Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Learning-Based Approaches for Forward Kinematic Modeling of Continuum Manipulators

I. Mahamat Loutfi * **A.H. Bouyom Boutchouang** ** **A. Melingui** ** **O. Lakhal** *** **F. Biya Motto** * **R. Merzouki** ***

* *Department of Physic's, Faculty of Sciences, University of Yaounde I, Yaounde 8390, Cameroon (loutfimrane@gmail.com.*
** *Department of Electrical and Telecommunications Engineering, ENSP, University of Yaounde I, Yaounde 8390, Cameroon (achillemelingui@gmail.com)*
*** *CRIStAL Laboratory, CNRS-UMR, Villeneuve d'Ascq 59655, France*

Abstract: Forward kinematic model (FKM) is an essential module in the control law design of manipulator robots. Unlike rigid manipulators where it can be easily established, it remains a real challenge for their continuum counterparts. Model-based and learning-based approaches are commonly used for the forward kinematic modeling of continuum manipulators. Model-based approaches generally lead to imprecise FKM models due to several modeling assumptions, while learning-based approaches generally yield acceptable performance. However, the choice of an appropriate learning model remains a challenging task. In the framework of the forward kinematic modeling of continuum manipulators, this paper proposes an experimental and structural comparative study of the commonly used learning models, namely the multilayer perceptron (MLP), radial based functions (RBF), support vector regression (SVR), and Co-Active adaptive neuro-fuzzy inference system (CANFIS). The Compact Bionic Handling Assistant (CBHA) robot is used as an experimental platform and the predictions of the different learning models are compared respectively to a high precision motion capture system. According to the comparative study, we noted better accuracy for SVRs, rapid convergence for RBFs, and a good compromise between learning time and accuracy for MLPs. CANFIS offers accuracy close to that of SVRs but with much shorter learning time.

Keywords: Multi-section Continuum Manipulators, kinematic modeling, Multilayer Perceptron (MLP), Radial Basis Function (RBF), Support Vector Regression (SVR), and Co-Active Adaptive Neuro-Fuzzy Inference System (CANFIS).

1. INTRODUCTION

Machine learning is a tool that is increasingly used in the study of new generations of robots that combine mechanical flexibility, material elasticity and lightness. Such robots mimic the behavior of living beings as octopus arms, muscles, tentacles , elephant trunks, and cephalopod members. These properties make them uniquely suited for a large number of applications, including surgery, underwater operations, and exploration.

Continuum manipulators are designed with flexible materials and inherit non-stationary behaviors due to the hysteresis effects of some actuators, viscoelasticity and the loss of certain physical properties of the materials that they are made from. Unlike their rigid counterparts, which are made of rigid bodies, and from which the FKM can easily be derived, these characteristics make their kinematics difficult to establish. Contributions on the FKM of continuum manipulators can be summarized into two main approaches, namely, model-based and learning-based approaches. Model-based approaches involve the establishment of a kinematic model based on approximate assumptions about the physical structure of the manipulator robot. Regarding learning-based approaches, the parametric space is divided into several groups according to the robot's operating modes. A mathematical model derived from the learning algorithms makes it possible to establish a relationship between effects (expert observation, sensor measurements, and statistical data) and causes (input references).

As regards to model-based approaches, various methods have been proposed to provide a solution to FKM of continuum manipulators. To establish the forward kinematics of a continuum manipulator composed of 4 sections, Hannan and Walker (2003) considered that the bending motion of a section of the manipulator could be better described using a constant curve (Webster III and Jones (2010)). Godage et al. (2011) used shape functions that incorporate the manipulator structure to model a multi-section variable-length continuum arm. Escande et al. (2012) used the constant curvature approach to model a two-section bionic manipulator mimicking the elephant's trunk. The same approach was used by Rolf and Steil (2012) for the forward kinematics of the bionic handling assistant (BHA). However, the assumption of constant curvature is not always verified, especially for conical continuum manipulators, and the aforementioned contributions generally lead to imprecise FKM models. For improving the accuracy of kinematic models, variable curvature has been implemented by some researchers. Thus, Chirikjian and Burdick (1994) considered the backbone curve of the robot as a series of a very large number of links and proposed a shape of a particular mathematical curve to solve the forward kinematics. However, the proposed method is limited to continuum robots with a limited set of shapes and movements. Jones and Walker (2006) used the same approach by formulating the trunk kinematics problem as a series of substitutions applied to a modified homogeneous transformation matrix computed using a Denavit-Hartenberg (D-H) type approach. Mahl et al. (2014) proposed a variable curvature to model a BHA manipulator as a series of constant curvature (CC) segments. However, although these contributions improve the performance of the kinematic models obtained in a significant way, they remain poor.

Learning-based techniques have also been explored to provide a solution to forward kinematics of continuum manipulators. These techniques are more attractive because they can tolerate a wide range of uncertainties and handle high nonlinearities. Thus, Giorelli et al. (2013) used a fully-connected feed-forward neural network (FNN) able to determine the tip position of a soft manipulator moving in three-dimensional space according to the forces applied to the cables which drive this non-constant curvature manipulator. Melingui et al. (2014) proposed two neural network architectures, namely RBF and MLP, to provide a solution to CBHA forward kinematics. However, although all these learning models lead to more or less precise models than model-based approaches, choosing the right learning model remains a challenging task.

This paper presents an experimental and structural comparative study on the performance of RBF, MLP, CANFIS and SVR models in the framework of forward kinematic modeling of multi-section continuum manipulators. A two-section continuum manipulator called the CBHA was used as a case study. The samples of the learning database are collected using a high precision motion capture system. The comparison is conducted based on correlation analysis (Taylor (1990)). A correlation coefficient representing the degree of linear association between two variables. The closer the R coefficient is to 1, regardless of the direction, the stronger the existing association, indicating a linear relationship between the two variables. Besides, the performances are also evaluated based on learning time, mean square error on the test set, and the remaining Cartesian errors.

The rest of the paper is organized as follows. Some recall theories of the different used learning models are presented in Section II. The implementation of the different learning models and a discussion are provided in Section III. Conclusion and future work follow in section IV.

2. LEARNING-BASED APPROACHES

Artificial neural networks are mathematical models represented by a set of simple computing units and linked to each other by a system of connections (Cheng and Titterington (1994)). Neural networks have the advantage of possessing many useful properties and capabilities of non-linearity, input-output mapping, adaptability, fault tolerance, very large scale integration, and implementation (Haykin (1994)). Thanks to these characteristics, neural networks have been used to solve many regression (Li et al. (2011)) and classification (Bischof et al. (1992)) problems, and have great success in the field of artificial intelligence in general, and the field of robotics in particular.

Support vector machines (SVMs) have long been used to solve data classification problems. The solution function is given as a linear combination of some data samples, called support vectors. SVRs are considered extensions of SVMs; they showed good generalization ability for various problems of function approximation and time series prediction (Vapnik (2013); Smola and Schölkopf (2004)). SVRs transform the regression problem into a quadratic programming (QP) problem such that global solutions can be obtained by QP solvers and regression problems can be solved without the local minimum problems.

However, the choice of appropriate learning models remains difficult because the sample data pairs collected for learning do not provide information about the learning model to use. In the framework of the forward kinematic modeling of continuum robots, four commonly used regression approaches are implemented in this work. The following subsections recall some basic concepts of these learning models.

2.1 Multilayer Perceptron (MLP)

A Multilayer perceptron (MLP) neural network architecture composed of an input layer, one or more hidden layers and an output layer. The input layer receives no signal from the other layers while the output layer does not send any signal to another layer. The hidden layers transfer information from the input layer to the output layer through a set of interconnected units called neurons. In this structure, each layer is equipped with processing elements and each element is fully interconnected by weighted connections to the elements of the next layer. It is worth noting that the neurons in each layer work in parallel to generate the different outputs. The input layer receives the data; this data is processed in the hidden layers and displayed at the output layer. All connections between the different layers are assigned a weight that is fitted using a learning algorithm. Non-linearities are introduced into the architecture using activation functions such as sigmoid, hyperbolic tangent or ReLU. At any neuron i, the excitation level $v_i(x)$ is calculated as follows:

$$v_i(x) = \sum_{j=1}^{n} \beta_{ij} x_j \tag{1}$$

where x_i is the input of neuron i, n is the number of the neurons in the previous layer and β_{ij} is the associated weight. The activation function $y_i = f(v_i)$ is applied at the excitation level to determine its output. The commonly used transfer function for the hidden layers is the sigmoid function:

$$y_i = \left(1 + e^{-v_i}\right)^{-1} \tag{2}$$

The learning process consists of computing the error of the global network output to update all weighted connections. The error δy_i between the desired value and the real output of each neuron i is given by:

$$\delta\beta_{ij} = -\alpha \frac{\partial \delta y_i}{\partial \beta ij} \tag{3}$$

where α is the learning rate and β_{ij} is the weight between input x_i and neuron i.

2.2 Radial Basis Function (RBF)

In the literature, radial basis function (RBF) neural networks have been used in various fields of application (Ghorbani et al. (2016)) and have become an alternative to MLPs neural network. The architecture of the RBF neural network differs from that of its MLP counterpart in the hidden layer. The fundamental structure of the RBF neural network consists of an input layer, a hidden layer of n elements and an output layer. Thus, the input neurons represented by the input variables z_j transfer their signals to the hidden layer. Then, a kernel function ϕ_j (a Gaussian function in most cases) characterized by a center c_j and a width σ_j is assigned to each neuron of the hidden layer. Due to the non-linearity of the hidden layer elements, all the outputs of the hidden layer can be combined in a linear way to obtain the network output, this significantly improves data processing. The expressions of the kernel and the output function are given respectively by:

$$\phi_j\left(\|z - c_j\|\right) = \exp\left(-\frac{1}{2}\left(\frac{\|z - c_j\|}{\sigma_j}\right)^2\right) \tag{4}$$

$$f\left(z\right) = \sum_{j=1}^{p} \lambda_j \phi_j\left(\|z - c_j\|\right) \tag{5}$$

where p and λ_j are respectively the number and the weight of the radial functions.

2.3 Support Vector Regression (SVR)

Support Vector Machines (SVM) proposed by (Vapnik (2013)) are a very specific class of algorithms, characterized by the usage of kernels, absence of local minima, the sparseness of the solution and capacity control obtained by acting on the margin, or the number of support vectors. SVM was originally developed to solve classification problems (Vapnik (2013); Smola and Schölkopf (2004)). However, with the introduction of the Vapnik ε-insensitive loss function, the SVM was extended to solve regression estimation problems. In this case, it simply calls Support Vector Regression (SVR). One of the advantages of SVM and SVR as the part of it is that it can be used to avoid difficulties of using linear functions in the high dimensional feature space and the optimization problem is transformed into dual convex quadratic programs. In the regression case, the loss function is used to penalize errors that are greater than the threshold.

The objective of the linear support vector regression problem with ε-insensitive loss function introduced by (Vapnik (2013)) is finding a function $f : R^d \to R$ such as

$$f(\mathrm{x}) = \phi(\mathrm{x})w + b \tag{6}$$

where $\mathrm{x} \in \mathrm{R^d}$ denotes the dot product in R^d and ϕ is a non-linear transformation from R^d to the high-dimensional space R^h.

Given a set of independent and identically distributed samples, i.e., $\{(x_i, y_i)\}_{i=1}^n$ where $x \in R^d$ is an incoming vector and $y \in R$ an observable output. Single-ended SVR is intended to solve the problem of finding the parameters $w \in R^h$ and $b \in R$ that minimize the objective function defined by:

$$\min_{w \in R^h, b \in R} \frac{1}{2}\|w\|^2 + C\sum_{i=0}^{n} L_\varepsilon(y_i - f(x_i)) \tag{7}$$

where $L_\varepsilon = \max\{0,\ |f(x_i) - y_i| - \varepsilon\}$ is the ε-insensitive error loss function, the parameter C measures the trade-off between generalization ability and accuracy in the training data, and parameter ε defines the degree of tolerance to errors. The linear combination of learning samples in the transformed space with absolute errors equal to or greater than ε is the final solution (w and b).

The problem of multidimensional regression must be generalized in the case of a multi-output system. This case was addressed in (Melingui et al. (2018)) where the outputs of dimension M can be seen as several single-output cases where (6) is extended as follows:

$$f^j(x) = w^j \cdot \phi(x) + b^j\ ,\ j = 1, 2, \dots, M \tag{8}$$

with w^j and b^j define the parameters of the regressor. These regression problems can be solved by minimizing the following objective function:

$$\min_{W,B} \frac{1}{2}\sum_{j=1}^{M}\left\|w^j\right\|^2 + C\sum_{i=1}^{n} L(\mu_i) \tag{9}$$

with $L(.)$ the extended loss function of Vapnik ε-insensitive based on the norm L_2 (Sánchez-Fernández et al. (2004)).

The interested reader is referred to (Sánchez-Fernández et al. (2004); Melingui et al. (2014)) for a detailed description of the SVR.

2.4 Co-Active Adaptive Neuro-Fuzzy Inference Systems (CANFIS)

Co-Active adaptive neuro-fuzzy inference systems (CANFIS) are an extension of the traditional Adaptive Neuro-Fuzzy Inference Systems (ANFIS) that are intended to address multi-output multi-input (MIMO) systems. Its topology consisted of five (5) layers is depicted in Fig. 1. Each layer of this architecture intends to perform an inference fuzzy system step of the Takagi-Sugeno type. The idea is to use a mathematical concept in a fuzzy logic framework to map the relationship between the outputs and the inputs of a given system. In the case of a multi-output system, the use of CANFIS topology becomes essential, in that it allows better handling of the dependencies between the different outputs of the system and makes each output estimate less vulnerable to noise.

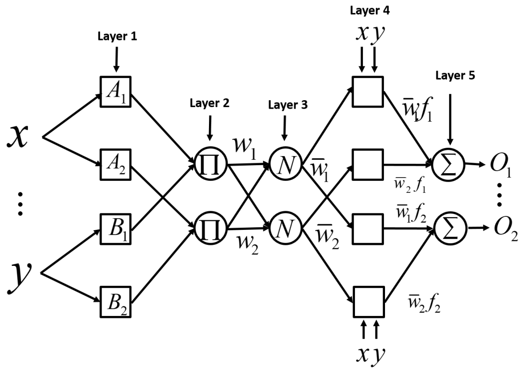

Fig. 1. CANFIS Network Architecture

3. FORWARD KINEMATIC MODEL: CBHA PLATFORM

This section focuses on the use of MPL, RBF, CANFIS, and SVR models for the estimation of the forward kinematic model of a particular continuum platform, namely the Compact Bionic Handling Assistant (CBHA) Manipulator robot. The section starts with a description of the experimental platform, follows by the learning database building. It ends with the presentation of the results obtained and a discussion.

3.1 Description of the Compact Bionic Handling Assistant Manipulator

The CBHA robot is a two-section continuum manipulator inspired by the elephant's trunk and made from polyamide materials. It consists of two bending sections, each equipped with three pneumatic actuators (tubes), a wrist axis and a compliant gripper as shown in Fig. 2. The pressure supply in each tube is controlled by a PID regulator. The elongations of the different tubes are provided by six-wire potentiometers placed along each tube. The elastic deformation of the CBHA results in movements with an infinite number of degrees of freedom. The properties of the polyamides and the pneumatic actuators that compose it make it a challenging platform as well for modeling as for control. It mainly inherits a compliance and memory effect from the properties of polyamide materials and a hysteresis effect caused by its pneumatic actuators.

3.2 Learning database building

The performance of the different models is conducted by comparing the poses provide by each model with those obtained from an optitrack motion capture system (OptiTrack (2018)). The experiment setup is depicted in Fig. 2. The motion capture system, composed of four cameras, can track a rigid body motion with an accuracy of ± 0.1 mm. The reflective markers are attached to the end effector of the robot, and the latter is moved throughout its workspace. The exploration of the CBHA's workspace is performed by providing pressure into the six CBHA's tubes. The input pressure vectors are obtained by discretizing the pressure variable with a step of 0.5 bar. The manipulator having 6 tubes, we define a pressure input vector $[p_1, p_2, \ldots, p_6]$, with $p_i \in [0.0, 0.5, 1.0, 1.5, 2.0]$ $(i = 1, 2, \ldots, 6)$ assigned to each tube. Thus, the different pressure variations in the tubes give a database of $5^6 = 15625$ samples. To collect these samples, the robot operates for approximately 13 hours, since each sample is recorded after a complete mechanical equilibrium of the robot that takes approximately 3 seconds. This delay is due to the pressure regulation system in the different tubes because the regulation is performed tube by tube. For each input pressure vector, the end-effector pose of the robot and the corresponding wire-potentiometer voltages are recorded.

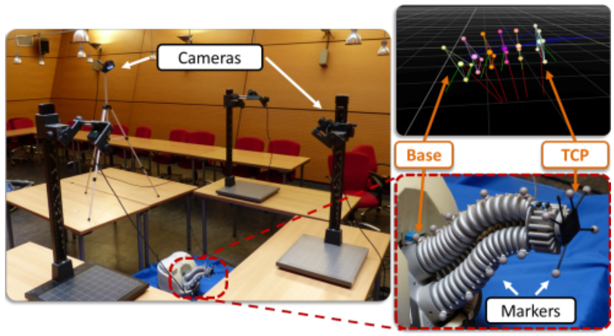

Fig. 2. OptiTrack Motion Capture System

For a good generalization of the different models and to avoid over-fitting, the early stopping method is applied for the training (Prechelt et al. (1994)). The method requires that after some epochs using the training set, the model's weight matrices are fixed, and the latter is used in forward mode using the validation set. The process is repeated until the mean square error (MSE) of the validation set reaches its minimum value. The training was stopped when the stopping criterion is met or when the maximum number of epochs is reached. Among the models that meet the stopping criteria, the model with the minimum MSE is selected as the best. The database is normalized in the range $[0.1, 0.9]$ and is randomly divided in the ratio $70 : 15 : 15$ for the training, validation and test set, respectively. The wire-potentiometer voltages are considered as inputs and the position and orientation variables are considered as outputs. The learning process of each model performed in MATLAB software using an Intel(R) Core(TM) $i7-7500U$ CPU processor at 2.70 GHz and 2.90 GHz.

There are methods for choosing the training parameters for these different architectures. However, these methods do not always guarantee optimal performance. Therefore, many training operations were performed on each architecture with different values of the training parameters and only the best network for each architecture was selected.Thus, an MLP of two layers with 28 neurons each and sigmoid activation function in the hidden layer and linear activation function in the output layer achieved satisfactory performance. Regarding RBF, Gaussian radial basis functions with a width of 0.5 have been used, and a RBF model consists of 60 neurons in the hidden layer achieved satisfactory performance. Concerning CANFIS, a model consists of 15 Gaussian membership functions for each input variable and 15 Takagi and Sugeno's type rules achieved satisfactory performance. The fuzzy C-means clustering was used for the initialization of Gaussian function centers and a hybrid learning rule (Jang (1993)) which combines the gradient method and the least squares estimate (LSE) were used to identify network parameters. The following SVR's parameters $C = 4000$, $\epsilon = 0.4e^{-6}$, and $\sigma = 1.2$ achieved satisfactory performance. Where C is a fixed constant which controls the trade-off between the training error and the regularization term, ϵ is the insensitive error, and σ is the standard deviation of Gaussian kernel functions. A backward variable selection by block deletion is implemented for variable selection.

Table 1 lists the results obtained for each model. The mean square error (MSE) obtained during the training on the validation data, the learning time, as well as the parameters of each model are provided. One notes MSE of 5.3481×10^{-5}, 7.0691×10^{-5}, 2.3383×10^{-5}, and 4.5232×10^{-6} for MLP, RBF, CANFIS, and SVR, respectively. The learning times are approximately $01h45mn$, $01h15mn$, $02h30mn$, and $03h41mn$ for MLP, RBF, CANFIS, and SVR, respectively. The absolute values of the maximum Cartesian errors are listed in the Table 2. The maximum position errors of $4.164mm$, $6.039mm$, $2.79mm$ and $3.638mm$ are noticed for MLP, RBF, CANFIS, and SVR, respectively, while the maximum orientation errors are $11.604°$, $15,148°$, $5.667°$, and $6.976°$ respectively. The correlation coefficients representing the degree of linear association between the estimated values and the measured values variables are reported in Table 3 and Table 4. A coefficient of approximately 0.99 is obtained for each model. A comparison with a commonly used analytical approach, called Constant Curvature (CC) approximation, was made. Overall, the performances achieved by each learning model are satisfactory in terms of pose tracking of the end-effector of the CBHA robot. Each model can follow the final effector of the CBHA robot with a maximum position error of 6mm and a maximum orientation error of 15°. The R coefficient is also greater than 0.99 demonstrating the strong linear relationship between the estimated values and the measured values. We can conclude that learning-based approaches can effectively model the highly non-linear relationship between the configuration space and the workspace of the CBHA robot without issues of geometric singularities in stretched positions generally encountered in model-based approaches.

Table 1. Characteristics of MLP, RBF, SVR and CANFIS Architectures

Models	Parameters	MSE	learning time
MLP	28 neurons	$5.3481e^{-5}$	$01h45$
RBF	60 neurons, $\sigma = 0.5$	$7.0691e^{-5}$	$01h15$
CANFIS	15 If-then rules	$2.3383e^{-5}$	$02h30$
SVR	$C = 4000$, $\sigma = 1.2$	$4.5232e^{-6}$	$03h41$

Table 2. Absolute value of Euclidean errors with position in (mm) and orientation in degree

Models	X	Y	Z	ψ	θ	ϕ
MLP	2.679	3.219	4.164	11.604	2.224	10.876
RBF	5.960	5.302	6.039	15.148	3.183	12.096
SVR	1.406	1.191	2.796	4.870	0.856	5.667
CANFIS	2.033	3.116	3.638	6.852	1.746	6.976
CCA	6.102	5.294	7.212	22.801	4.432	12.341

Table 3. linear correlation coefficient Position R

Models	$X(mm)$	$Y(mm)$	$Z(mm)$
MLP	0.99994	0.99991	0.99937
RBF	0.99971	0.99945	0.998
SVR	0.99998	0.99997	0.99974
CANFIS	0.99995	0.99995	0.99941
CCA	0.99969	0.9996	0.9978

Regarding individual performance, SVR model achieves the best performance, follows by CANFIS, MLP, and RBF, respectively. However, RBF has the best convergence time, followed by MLP, CANFIS, and SVR, respectively.

In addition to the advantages common to the different models, in particular, the best approximation and generalization capabilities, each of these models has advantages and disadvantages. RBFs have the advantage, for example, of having a simple network architecture. They can model any non-linear function using only one hidden layer. This eliminates some design decisions such as the number of hidden layers. The convergence time is very fast; but also the linear transformation of the output layer can be optimized using traditional linear modeling techniques, which are fast and do not suffer from problems such as local minima encountered in MLP learning techniques. However, the generalization capabilities of RBFs are lower than MLPs because the approximation is local and the extrapolation beyond known data is poor. SVRs offer a global solution and if the solution is not unique, then the set of global solutions is convex because the regression problem is transformed into a convex optimization problem. This is a real advantage over neural networks, which face local minimum problems and for this reason, may not be robust across the entire learning base. However, a common disadvantage of non-parametric techniques such as SVRs is the lack of transparency of the results. For example, it is difficult to represent the estimated variable by a simple parametric function of the input variables. They are not suitable for large data sets, and methods to reduce the number of support vectors are generally used. Besides the very high computation time, a practical difficulty is the selection of kernel function parameters. CANFIS has the advantage that the parameters of the inference system are coded as weights in the neural network and can, therefore, be optimized using powerful learning methods. However, the computation time is very high, and the learning process requires mastery of both the basic concepts of fuzzy logic and neural networks.

The results on the CCA show that the learning-based approaches have efficient that the model-based approaches

Table 4. linear correlation coefficient Orientation R

Models	$\psi°$	$\theta°$	$\phi°$
MLP	0.99967	0.99982	0.99899
RBF	0.9941	0.9998	0.99812
SVR	0.9994	0.99995	0.99969
CANFIS	0.99883	0.99988	0.99908
CCA	0.9903	0.99969	0.9997

since these latter don't take in account the unmodelable and uncertain dynamics.

To summarize, we can conclude that learning-based approaches are a good alternative for forward kinematic modeling of continuum robots. Whatever the learning model used, the performances achieved are satisfactory. However, RBFs and SVRs are recommended for small databases and MLPs and CANFISs for large databases.

4. CONCLUSION

In this paper, a comparative study of four learning-based techniques in the framework of the forward kinematic modeling of multi-section continuum manipulators has been proposed. MLP, RBF, SVR, and CANFIS models were implemented for forward kinematic modeling of a two-section continuum manipulator. The predictions of the different topologies were compared respectively to a high precision motion capture system. Many performance criteria have been used for performance assessment of the different learning-based techniques, namely, the correlation coefficient, learning time, mean square error, and the remaining Cartesian errors. In the view of the results obtained, better accuracy of SVR model has been observed followed by CANFIS, RBF, and MLP, respectively. SVR model is the most time-consuming relative to learning time, followed by CANFIS, MLP and RBF models, respectively. However, the calculation time of the different regressors obtained after the learning phase is almost the same; the RBF being the least and CANFIS the most time-consuming. In future work, it will be interesting to assess the performance of these learning models in the framework of inverse kinematic modeling of continuum manipulators.

REFERENCES

Bischof, H., Schneider, W., and Pinz, A.J. (1992). Multispectral classification of landsat-images using neural networks. *IEEE transactions on Geoscience and Remote Sensing*, 30(3), 482–490.

Cheng, B. and Titterington, D.M. (1994). Neural networks: A review from a statistical perspective. *Statistical science*, 2–30.

Chirikjian, G.S. and Burdick, J.W. (1994). A modal approach to hyper-redundant manipulator kinematics. *IEEE Transactions on Robotics and Automation*, 10(3), 343–354.

Escande, C., Merzouki, R., Pathak, P.M., and Coelen, V. (2012). Geometric modelling of multisection bionic manipulator: Experimental validation on robotinoxt. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, 2006–2011. IEEE.

Ghorbani, M.A., Zadeh, H.A., Isazadeh, M., and Terzi, O. (2016). A comparative study of artificial neural network (mlp, rbf) and support vector machine models for river flow prediction. *Environmental Earth Sciences*, 75(6), 476.

Giorelli, M., Renda, F., Ferri, G., and Laschi, C. (2013). A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 5033–5039. IEEE.

Godage, I.S., Guglielmino, E., Branson, D.T., Medrano-Cerda, G.A., and Caldwell, D.G. (2011). Novel modal approach for kinematics of multisection continuum arms. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 1093–1098. IEEE.

Hannan, M.W. and Walker, I.D. (2003). Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *Journal of field robotics*, 20(2), 45–63.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Jang, J.S. (1993). Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665–685.

Jones, B.A. and Walker, I.D. (2006). Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1), 43–55.

Li, C., Bovik, A.C., and Wu, X. (2011). Blind image quality assessment using a general regression neural network. *IEEE Transactions on Neural Networks*, 22(5), 793–799.

Mahl, T., Hildebrandt, A., and Sawodny, O. (2014). A variable curvature continuum kinematics for kinematic control of the bionic handling assistant. *IEEE transactions on robotics*, 30(4), 935–949.

Melingui, A., Ahanda, J.J.B.M., Lakhal, O., Mbede, J.B., and Merzouki, R. (2018). Adaptive algorithms for performance improvement of a class of continuum manipulators. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1531–1541.

Melingui, A., Escande, C., Benoudjit, N., Merzouki, R., and Mbede, J.B. (2014). Qualitative approach for forward kinematic modeling of a compact bionic handling assistant trunk. *IFAC Proceedings Volumes*, 47(3), 9353–9358.

OptiTrack (2018). Motion capture systems. URL `https://optitrack.com/motion-capture-robotics/`. Visited on 06 December 2018.

Prechelt, L. et al. (1994). Proben1: A set of neural network benchmark problems and benchmarking rules.

Rolf, M. and Steil, J.J. (2012). Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3440–3446. IEEE.

Sánchez-Fernández, M., de Prado-Cumplido, M., Arenas-García, J., and Pérez-Cruz, F. (2004). Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems. *IEEE transactions on signal processing*, 52(8), 2298–2307.

Smola, A.J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3), 199–222.

Taylor, R. (1990). Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, 6(1), 35–39.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Webster III, R.J. and Jones, B.A. (2010). Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13), 1661–1683.