail: othman.lakhal@univ-lille1.fr; rochdi.merzouki@polytech-lille.fr).

B. Daachi is with LISSI Laboratory, University Paris Est Créteil, 94400 Vitry/Seine, France and also with the CNRS-AIST Joint Robotics Laboratory, UMI3218/CRT, Tsukuba 305-8568, Japan (e-mail: daachi@u-pec.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2015.2396114

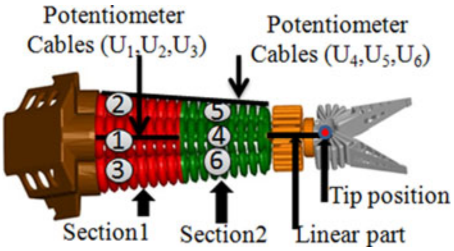
Potentiometer Cables (U$_1$,U$_2$,U$_3$)
Potentiometer Cables (U$_4$,U$_5$,U$_6$)
2
1
3
5
4
6
Tip position
Section1
Section2
Linear part

Fig. 1. CBHA continuum manipulator.

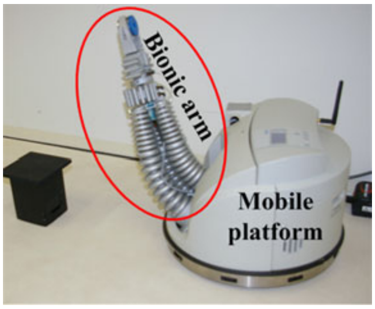

Fig. 2. Robotino XT bionic mobile manipulator.

approach is introduced to solve the IKM of the bionic handling assistant (BHA) robot. Starting from a particular goal, a new goal is randomly drawn from a set of target positions, and then, the endpoints are linearly interpolated.

Kinematically, the class of compact bionic handling arm (CBHA) robots has several nonlinear behaviors that make it difficult to obtain an accurate mathematical model. These undesirable effects include the memory effect of the polyamide material and the hyperredundancy of the robot (more than 40 DOF). Kinetically, these robots have nonstationary behaviors due to hysteresis effects of their actuation systems and the presence of nonneglected frictions between the tubes and the wire-potentiometers sensors.

An NN-based solution introduced in our previous work [17] is extended in this paper by considering fast CBHA's motions in order to overcome the aforementioned modeling and control issues. In contrast to [17], redundancies are taken into account by considering both CBHA's sections for the generation of desired trajectories. The design and stability of the adaptive controller are also addressed. Comparing existing control approaches [7],[9], several key features of this paper make it unique. The use of subcontrollers allows a better assessment of the stability of the control architecture while ensuring the convergence of Cartesian errors. Use of NN models reduces the execution time and the cost of implementation. An exhaustive exploration of the CBHA's workspace becomes unnecessary because the parameters of the adaptive controller are adjusted in real time. This ensures the tracking of any point of the robot's workspace.

Among scientific developments on this topic, Jordan and Rumelhart [18] proposed a distal supervised learning (DSL) scheme to avoid the direct inverse modeling entirely, which is not a goal-directed method, because the direct inverse algorithm samples in configuration space without regard to particular targets or errors in the task space. Stitt and Zheng [19] used the DSL approach to control biped robot movements, and Howard and Huckvale [20] used the DSL approach to build a system that can learn to mimic speech using its own vocal tract. In our previous work [21], [22], a squared penalty term was added to the objective function of the inverse NN (INN) into a DSL scheme to select one particular inverse model from the redundancy manifold. This DSL scheme is used in this paper to handle the kinematics of the CBHA.

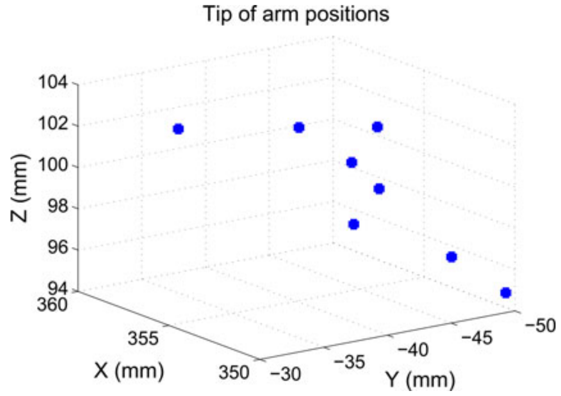

Fig. 3. Variation of the end-effector position of the CBHA for the same pressure input: Illustration of nonstationary behaviors.

Concerning adaptive NN control, some contributions have been made on rigid manipulators, such as in Daachi *et al.* [14], [15], where the authors used an NN adaptive controller allowing accurate tracking of trajectories inside the identified workspace. Le and Kang [23] proposed an adaptive NN control for parallel manipulators.

This paper is organized as follows: Section II presents the CBHA control problem statement. The kinematic modeling is developed in Section III, whereas the kinematic and kinetic controls are presented in Section IV. Section V provides experimental results and discussions, and Section VI gives the conclusions and future work.

II. Problem Statement

The CBHA depicted in Fig. 1 is attached to an omnidirectional mobile platform named Robotino to form the Robotino XT as shown in Fig. 2. It consists of two main segments, each equipped with three pneumatically actuated bellows. A rotational joint as the wrist is controlled by means of two actuators. Two compliant jaws constituting the gripper are controlled by one actuator. Each actuator can be controlled separately, the venting of the tubes allows resetting of its shape, and the compressed air supply leads to its expansion. CBHA comprises nine sensors, where six wire-potentiometers sensors installed on the surface of each flexible tube measure their elongations; two sensors are used for the rotating part; and the last one for detecting the status of the gripper.

Until now, the CBHA placed on the mobile platform is controlled in an open-loop system using a joystick interface. The objective is to design an autonomous controller. The main difficulty related to its structure is the development of accurate analytical models that can be exploited to develop control strategies. In this paper, the kinematic and kinetic models

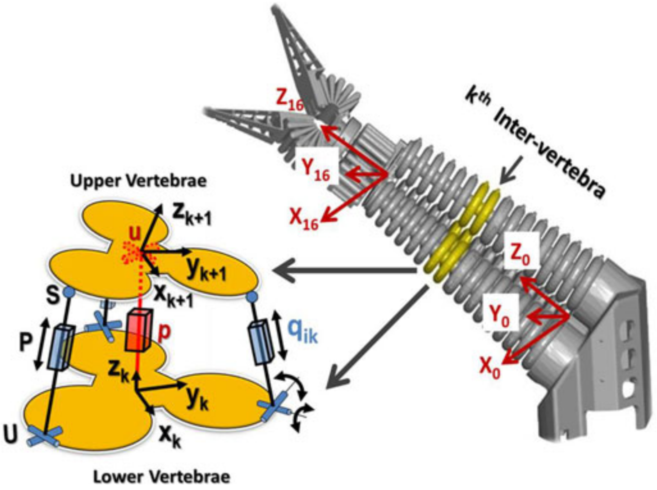

Fig. 4. Structure of 3UPS-1UP robot for one inter-vertebra.

based on artificial NN are used to control the CBHA. Note that the presence of pneumatic and mechanical uncertainties caused by friction, the memory of the structure, and hysteresis can strongly affect the positioning of the CBHA. Thus, for a same input pressure, it is possible to obtain different positions of the CBHA's end-effector. The example of Fig. 3 shows the position of the CBHA's end-effector when supplying the same pressure ($P = [1, 0, 1, 1, 1, 1]$) 25 times into CBHA's actuators with a time-out of 10 s sufficient to reach the mechanical equilibrium. The presence of hysteresis in the actuated pressure and the memory effect of the composite material do not allow an accurate position control of the CBHA's end-effector based only on pressure control. Thus, the length measurements of the tubes remain the sole reliable information for the CBHA positioning. These lengths can be controlled by dynamically adjusting the pressure in each actuator.

III. Kinematic Modeling of the CBHA

In this section, we first present the state equation of the CBHA, followed by its modeling based on a quantitative and qualitative approach. The learning base is presented thereafter, and the section ends with the initialization of the NN parameters.

A. State Equations

A CBHA can be modeled by an n-dimensional nonlinear discrete system with the following state equations:

$$x(k+1) = f[x(k), u(k)] + g(\zeta) \quad (1)$$

$$y(k) = h[x(k)] \quad (2)$$

$$\Delta t(k) = t(k+1) - t(k) \quad (3)$$

where $x(k) \in R^6$ is the state vector representing the wire-potentiometer voltages, $u(k) \in R^6$ is the control signal vector reflecting the pressure in each tube, and $y(k) \in R^3$ is the output of the system, namely the end-effector position of the CBHA. $g(\zeta)$ is a nonlinear function representing the nonstationary behaviors of the CBHA. f and h are unknown nonlinear functions.

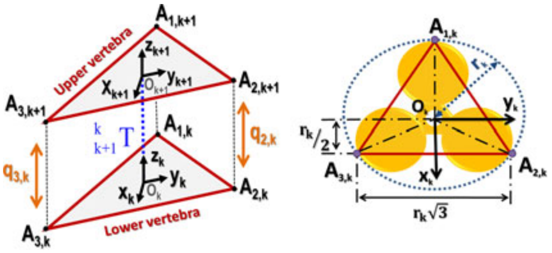

Fig. 5. Geometric of 3UPS-1UP robot for one inter-vertebra.

Equations (1) and (2) represent the kinetic and kinematic parts of the CBHA, respectively.

B. CBHA Modeling Based on Quantitative Approach

Equation (2) maps the relationship between the wire-potentiometer voltages and the position of the CBHA's end-effector. In our previous works [24], a forward kinematic model has been proposed based on the modeling principle of hybrid parallel rigid manipulators. The CBHA is considered as a concatenation of rigid vertebrae connected to each other by an inter-vertebra joint as shown in Fig. 4. Under these assumptions, the CBHA has been modeled as a serial-parallel manipulator. The motion of the upper vertebra linked in frame $\Re_{k+1}$ relative to the lower vertebra linked in frame $\Re_k$ is limited by the constraints imposed by the inter-vertebra joint. This behavior is reproduced by a set of 3UPS-1UP extensible joints as shown in Fig. 4.

The prismatic joint at the center of the 3UPS-1UP structure is passive and is used to constrain the linear extension and to neglect the torsion about its axis. There are three other prismatic joints located at the sides; these joints represent the actuated degrees of freedom of the 3UPS-1UP structure. Therefore, these joints are used to control the position and orientation of the upper vertebra relative to the lower vertebra. They are represented by the translation lengths $q_{m,k}$, where $m = 1, \ldots, 3$ is the number of active prismatic joints and $k = 1, \ldots, N-1$ is the inter-vertebra number. Fig. 5 describes the geometry of each inter-vertebra. The points $A_{1,k}, A_{2,k}$, and $A_{3,k}$ form an equilateral triangle. The upper vertebra points $A_{m,k+1} \equiv (A_{1,k+1}, A_{2,k+1}, A_{3,k+1})$ and the lower vertebra points $A_{m,k} \equiv (A_{1,k}, A_{2,k}, A_{3,k})$ are linked by the relationship

$$\begin{bmatrix} Q_{m,k} \\ 1 \end{bmatrix} = {}^{k+1}_{k}T \begin{bmatrix} A_{m,k+1} \\ 1 \end{bmatrix} - \begin{bmatrix} A_{m,k} \\ 1 \end{bmatrix} \quad (4)$$

where ${}^{k+1}_{k}T = \mathrm{Rot}(y, \theta_k) * \mathrm{Rot}(x, \psi_k) * \mathrm{Trans}(z, Z_k)$, where Rot and Trans are the rotation and translation matrices, respectively.

The prismatic variables $q_{m,k}$ are the distances between the connection points $A_{m,k+1}$ and $A_{m,k}$

$$q_{m,k} = {Q_{m,k}}^T Q_{m,k} \quad (5)$$

with $Q_{m,k} = [q_{1,k}, q_{2,k}, q_{3,k}]^T$, $A_{1,k} = [r_k, 0, 0]^T$, $A_{2,k} = [\frac{1}{2}r_k, \frac{\sqrt{3}}{2}r_k, 0]^T$, and $A_{3,k} = [\frac{1}{2}r_k, -\frac{\sqrt{3}}{2}r_k, 0]^T$.

The kinematic equations of an inter-vertebra yield [24]

$$\begin{aligned} q_{1k}^2 &= Z_k^2 + 2Z_k r_{k+1} s(\theta_k) - 2r_{k+1} r_k c(\theta_k) \\ &\quad + r_{k+1}^2 + r_k^2 \end{aligned} \tag{6}$$

$$\begin{aligned} q_{2k}^2 &= Z_k + Z_k r_{k+1} \left(\sqrt{3}c(\theta_k)s(\psi_k) - s(\theta_k)\right) - r_{k+1}^2 r_k^2 \\ &\quad \left(\frac{\sqrt{3}}{2}s(\theta_k)s(\psi_k) + \frac{3}{2}c(\psi_k) + \frac{1}{2}c(\theta_k)\right) \\ &\quad + r_{k+1}^2 + r_k^2 \end{aligned} \tag{7}$$

$$\begin{aligned} q_{3k}^2 &= Z_k - Z_k r_{k+1} \left(\sqrt{3}c(\theta_k)s(\psi_k) + s(\theta_k)\right) + r_{k+1}^2 r_k^2 \\ &\quad \left(\frac{\sqrt{3}}{2}s(\theta_k)s(\psi_k) - \frac{3}{2}c(\psi_k) - \frac{1}{2}c(\theta_k)\right) \\ &\quad + r_{k+1}^2 + r_k^2 \end{aligned} \tag{8}$$

where $c(.)$ and $s(.)$ are the cosine and the sine functions, respectively. θ_k and ψ_k indicate the pitch and the yaw angle relative to vertebra frame $\Re_k$, respectively. r_k and r_{k+1} are the lower and the upper radius, respectively. Z_k is the distance of the kth inter-vertebra measured along the main axis of the backbone.

For rigid parallel robots, the kinematic model is not strongly nonlinear and it can be solved using numerical techniques like Newton–Raphson method [25]. However, for continuum manipulators, it is mathematically intractable; the numerical techniques are computationally intensive and do not guarantee a closed form solution. In [24], two multilayer perceptrons (MLP) have been used to provide approximated solutions of the inverse kinematic equations.

Assume $\tilde{\theta}_k^1$, $\tilde{\psi}_k^1$, and $\tilde{Z}_k^1$ are the approximated solution of the first section, and $\tilde{\theta}_k^2$,$\tilde{\psi}_k^2$, and $\tilde{Z}_k^2$ are that of the second section. The equation (2) yields (9), where $c(\psi)$ and $s(\psi)$ are the cosine and the sine functions, respectively. $c(.)$ and $s(.)$ indicate the pitch and the yaw angle relative to vertebra frame $\Re_k$, respectively. (X_0, Y_0, Z_0) and (X_{16}, Y_{16}, Z_{16}) are the coordinates of the base and the top frame of the CBHA's backbone, respectively. We notice that the function $h(.)$ is strongly nonlinear, so it is difficult to obtain its inverse

$$\begin{aligned} &\begin{pmatrix} X_{16} \\ Y_{16} \\ Z_{16} \\ 1 \end{pmatrix} \\ &= \left(\prod_{k=1}^{8} \begin{pmatrix} c(\tilde{\theta}_k^1) & s(\tilde{\theta}_k^1)s(\tilde{\psi}_k^1) & s(\tilde{\theta}_k^1)c(\tilde{\psi}_k^1) & 0 \\ 0 & c(\tilde{\psi}_k^1) & -s(\tilde{\psi}_k^1) & 0 \\ -s(\tilde{\theta}_k^1) & s(\tilde{\psi}_k^1)c(\tilde{\theta}_k^1) & c(\tilde{\psi}_k^1)c(\tilde{\theta}_k^1) & \tilde{Z}_k^1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right. \\ &\quad \left. \prod_{k=9}^{16} \begin{pmatrix} c(\tilde{\theta}_k^2) & c(\tilde{\theta}_k^2)s(\tilde{\psi}_k^2) & s(\tilde{\theta}_k^2)c(\tilde{\psi}_k^2) & 0 \\ 0 & c(\tilde{\psi}_k^2) & -s(\tilde{\psi}_k^2) & 0 \\ -s(\tilde{\theta}_k^2) & s(\tilde{\psi}_k^2)c(\tilde{\theta}_k^2) & c(\tilde{\psi}_k^2)c(\tilde{\theta}_k^2) & \tilde{Z}_k^2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right) \\ &\quad \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix} \end{aligned} \tag{9}$$

The proposed forward kinematic model can be implemented in real time despite the important number of matrices. However, we have noticed that the inverse model is computationally intensive [24]. The forward kinematic model being an essential module for a successful robotic system control, we have developed a model-based qualitative approach [26], which has shown good performance relative to models based on quantitative approaches [4], [24]. Moreover, the inverse function can be easily obtained through the DSL scheme [21], [22].

With regard to the kinetic model of the CBHA, only an NN-based model is proposed in this paper.

C. Modeling of the CBHA Based on NN Approach

As long as (1) and (2) evolve on a compact subset C of $R^6 \longrightarrow R^6$ and $R^6 \longrightarrow R^3$, respectively, there is always an NN able to approximate (1) and (2) with a high degree of accuracy [27], since NN are universal approximators [28].

Equations (1) and (2) depend on the time interval $\Delta t(k)$. However, the CBHA reaches the mechanical equilibrium after about 10 s. Therefore, for the sake of simplicity (size of NN), data are recorded only when the robot reaches the mechanical equilibrium ($\Delta t(k) = 10\,\text{s}$) so that the state of the CBHA only depends on the actual inputs (pressures).

Equation (1) of the CBHA can be rewritten as

$$x(k+1) = N_C[W_C(k), u(k)] + \xi_k \tag{10}$$

where $N_C[W_C(k), u(k)]$ is a modified Elman neural networks (MENN), and $W_C(k)$ is the weight matrices. ξ_k is an approximation error with $|\xi_k| \leq \bar{\xi}$, where $\bar{\xi}$ is a small positive number. The MENN learns to reproduce the CBHA's kinetic behaviors. The nonlinear function $g(\zeta)$ changes in time; it is approximated in real time by the adaptive controller.

Equation (2) of the CBHA can be reexpressed by

$$y(k) = N_K[W_K(k), x(k)] + \xi_k \tag{11}$$

where $N_K[W_K(k), x(k)]$ is an MLP, and $W_K(k)$ is the weight matrices. ξ_k is an approximation error with $|\xi_k| \leq \bar{\xi}$, and $\bar{\xi}$ is a small positive number. Equation (11) represents the forward kinematic model of the CBHA.

D. Data Acquisition

To build the learning base, the end-effector position of the CBHA is evaluated experimentally by means of a trilateration system (see Fig. 6) developed in [4]. The test bench includes:

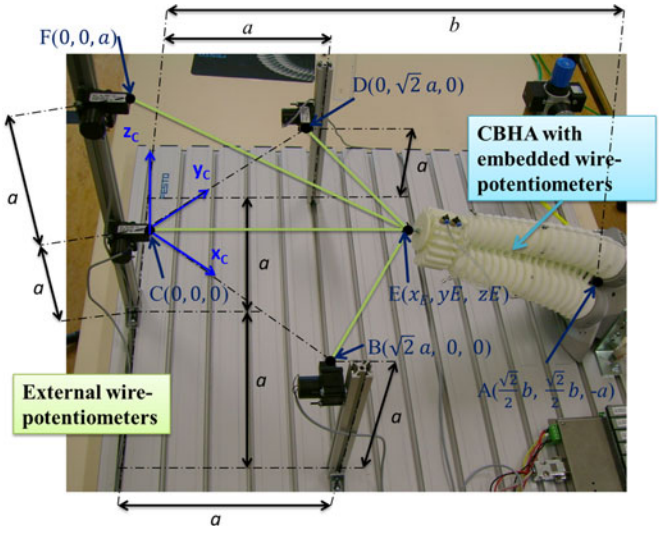

Fig. 6. Trilateration system for learning database building.

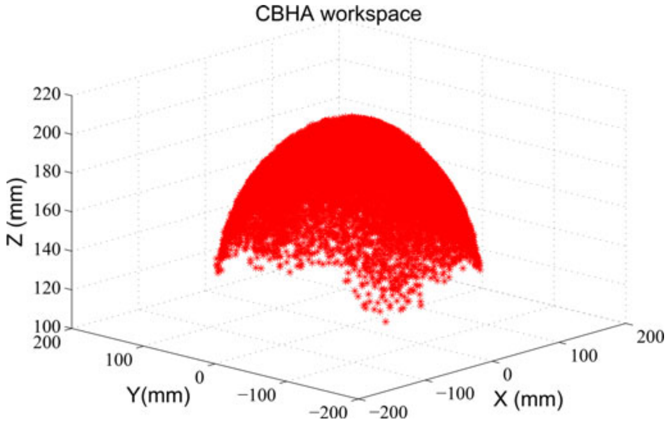

Fig. 7. CBHA's workspace obtained from the learning database.

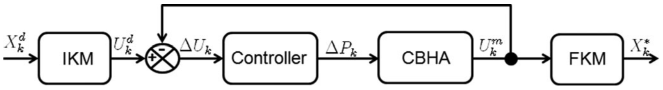

Fig. 8. CBHA: Kinematic control scheme.

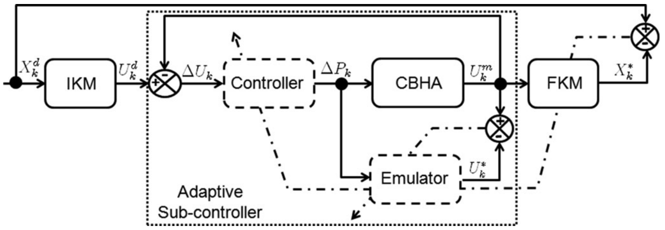

Fig. 9. CBHA: Adaptive kinematic control scheme.

1) one profiled metallic platform;
2) four external proportional potentiometers;
3) six wire-potentiometers placed along the tubes.

From external-potentiometer values and using simple geometric transformations, we can evaluate the end-effector position of the CBHA with an accuracy of approximately ±0.003 m [29]. The learning base is built as follows: The CBHA's posture (wire-potentiometer values) is varying proportionally with the actuated pressures. These pressures are controlled by means of internal PID controllers and range of $[0; 1.5]$ bars. Thus, using a step size of 0.5, each tube can be controlled by one of these values $(0; 0.5; 1; 1.5)$. With six controlled inputs, we get a learning base of $4^6 = 4096$ samples. The resulting CBHA's workspace is represented in Fig. 7.

E. Initialization of NN Parameters and Learning Process

Each NN is trained on the corresponding training set. The learning base is divided as follows: 70% for the training set, 15% for the validation set, and 15% for the test set. The training set is used during the learning phase, and the test set is only used to evaluate the performance of the neuronal models. The validation set is used during the learning phase to avoid the overfitting. For a good generalization of NN models and to avoid overfitting, the early-stopping method is applied for training. The latter requires that after a period of training (epochs) using the training set, the weight matrices of the NN are fixed, and the NN is operated in the forward mode using the validation set. The process is reiterated until the mean square error (MSE) on the validation set reaches its minimum value. The MSE calculated in the training set is minimized by using the back-propagation descent method including the momentum term. Finding the best NN structure is a difficult and unsolved problem [30]; based on the performance achieved on the test set, the NN's parameters are determined. The sigmoid function is used as the activation function.

IV. CBHA Control

In this paper, we focus on the path tracking control of the CBHA's end-effector without physical interactions with the environment. This task can be performed by means of a kinematic control scheme because the CBHA is free from handling tasks. In this paper, the path tracking control of the CBHA's end-effector is performed using an adaptive kinematic control. The CBHA's models obtained offline suffer from imprecisions and do not permit an accurate control because of nonstationary behaviors, hence the adaptive control is used. To illustrate this aspect, a kinematic control (see Fig. 8) and an adaptive kinematic control (see Fig. 9) are developed in this paper.

Adaptive controls based on NN have been widely used in the past for control of nonlinear systems [14], [15], [23],[31], [32]. The algorithm proceeds in two steps. In the first step, an NN (an emulator) learns to reproduce the system's behaviors. In the second step, a controller, another NN learns to control the emulator. The complexity with the CBHA is the difficulty to get an accurate emulator because the latter mimics the behavior that the system had at the time on building of the learning base. However, because of nonstationary behaviors, this model changes over time. In the following, we propose the control

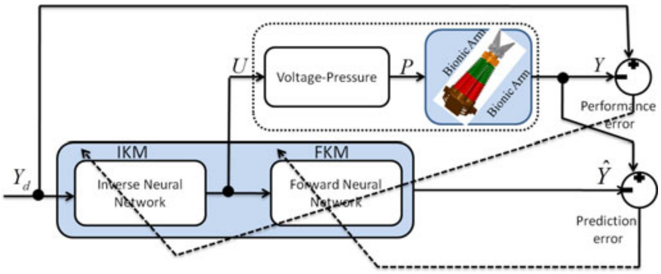

Fig. 10. DSL scheme.

TABLE I
RESULTS ACHIEVED BY EACH NN MODEL ON THE TEST SAMPLES

MSE (Validation set)	MSE (Test set)	
$2.6e^{-5}$	$3.7e^{-5}$	FKM
$7.6e^{-5}$	$1.1e^{-4}$	IKM

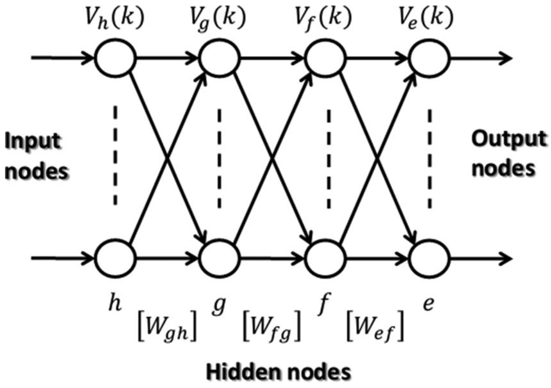

Fig. 11. MLP controller.

scheme represented in Fig. 9 to control the end-effector position of the CBHA. It includes two subcontrollers: A DSL subcontroller and an adaptive subcontroller. The DSL subcontroller comprises the IKM and FKM models as shown in Fig. 10. It deals with kinematic information and remains reliable over time. The adaptive subcontroller system consists of a controller and an emulator. It handles the nonstationary kinetics information and remains uncertain over time. In Fig. 9, X_k^d and X_k^* are the desired and the predicted positions, respectively. U_k^d is the desired vector of voltages, and ΔP_k is the vector of pressures. U_k^m is the measured vector of voltages, and U_k^* is the predicted vector of voltages. The proposed control scheme guarantees the convergence of Cartesian errors. Indeed, if the emulator reproduces the behavior of the CBHA, i.e., $|U_k^m - U_k^*| \leq \epsilon$, ϵ being a small positive number, we can back-propagate the Cartesian errors through the emulator using the back-propagation algorithm. The emulator offers a possibility to translate the Cartesian errors in the controller output. The real CBHA cannot be used because the errors cannot be propagated through it. The convergence is guaranteed, and the computation time for a given input depends on the desired accuracy. The following sections describe each subcontroller.

A. DSL Subcontroller: Kinematic Control

The DSL technique is proposed to elaborate the kinematic control of the CBHA. In DSL approach, the forward NN (FNN) is first learned to approximate the forward kinematic model [18]. Second, a particular inverse solution is obtained by placing the INN and FNN in series. At this stage, the composite learning system can be trained by any supervised learning algorithm (delta rule, back-propagation, etc.). However, the learning algorithm must be constrained so that it does not alter the forward model while the composite system is being trained [18], [21], [22]. In case of the CBHA, the IKM is approximated by the DSL scheme represented in Fig. 10.

$Y_d = [x_d, y_d, z_d]^T$ denotes the desired vector of the end-effector position of the CBHA, and $Y = [x, y, z]^T$ is the real vector of the CBHA's end-effector position. $U = [U_1, U_2, \ldots, U_6]^T$ is the predicted vector of the wire-potentiometer voltages, and $\hat{Y} = [\hat{x}, \hat{y}, \hat{z}]^T$ is the predicted vector of the CBHA's end-effector position. The FNN consists of six inputs (U) and three outputs ($\hat{Y}$), while the INN consists of three inputs (Y_d) and six outputs (U). The prediction error ($Y - \hat{Y}$) and the performance error ($Y_d - Y$) are used for FNN and INN learning, respectively.

A squared penalty term is added to the objective function of the INN to select a particular inverse kinematic function. The cost function yields

$$J = \frac{1}{2} (Y_d - Y)^T (Y_d - Y) + \lambda \frac{1}{2} \|U\|^2 \tag{12}$$

where $\|.\|$ is the Euclidean norm. It has shown that [33] the larger the coefficient λ is, the smaller U becomes. The penalty term λ provides a possibility to effectively control the magnitude of U. Hence, to select a particular inverse solution. In case of the CBHA, the value of $\lambda = 0.001$ has achieved good performance by using the MSE as the convergence criterion on the test set, and an inverse function that minimizes the Euclidean norm of wire-potentiometer voltages has been selected.

The MSE is used as a convergence criterion to select the size of the NN. From an MLP with two hidden layers, the number of neurons in each hidden layer is varied from 2 to 40 with a step of two neurons. Based upon the learning performance achieved on the test set, the NN with two hidden layers of 16 neurons has achieved good performance. Hence, the FNN and the INN are MLP with two hidden layers of 16 neurons. The FNN model has been validated using the test set, and the real measurements obtained by coupling the CBHA with an industrial robot manipulator [26]. While, the INN model has been validated using the samples of the test set and by conducting a real-time implementation on the CBHA [21], [22]. The performance achieves on the test set by the selected NN is reported in Table I. The first column represents the MSE obtained in the validation set, while the second column depicts the MSE obtained in the test set. The last column represents the approximated models.

B. Adaptive Subcontroller: Kinetic Control

The adaptive subcontroller comprises an emulator and a controller. In this section, we first develop the MENN followed by

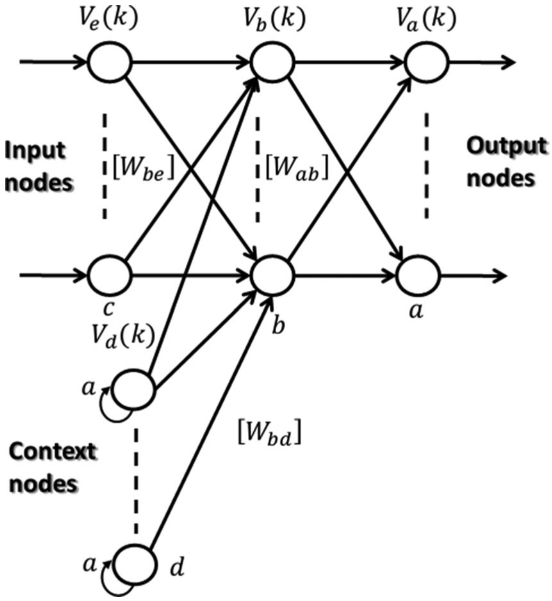

Fig. 12. Modified Elman NN.

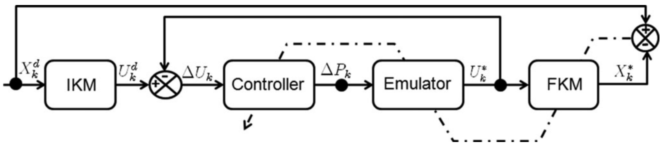

Fig. 13. CBHA: Diagram of offline learning of the controller.

MLP controller. The convergence of the adaptive subcontroller ends the section.

1) Kinetic Model: MENNs: The MENN depicts in Fig. 12 proposed by Pham and Liu[34] is a modified version of the original Elman network introduced by Elman [35]. It differs from the original Elman network by having self-feedback links with fixed gain α in the context nodes. The MENN has been widely used for system identification and control [36]–[39]. The MENN, also called the emulator, is a type of recurrent NN with three layers of neurons: Input nodes, hidden nodes, and output nodes. A hidden node comprises a context nodes used to memorize its previous activations. By using the MSE as convergence criterion, a MENN model composed of 20 nodes in the hidden layer and 20 nodes in the context layer has achieved good performance.

The mathematical equations of the MENN are given by

$$[V_b(k)] = f([w_{bc}][V_c(k)] + [w_{bd}][V_d(k)]) \quad (13)$$

$$[V_d(k)] = \alpha[V_d(k-1)] + [V_b(k-1)] \quad (14)$$

$$[V_a(k)] = g([w_{ab}][V_b(k)]) \quad (15)$$

where $f(x)$ is taken as a sigmoidal function: $f(x) = 1/(1 + e^{-x})$. $g(x)$ is a linear function. $[V_i]$ denotes a N_i-dimensional column vector and $[w_{ij}]$ denotes a $N_i \times N_j$-dimensional matrix.

By using the sum-of-square errors as objective function

$$E(k) = \frac{1}{2}\sum_{a=1}^{N_a}(V_a^d(k) - V_a(k))^2 \quad (16)$$

where V_a^d and V_a are the desired and the MENN outputs, respectively. Differentiating E with respect to w_{ab}, w_{bc}, and w_{bd}, respectively, based on gradient descend method, we get the following update equations for MENN weight matrices:

$$\Delta w_{ab} = \eta_1 \delta_a V_b(k) \quad (17)$$

$$\Delta w_{bc} = \eta_2 \delta_b V_c(k) \quad (18)$$

$$\Delta w_{bd} = \eta_3 \frac{\partial V_b(k)}{\partial w_{bd}} \sum_{a=1}^{N_a}(\delta_a w_{ab}) \quad (19)$$

where $(a = 1, 2, \ldots, N_a,\ \ b = 1, 2, \ldots, N_b)$ and $(c = 1, 2, \ldots, N_c,\ \ d = 1, 2, \ldots, N_d)$. η_1, η_2, and η_3 are the learning rate of $[w_{ab}]$, $[w_{bc}]$, and $[w_{bd}]$, respectively. δ_a, δ_b, and $\frac{\partial V_b(k)}{\partial w_{bd}}$ are expressed as

$$\delta_a = (V_a^d(k) - V_a(k))g_a^{'}(.) \quad (20)$$

$$\delta_b = f_b^{'}(.)\sum_{a=1}^{N_a}(\delta_a w_{ab}) \quad (21)$$

$$\frac{\partial V_b(k)}{\partial w_{bd}} = f_b^{'}(.)V_b(k-1) + \alpha\frac{\partial V_b(k-1)}{\partial w_{bd}}. \quad (22)$$

2) MLPs Controller: The MLP represented in Fig. 11 is used to control the CBHA's kinetics. The same procedure used to select the size of the FNN and INN architectures is applied to design the MLP controller. An MLP with two hidden layers of 16 neurons achieves good performance. It uses control errors as inputs ΔU_k, and provides the control signals in output ΔP_k. The training of the controller is conducted using the learning scheme presented in Fig. 13. Cartesian errors are back-propagated through the forward kinematic model and the emulator. The offline controller obtained is used for the kinematic control (see Fig. 8) and the adaptive kinematic control (see Fig. 9).

Using the sum-of-square errors as the objective function and the gradient descent method, we get

$$E = \frac{1}{2}\sum_{a=1}^{N_a}(e_i)^2 \quad (23)$$

where e_i is the Cartesian error back-propagated through the FKM at the emulator output. The weight matrices of the controller are updated as follows:

$$\Delta w_{ef} = \eta_4 \delta_e V_f(k) \quad (24)$$

$$\Delta w_{fg} = \eta_5 \delta_f V_g(k) \quad (25)$$

$$\Delta w_{gh} = \eta_6 \delta_g V_h(k) \quad (26)$$

where

$$\delta_e = f_e^{'}(.) \sum_{b=1}^{N_b} (\delta_b w_{bc}) \tag{27}$$

$$\delta_f = f_f^{'}(.) \sum_{e=1}^{N_e} (\delta_e w_{ef}) \tag{28}$$

$$\delta_g = f_g^{'}(.) \sum_{f=1}^{N_f} (\delta_f w_{fg}) \tag{29}$$

η_4, η_5, and η_6 are the learning rate of $[w_{ef}]$, $[w_{fg}]$, and $[w_{gh}]$, respectively.

3) Convergence of the Adaptive Subcontroller: The update of (17)–(19), (24)–(26) need a proper choice of the learning rate. The convergence of the MENN is guaranteed if the learning rate η_1, η_2, and η_3 are chosen according to the following considerations:

$$0 < \eta_1(k) < \frac{2}{N_a} \tag{30}$$

$$0 < \eta_2(k) < \frac{8}{N_b N_c |\max\limits_{c}(V_c(k))| |\max\limits_{ab}(w_{ab})|} \tag{31}$$

$$0 < \eta_3(k) < \frac{32(1-\alpha)^2}{N_b N_d |\max\limits_{ab}(w_{ab})|^2}. \tag{32}$$

Since the emulator is used to back-propagate the Cartesian errors, the stable convergence of the controller is guaranteed if the learning rate η_4, η_5, and η_6 are chosen as follows:

$$0 < \eta_4(k) < \frac{32}{N_e N_f |\max\limits_{ab}(w_{ab})| |\max\limits_{bc}(w_{bc})| |\max\limits_{f}(V_f(k))|} \tag{33}$$

$$0 < \eta_5(k) < \frac{128}{N_f N_g |\max\limits_{ab}(w_{ab})| |\max\limits_{bc}(w_{bc})|} \times \frac{1}{|\max\limits_{ef}(w_{ef})| |\max\limits_{g}(V_g(k))|} \tag{34}$$

$$0 < \eta_6(k) < \frac{512}{N_g N_h |\max\limits_{ab}(w_{ab})| |\max\limits_{bc}(w_{bc})|} \times \frac{1}{|\max\limits_{ef}(w_{ef})| |\max\limits_{fg}(w_{fg})| |\max\limits_{h}(V_h(k))|} \tag{35}$$

with $V_h(k) = \Delta U_k$, assuming $U_k^m \cong U_k^x$.

Note that a convergence of the adaptive subcontroller involves the convergence of the overall control system, since the convergence of the DSL subcontroller is guaranteed Table I. In addition, its weight matrices are bounded and kept constant over time.

C. Stability Proof

Considering the positive definite Lyapunov energy function in discrete form which is defined as

$$E(k) = \frac{1}{2} \sum_{a=1}^{N_a} (e_a(k))^2 \tag{36}$$

where $e_a(k) = V_a^d(k) - V_a(k)$.

The variation on the Lyapunov energy function yields

$$\Delta E = E(k+1) - E(k) = \frac{1}{2} \sum_{a=1}^{N_a} [(e_a(k+1))^2 - (e_a(k))^2]. \tag{37}$$

The error of energy function at discrete time $(t+1)$ during the learning process can be expressed by [39]

$$e_a(k+1) = e_a(k) + \sum_{b=1}^{N_b} \frac{\partial e_a(k)}{\partial w_{ab}} \Delta w_{ab} = e_a(k) - \sum_{b=1}^{N_b} \frac{\partial V_a(k)}{\partial w_{ab}} \Delta w_{ab}. \tag{38}$$

Thus, the variation on the Lyapunov function becomes

$$\Delta E = \frac{1}{2} \sum_{a=1}^{N_a} (e_a(k))^2 [(1 - \eta_1(k) \left[\frac{\partial V_a(k)}{\partial W_b}\right]^T \left[\frac{\partial V_a(k)}{\partial w_{ab}}\right])^2 - 1]. \tag{39}$$

ΔE can be rewritten as follows:

$$\Delta E = -\frac{1}{2} \sum_{a=1}^{N_a} (e_a(k))^2 \beta_a(k) \tag{40}$$

with

$$\beta_a(k) = \frac{1}{2}[1 - (1 - \eta_1(k) \| \frac{\partial V_a(k)}{\partial W_a} \|^2)^2] \tag{41}$$

where $\|.\|$ denotes the Euclidean norm. By considering the activation function f of the hidden nodes as a sigmoid type, and g as a linear activation function with $g^{'}(.) = 1$, we have

$$| \frac{\partial V_a(k)}{\partial W_a} | = |V_b(k)| < 1 \tag{42}$$

with $(a = 1, 2, \ldots, N_a,\ b = 1, 2, \ldots, N_b)$. W_a represents the n_a-dimensional vector. According to the definition of the Euclidean norm, we get

$$|| \frac{\partial V_a(k)}{\partial W_a} || < \sqrt{N_a}. \tag{43}$$

Hence, if $\eta_1(k)$ is chosen as $0 < \eta_1(k) < 2/N_a$, we have $\beta_a > 0$ and $\Delta E < 0$. Noticing that $E(k) > 0$, according to the Lyapunov stability theory, it is shown that the training error converges to zero as $t \longrightarrow \infty$. Similarly, the convergences of the emulator and controller learning rates $\eta_2(k)$, $\eta_3(k)$, $\eta_4(k)$, $\eta_5(k)$, $\eta_6(k)$ are demonstrated.

V. Experimental Results and Discussions

In this section, we first describe how the CBHA controller operates online followed by the conducted experiments for

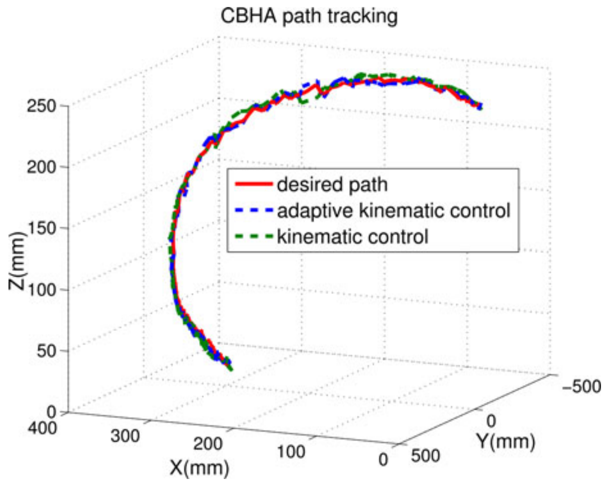

Fig. 14. Experiment 1: Desired path, paths achieved by the kinematic control, and adaptive kinematic control.

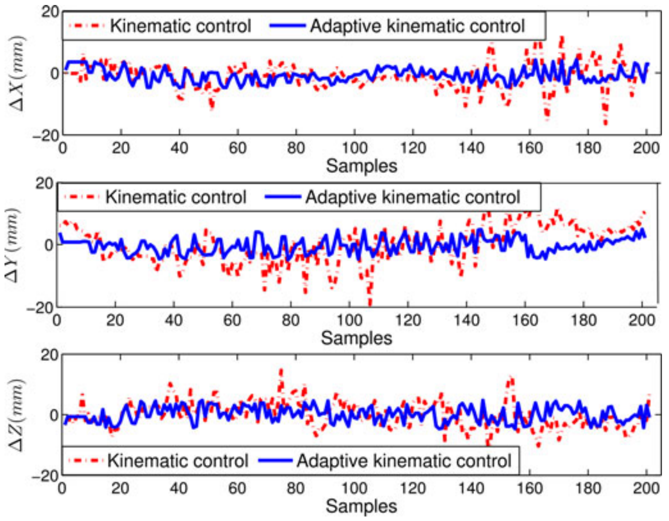

Fig. 15. Experiment 1: Euclidean errors obtained from the kinematic control and adaptive kinematic control. From top to bottom, we have the Euclidean error in *X*-, *Y*-, and *Z*-axes.

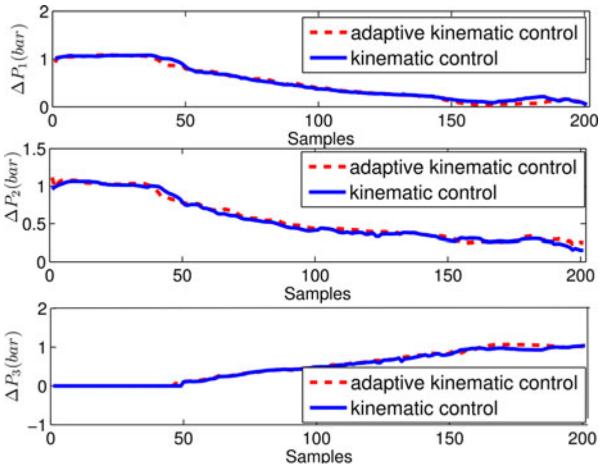

Fig. 16. Experiment 1: Control signals (pressures) for the first section.

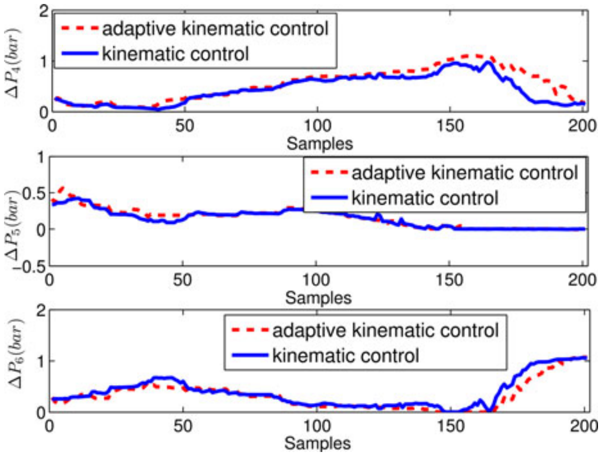

Fig. 17. Experiment 1: Control signals (pressures) for the second section.

kinematic and adaptive kinematic control. The section ends with a discussion.

A. CBHA Controller Setting

In real time, because of nonstationary CBHA's behaviors, the weight matrices of the emulator and the controller are adjusted for each desired target position. The weight matrices of the emulator are first tuned to reproduce the real CBHA's behaviors; thereafter, the weight matrices of the NN controller are tuned to provide a suitable control signal to the CBHA. The emulator and the NN controller weight matrices are tuned such that $|U_k^d - U_k^*| \le e^{-5}$ and $|X_k^d - X_k^*| \le e^{-5}$. It is worth to note that, for a given target, the emulator needs one run to reproduce the CBHA's behavior, while the NN controller needs one to three runs depending on the position of the target in the CBHA's workspace. The number of runs is considered as the number of times that a control signal is applied to the CBHA. The latter needs approximately 10 s to reach the mechanical equilibrium. However, for adaptive tracking, it is not necessary to wait for the mechanical equilibrium of the CBHA. To reach a given target, in the first trial, a time out of 3 s is used to bring the end-effector position of the CBHA in the vicinity of the target. A time out of 1 s is used for the rest of the trials. Thus, the control process spends 3–5 s to reach the targets, which are in the vicinity of the initial position, and 5–7 s for the remote targets. For path tracking, the same process is applied, 3 s for the first trial and 1 s for the rest.

B. Experiments and Results

Two experiments have been performed in real time. The first path (201 points) is generated by using a combination of the pressures in each tube, and the second path (180 points) is generated by moving the end-effector of the CBHA by hand in order to ensure the belonging of each point of the desired path to the CBHA's workspace. The desired path is obtained by using the forward kinematic model developed in our previous

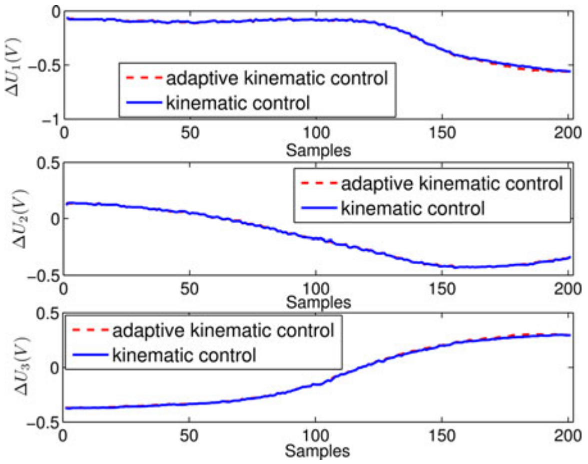

Fig. 18. Experiment 1: Control error $\Delta U(V)$ for the first section.

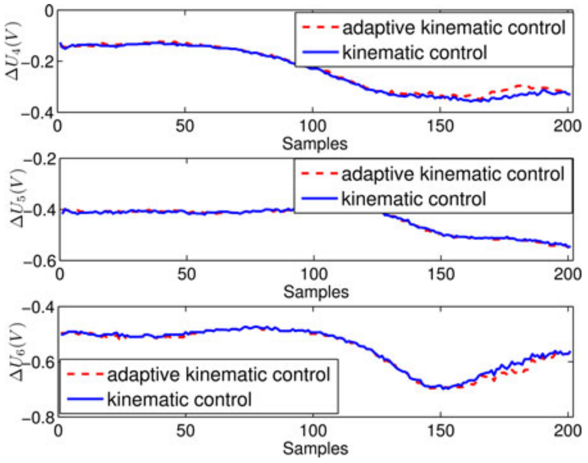

Fig. 19. Experiment 1: Control error $\Delta U(V)$ for the second section.

works [26]. The step size varies from 1 to 5 mm and 1 to 15 mm in the first and the second experiments, respectively. The first experiment evaluates the slow motions of the CBHA where only the first section is considered in the generation of the desired path. The second experiment focused on fast motions of the CBHA where both sections are involved in the generation of the desired path. Through these experiments, we want to assess the robustness of the controller in the absence and presence of redundancies and in case of fast motions.

Regarding the first experiment, the paths achieved by the kinematic and the adaptive kinematic control are presented in Fig. 14. Cartesian errors are depicted in Fig. 15, and the control signals for the first and the second sections are depicted in Figs. 16 and 17, respectively. Control errors ΔU for the first and the second sections are presented in Figs. 18 and 19, respectively. The kinematic control takes 311 s, with a time mean of 1.55 s per point. The adaptive kinematic control takes 960 s, with a time mean of 4.77 s per point. We note a Euclidean error of approximately ±14 and ±5 mm in kinematic and adaptive kinematic control, respectively.

For the second experiment, the paths achieved by the kinematic control and adaptive kinematic control are presented in Fig. 20. Cartesian errors are depicted in Fig. 21, while the control signals for the first and the second sections are represented in Figs. 22 and 23, respectively. Control errors ΔU for the first and the second sections are depicted in Figs. 24 and 25, respectively. The kinematic control takes 503 seconds, with a time mean of 2.79 seconds per point. While the adaptive kinematic control takes 1380 s, with a time mean of 7.667 s per point. We note a Euclidean error of approximately ±20 and ±5 mm in kinematic and adaptive kinematic control, respectively. A real-time implementation can be visualized on the video linked to [40].

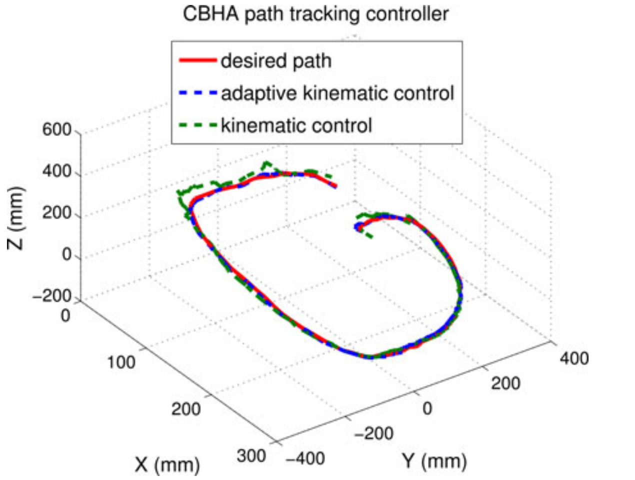

Fig. 20. Experiment 2: Desired path, paths achieved by the kinematic control, and adaptive kinematic control.

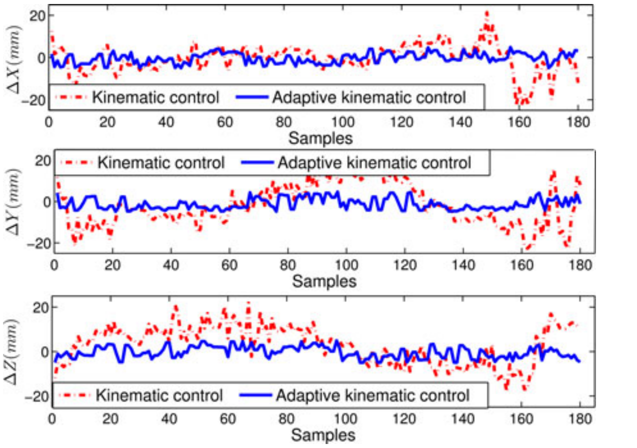

Fig. 21. Experiment 2: Euclidean errors obtained from the kinematic control and adaptive kinematic control. From top to bottom, we have the Euclidean error in *X*-, *Y*-, and *Z*-axes.

By observing the performance achieved by the controllers in each experiment, it is clear that the execution time depends on the step size of the tracked path; the smaller it is, the more rapid is the tracking process. We can conclude that the controller performs better with slow motions of the CBHA. The first experiment shows that for slow motions of the CBHA, there is no significant difference between a kinematic and an adaptive kinematic controller. The kinematic and the adaptive controllers track better the desired path, while the control signals (see Figs. 16 and 17) and the control errors (see Figs. 18

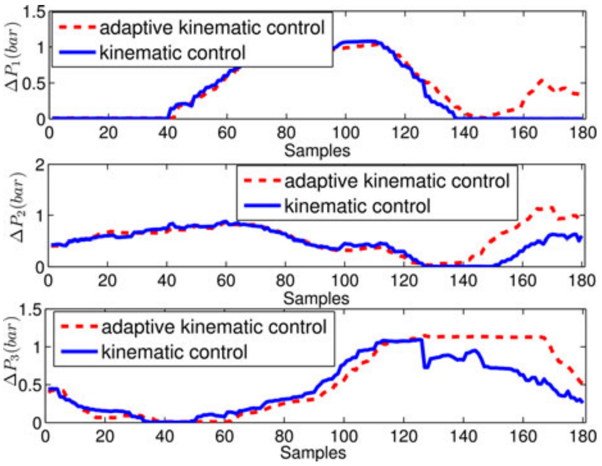

Fig. 22. Experiment 2: Control signals (pressures) for the first section.

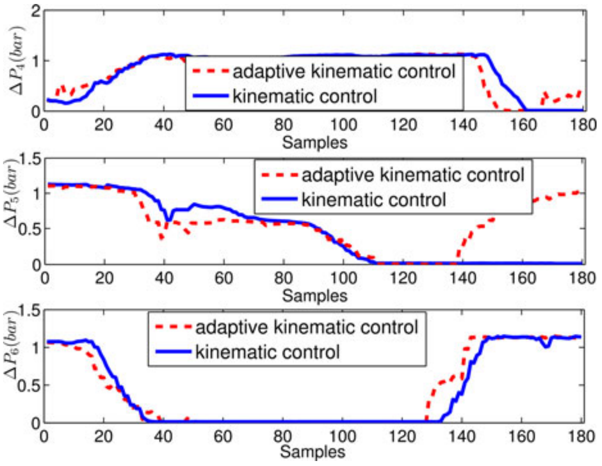

Fig. 23. Experiment 2: Control signals (pressures) for the second section.

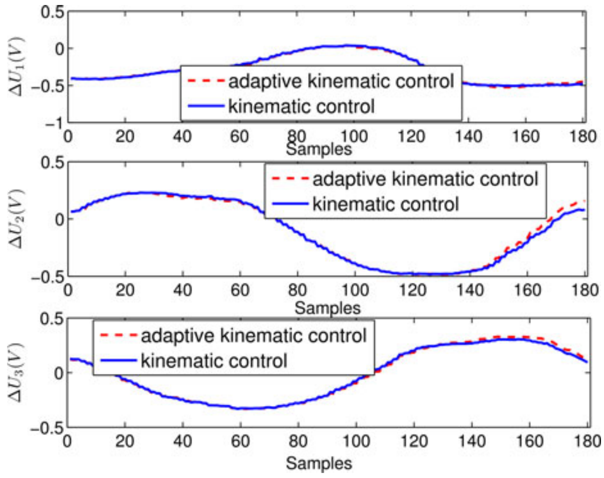

Fig. 24. Experiment 2: Control error $\Delta U(V)$ for the first section.

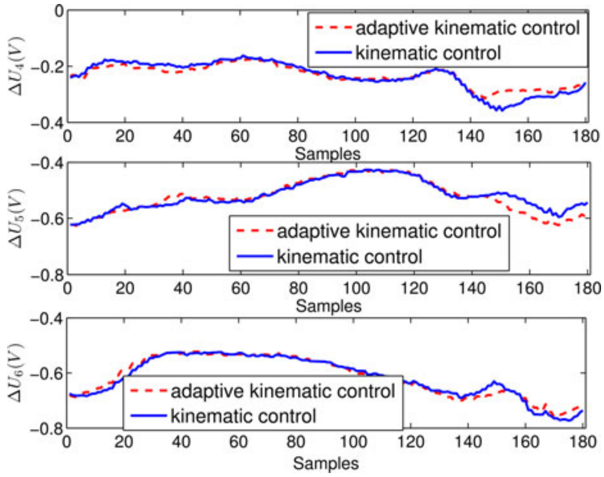

Fig. 25. Experiment 2: Control error Δ *U(V)* for the second section.

and 19) are almost identical. We also note low amplitude of the control signals.

Focusing on robustness, the proposed controller faces redundancies and fast motions of the CBHA, we note that the kinematic controller fails in certain portions of the desired path (see Figs. 14 and 20). The adaptive kinematic controller spends more time to track the desired path, and we observe a significant difference between the control signals (see Figs. 22 and 23) and the control errors (see Figs. 24 and 25). Focusing on this experiment, we can conclude that fast motions of the CBHA and redundancies could be better handled via the adaptive control. A performance that holds even in the presence of redundancies is that both the controllers better handle the first section of the CBHA (see Figs. 18 and 24). Through Figs. 16, 17, 22, and 23, we can see how the adaptive kinematic controller deals with nonstationary behaviors. A significant difference in control signals is observed especially in the second CBHA's section. On some points of the curves, we observe that when the kinematic controller provides zeros or constant pressures, the adaptive kinematic controller handles the nonstationary behaviors by providing the proper pressures.

Remaining controller errors are expected to result from the second section of the robot, i.e., the redundancy problem. It is necessary to note that even by manipulating only the first section, the redundancy problem appears, especially for high pressures. Indeed, when high pressures are supplied to the tubes, the cable that passes through the middle of the backbone reaches its extension limit, and there is a compression of the second section. This usually occurs when nearly equal pressures are applied simultaneously to the three tubes of a same CBHA's section. Remaining controller errors are also expected to result from different modeling errors accumulated during the development of each NN, the memory effect of the polyamide material, and hysteresis phenomenon present in the actuators of the CBHA.

Continuum robots have different characteristics (sizes, nonlinearities, environmental constraints, and so on) and are usually not controlled with the same inputs. Therefore, it is difficult to make a direct comparison between the approaches based only on the results obtained. Table II lists some contributions on modeling and control of continuum robots. Focusing on the forward kinematic of the CBHA, we note a precision of $4\,\mathrm{mm}$ for NN approach [26] and $8\,\mathrm{mm}$ for geometric method [4]. Regarding the IKM of the BHA, the model accuracy is $50\,\mathrm{mm}$ for the Jacobian-based approach [1] and $6\,\mathrm{mm}$ for goal babbling method [16]. An overview of results listed in Table II leads us to

TABLE II
RECENT CONTRIBUTIONS IN CONTROL OF CONTINUUM ROBOTS

Robots	Characteristics	Modeling Method	Kinematic modeling	Target Control
CBHA	Continuous shape, Structure: Soft, two sections, three tubes per section, 17 vertebrae, 3 DoF per inter-vertebra, Pneumatic actuation, Length: 0.50 m.	Qualitative: NN, Melingui *et al.* [17]	Forward (FKM), accuracy: 4 mm, Inverse (IKM), accuracy: 5 mm, Validation: Simulation and real-time experiments.	Path tracking, accuracy: 5 mm.
		Quantitative: Geometrical-based approach Escande *et al.* [4]	Forward (FKM), accuracy: 8 mm, Validation: Simulation and real-time experiments.	Not available
BHA	Continuous shape, Structure: Soft, three sections, three tubes per section, Pneumatic actuation, 30 vertebrae, 3 DoF per inter-vertebra, Length: 1.00 m	Quantitative: Geometrical-based approach, Mahl *et al.* [1]	Forward (FKM), accuracy: 20 mm, Inverse (IKM), accuracy: 50 mm, Validation: Simulation and real-time experiments	Not available
		Qualitative: Goal babbling learning, Rolf and Steil [16]	Forward (FKM), accuracy: 5 mm, Inverse (IKM), accuracy: 6 mm, Validation: Simulation and real-time experiments.	not available
OctArm V	Continuous shape, Structure: Soft, three sections, three control channels per section six actuators for section one, three actuators for sections two and three, Pneumatic actuation, 3DoF per section, Length: 0.83 m.	Quantitative: Geometrical-based approach, Trivedi *et al.* [41]	Forward (FKM), accuracy: 5% of the entire robot length; Inverse (IKM), accuracy : 5% of the entire robot length, Validation: Simulation and real-time experiments.	Not available
		Qualitative: NN, Braganza *et al.*[7]	Forward (FKM), accuracy: not available, Inverse (IKM), accuracy: not available, Validation: Simulation and real-time experiments.	Control of the robot's lengths in real time, accuracy: 5 mm.
Air-Octor	Continuous shape, Structure: Soft, two sections, one central chamber by section, Hybrid actuation (pneumatic + electric), 6 DOF (four Bending and two extension/retraction), Initial length: 0.31 m.	Quantitative: Geometrical-based approach, McMahan *et al.* [42]	Forward (FKM), accuracy: not available, Inverse (IKM), accuracy: not available, Validation: Simulation and real-time experiments.	Not available
		Quantitative: Geometrical-based approach, Jones and Walker [2]	Forward (FKM), accuracy: not available, Inverse (IKM), accuracy: not available, Validation: Simulation and real-time experiments.	Not available
Concentric tube	Continuous shape, Structure: Soft, actuated by mechanisms external to the backbone (e.g., cables, push-rods, etc.).	Quantitative: Geometrical-based approach, Rucker *et al.* [3]	FKM, accuracy: 2.91 mm, Validation: Simulation and real-time experiments.	Not available
		Quantitative: Geometrical-based approach, Anor *et al.* [43]	Validation: Simulation, accuracy: not available.	lengths control, accuracy: 5 mm
Biomimetic Wire-Driven Serpentine	Discontinuous shape, Structure: rigid. It consists of a start node, middle nodes, and end node controlled by two wire cables through servo motors. Length of a middle node: 12.5 mm. Middle node can only bend, no twist and elongation.	Quantitative: Geometrical-based approach, Li and Du [44]	Forward (FKM), accuracy: not available, Inverse (IKM), accuracy: not available, Validation: Simulation and real-time experiments.	Real-time trajectory tracking in free space, accuracy: 3 mm
Tendon-driven	Continuous shape, Structure: Soft, 2-D planar manipulator, Actuation: two stranded steel cables, length: 280 mm. The backbone is segmented into eight equal subsections.	Quantitative: Model based on Jacobian matrix, Yip and Camarillo [45]	Robot pose is given by Cameras, Validation: Simulation and real-time experiments.	path tracking, Accuracy: 3 mm
Elephant's Trunk Manipulator	Continuous shape, Structure: Soft, 32 DOF, four sections, Actuation: Cable servo system, length: 83.82 cm.	Quantitative: Geometrical-based approach, Hannan and Walker [46]	Forward (FKM), accuracy: not available, Inverse (IKM): accuracy: not available, Validation: Simulation and real-time experiments.	Not available

say that qualitative approaches offer good performance, and the accuracy also depends on the characteristics of the robot (sizes, nonlinearities, and so on). For example, concentric tube robots provided good performance compared to their bionics (CBHA, BHA) counterparts.

In summary, we can say that the control scheme based on NN models facilitates the real-time implementation on the CBHA with less cost. The proposed subcontrollers offer the possibility to better evaluate the stability of the controller and guarantee the convergence of the Cartesian errors, comparing to existing controllers [7], [16]. The controller is robust enough to dealt with inherent sensory noise, delays during the control, and the varying actuator ranges. It performs better with slow motions of the CBHA, and takes a reasonable time to track a given path. The performance achieved by the adaptive kinematic controller demonstrate the necessity to adjust in real time the weight matrices of the emulator and controller. Therefore, an exhaustive exploration of the robot workspace becomes unnecessary, and the tracking of any point of the robot's workspace is guaranteed.

VI. CONCLUSION

In this paper, an adaptive control of a class of continuum robot (CBHA) is developed. Two subcontrollers have been implemented in real time to control the end-effector position of the CBHA. The first subcontroller system deals with the stationary behaviors, while the second handles the nonstationary behaviors. It is observed that an adaptive NN controller allows tracking of a desired path in real time with high accuracy. The real-time experiments show the necessity of tuning adaptively the weight matrices of the designed emulator and controller. The convergence of CBHA in closed loop shows the robustness of the controller in presence of sensor noise, disturbing memory phenomenon of the polyamide material, and delay of the pneumatic actuations. The obtained experimental results demonstrate the reliability of the proposed approach.

In the future, a dynamic model of the CBHA will be studied for the purpose of handling tasks.

References

[1] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 935–949, Aug. 2014.

[2] B. A. Jones and I. D. Walker, "Practical kinematics for real-time implementation of continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1087–1099, Dec. 2006.

[3] D. C. Rucker, B. A. Jones, and R. J. Webster, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 769–780, Oct. 2010.

[4] C. Escande, T. Chettibi, R. Merzouki, V. Coelen, and P. P. M., "Kinematic calibration of a multisection bionic manipulator," *IEEE/ASME Trans. Mechatronics,*, vol. 20, no. 2, pp. 663–674, Apr. 2015.

[5] M. Hannan and I. Walker, "The 'elephant trunk' manipulator, design and implementation," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics,* 2001, vol. 1, pp. 14–19.

[6] C. Escande, "Towards modeling of a class of mobile omnidrive—Bionic manipulator robots," Ph.D. dissertation, University of Lille1, Villeneuve-dÁscq, France, 2013.

[7] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1270–1277, Dec. 2007.

[8] M. Giorelli, F. Renda, G. Ferri, and C. Laschi, "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," in*Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2013, pp. 5033–5039.

[9] M. Rolf, "Goal babbling for and efficient bootstrapping of inverse models in high dimensions," Ph.D. dissertation, Faculty Technol., Bielefeld University, Bielefeld, Germany, 2012.

[10] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[11] S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. New York, NY, USA: Springer-Verlag, 2010.

[12] F. L. Lewis, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Boca Raton, FL, USA: CRC Press, 1999.

[13] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback-error-learning neural network for trajectory control of a robotic manipulator," *Neural Netw.*, vol. 1, no. 3, pp. 251–265, 1988.

[14] B. Daachi, T. Madani, and A. Benallegue, "Adaptive neural controller for redundant robot manipulators and collision avoidance with mobile obstacles," *Neurocomputing*, vol. 79, pp. 50–60, 2012.

[15] B. Daachi and A. Benallegue, "A neural network adaptive controller for end-effector tracking of redundant robot manipulators," *J. Intell. Robot. Syst.*, vol. 46, no. 3, pp. 245–262, 2006.

[16] M. Rolf and J. J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1147–1160, Jun. 2012.

[17] A. Melingui, R. Merzouki, and J. Mbede, "Compact bionic handling arm control using neural networks," *Electron. Lett.*, vol. 50, no. 14, pp. 979–981, 2014.

[18] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Sci.*, vol. 16, no. 3, pp. 307–354, 1992.

[19] S. Stitt and Y. F. Zheng, "Distal learning applied to biped robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 137–142.

[20] I. S. Howard and M. A. Huckvale, "Training a vocal tract synthesizer to imitate speech using distal supervised learning," in *Proc. SPECOM*, 2005, pp. 159–162.

[21] A. Melingui, R. Merzouki, J. Mbede, C. Escande, and N. Benoudjit, "Neural networks based approach for inverse kinematic modeling of a compact bionic handling assistant trunk," in *Proc. IEEE 23rd Int. Symp. Ind. Electron.*, 2014, pp. 1239–1244.

[22] A. Melingui, R. Merzouki, J. Mbede, C. Escande, B. Daachi, and N. Benoudjit, "Qualitative approach for inverse kinematic modeling of a compact bionic handling assistant trunk," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 754–761.

[23] T. D. Le and H.-J. Kang, "An adaptive tracking controller for parallel robotic manipulators based on fully tuned radial basic function networks," *Neurocomputing*, vol. 137, pp. 12–23, 2014.

[24] O. Lakhal, A. Melingui, T. Morales, C. Escande, and R. Merzouki, "Forward kinematic of a class of continuum bionic handling arm," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics*, 2014, pp. 1337–1342.

[25] J.-P. Merlet, "Direct kinematics of parallel manipulators," *IEEE Trans. Robot. Autom.*, vol. 9, no. 6, pp. 842–846, Dec. 1993.

[26] A. Melingui, C. Escande, N. Benoudjit, R. Merzouki, and J. B. Mbede, "Qualitative approach for forward kinematic modeling of a compact bionic handling assistant trunk," in *Proc. 19th World Congr. Int. Federation Aut. Control*, Cape Town, South Africa, 2014, pp. 24–29.

[27] J. B. Cabrera and K. S. Narendra, "Issues in the application of neural networks for tracking based on inverse control," *IEEE Trans. Autom. Control,* vol. 44, no. 11, pp. 2007–2027, Nov. 1999.

[28] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[29] C. Escande, P. M. Pathak, R. Merzouki, and V. Coelen, "Modelling of multisection bionic manipulator: Application to robotinoxt," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2011, pp. 92–97.

[30] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Frontmatter and Index*. Hoboken, NJ, USA: Wiley, 2002.

[31] D. H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control Syst. Mag.*, vol. 10, no. 3, pp. 18–23, Apr. 1990.

[32] G. L. Plett, "Adaptive inverse control of linear and nonlinear systems using dynamic neural networks," *IEEE Trans. Neural Netw.,* vol. 14, no. 2, pp. 360–376, Mar. 2003.

[33] W. Wu, H. Shao, and Z. Li, "Convergence of batch BP algorithm with penalty for FNN training," in *Neural Information Processing*. Berlin, Germany: Springer, 2006, pp. 562–569.

[34] D. Pham and X. Liu, "Dynamic system modeling using partially recurrent neural networks," *J. Syst. Eng.* vol. 2, pp. 90–97, 1992.

[35] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[36] J. B. Mbede, X. Huang, and M. Wang, "Robust neuro-fuzzy sensor-based motion control among dynamic obstacles for robot manipulators," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 2, pp. 249–261, Apr. 2003.

[37] D. T. Pham and X. Liu, "Identification of linear and nonlinear dynamic systems using recurrent neural networks," *Artif. Intell. Eng.*, vol. 8, no. 1, pp. 67–75, 1993.

[38] D. Pham and X. Liu, "Training of Elman networks and dynamic system modelling," *Int. J. Syst. Sci.*, vol. 27, no. 2, pp. 221–226, 1996.

[39] X. Shi, Y. Liang, H. Lee, W. Lin, X. Xu, and S. Lim, "Improved Elman networks and applications for controlling ultrasonic motors," *Appl. Artif. Intell.*, vol. 18, no. 7, pp. 603–629, 2004.

[40] A. Melingui. (2014). [Online]. Available: http://www.youtube.com/watch?v=fRun7TX93xY

[41] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 773–780, Aug. 2008.

[42] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-octor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 2578–2585.

[43] T. Anor, J. R. Madsen, and P. Dupont, "Algorithms for design of continuum robots using the concentric tubes approach: A neurosurgical example," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 667–673.

[44] Z. Li and R. Du, "Design and implementation of a biomimetic wire-driven underactuated serpentine manipulator," *Trans. Control Mech. Syst.*, vol. 1, no. 6, pp. 250–258, 2012.

[45] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 880–889, Aug. 2014.

[46] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45–63, 2003.

Achille Melingui received the Ph.D. degree in automation and industrial computing from the University of Science and Technology of Lille 1, Villeneuve d'Ascq, France, and from the university of Yaoundé 1, Cameroon, in 2014.

He is currently a Lecturer in the Department of Electrical and Telecommunications Engineering at the École Nationale Supérieure Polytechnique of the University of Yaoundé I, Yaoundé, Cameroon. His research interests include modeling and control of mobile robots and continuum manipulators, neural networks, and fuzzy logic.

Othman Lakhal received the Diploma of Engineering in automation and electrical engineering from the University of Lille 1, Villeneuve d'Ascq, France, in 2013. He is currently working toward the Ph.D. degree at the CRIStAL laboratory of the University of Science and Technology of Lille 1, Lille.

His research interests include modeling, identification, and control of mobile robots and continuum manipulators.

Boubaker Daachi received the B.S. degree in computer science from the University of Sétif, Setif, Algeria, in 1995, the M.S. degree in robotics from the University of Paris 6, Paris, France, in 1998, and the Ph.D. degree in robotics from the University of Versailles, Versailles, France, in 2000.

He was an Associate Professor of robotics and computer science at the university of Paris Est Créteil, Paris, from September 2003 to August 2014. He evolved as a Researcher at CNRS-AIST JRL, Tsukuba, Japan, from September 2014 to August 2015. His research interests include adaptive control of complex systems, soft computing, and ambient intelligence. Application fields are robotics and pervasive and distributed systems. He has published more than 60 papers in scientific journals, books, and conference proceedings. He has been involved in the organizing committees of some national and international events.

Jean Bosco Mbede received the Ph.D. degree in industrial automation option robotics from the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in July 2000, and the HDR degree in sciences and industrial engineering from the University of Orléans, Orléans, France, in July 2005. He also received the Associate Professor Qualification by the 61st Section (Informatics Engineering, Automatic Control, and Signal Processing) and 60th Section (Mechanics, Mechanical Engineering, and Civil Engineering) of the National Universities Committee in France in February 2006 and May 2009, respectively.

He is currently a Senior Lecturer in the Department of Electrical and Telecommunications Engineering at the École Nationale Supérieure Polytechnique of the University of Yaoundé I, Yaoundé, Cameroon. His research interests include mechatronics, robotics, surgical and imaging systems, robust neurofuzzy motion planning, and biomechanics. He is a Reviewer for several international journals such as IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART B, etc.

Dr.-Habil. Mbede is a Senior Member of the American Institute of Aeronautics and Astronautics.

Rochdi Merzouki received the Ph.D. degree in robotics and automation from the University of Versailles, Versailles, France, in 2002.

He is currently a Professor of control and automation with the laboratory CRIStAL UMR-CNRS 9189 at Polytech'Lille, University of Science and Technology of Lille, Villeneuve d'Ascq, France. His research interests include system of systems, modeling, and supervision of mechatronics systems applied to robotics and intelligent transport. He is the author of many contributions published in journals, conferences, and books. He is a Project Manager of EU project InTraDE.