2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

A Feed-Forward Neural Network Learning the Inverse Kinetics of a Soft Cable-Driven Manipulator Moving in Three-Dimensional Space

Michele Giorelli, *Student Member*, IEEE, Federico Renda, *Student Member*, IEEE,
Gabriele Ferri, *Member*, IEEE and Cecilia Laschi, *Senior Member*, IEEE

Abstract— **In this work we address the inverse kinetics problem of a non-constant curvature manipulator driven by three cables. An exact geometrical model of this manipulator has been employed. The differential equations of the mechanical model are non-linear, therefore the analytical solutions are difficult to calculate. Since the exact solutions of the mechanical model are not available, the elements of the Jacobian matrix can not be calculated. To overcome intrinsic problems of the methods based on the Jacobian matrix, we propose for the first time a neural network learning the inverse kinetics of the soft manipulator moving in three-dimensional space. After the training, a feed-forward neural network (FNN) is able to represent the relation between the manipulator tip position and the forces applied to the cables. The results show that a desired tip position can be achieved with a degree of accuracy of 1.36% relative average error with respect to the total arm length.**

I. INTRODUCTION

A soft robot [1] [2] is a continuum robot [3] with intrinsic compliant capabilities, which depend on its soft structure [4]. The soft structure can be made of soft materials (e.g. rubber, silicone) and/or actuators. The actuators can form part of the structure, such as EAPs (Electro Active Polymer), pneumatic actuators, SMAs (Shape Memory Alloys), and cables, which are pulled externally by motors.

For soft manipulators, the solution of the inverse kinematics problem of soft manipulators is essential to generate paths in the task space in order to perform grasping or other tasks. The solution of the inverse kinematics is a hard task, because modelling of soft manipulators is carried out with continuum mechanics approach. In particular, non-constant curvature manipulators requires a continuum parametrization of backbone. Therefore, the direct and the inverse kinematics of this kind of manipulators can be solved by using integral resolutions, which are time consuming. On the other hand, the increasing attention towards conical shape manipulators [5]–[7] is justified by their intrinsic grasping capabilities. The conical shape manipulator is characterised by the curvature that increases along the arm, thus providing a spiral-like configuration, which allows for grasping of objects with a wider range of size.

A direct kinematics model [8] maps the joint space into the task space. In continuum robotics the joint space can be described as a set of actuation variables, e.g. the pressures imposed to pneumatic actuators, the forces applied to cables, etc. Therefore, a more general definition of the joint space can be the *actuation space*. The task space can be expressed by a set of mechanical deformations (curvature, torsion, shear and longitudinal strains) that describes the arm configuration. The inverse kinematics problem requires the determination of the actuation-variable values in order to move the continuum manipulator to a given shape configuration. In general, it is not easy to define the admissible shape configurations of a specific manipulator, thus it is useful to express the task space in terms of position and orientation of the tip, when a map exists between position/orientation of the tip and the shape configuration.

The inverse kinematics algorithms for continuum manipulators can follow either an analytical or a numerical approach. A modal approach was proposed by Chirikjian *et al.* [9], who solved the inverse problem of hyper-redundant robot in a closed analytical form by using a particular set of curvature functions. Alternatively, when it is impossible to describe a backbone curvature by this particular set, he proposed to use a numerical method. For continuum manipulator, just like for rigid robots, we can differentiate the direct kinematics model to find the velocity kinematics model, i.e. the linear transformation of actuation variables velocity into the tip velocity. The linear transformation is realized by the Jacobian matrix. To solve the inverse kinematics problem several numerical methods based on Jacobian matrix are implemented, as proposed by Walker *et al.* [5], Simaan *et al.* [10] and Giorelli *et al.* [11]. Camarillo *et al.* [12] found a linear transformation of cables displacement into backbone configuration, therefore he used the minimum-norm and minimax algorithms to solve the inverse problem. The latter method avoids slack problem (negative tensions), but it is an iterative algorithm, which has a high computational cost that makes it less attractive than the minimum-norm algorithm for real-time applications.

In traditional robotics, neural networks (NNs) have been widely employed for learning the inverse kinematics. The neural network models are used to solve the inverse kinematics problem of non-redundant [13]–[15] and redundant [16], [17] manipulators. Several architectures of neural networks are implemented, as FNNs [13]–[17], recurrent NNs [18], radial basis function NNs, and Kohonen NNs [19]. A lot of learning algorithms are employed, like the back-propagation (BP) algorithm [17], the generalized delta learning rule [14], the covariance matrix adaptation evolution strategy (CMA-ES) [15], the neuro-evolution of augmenting topologies

This work was supported in part by the European Commission in the ICT-FET OCTOPUS Integrating Project, under contract no.231608.

M. Giorelli, F. Renda, G. Ferri, C. Laschi are with The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy, (corresponding author: Michele Giorelli, phone +39-050-883396; e-mail: michele.giorelli at sssup.it)

(NEAT) [15], etc.

In continuum robotics, NNs were used to control manipulators. Braganza *et al.* [20] implement the low-level joint controller of a soft extensible manipulator by using a FNN to compensate for the dynamic uncertainties.

A kinematics model of a manipulator describes the motion of the arm without considering the forces that cause the motion. On the other hand, a *kinetics* model describes a relation between the motion of the manipulator and the actuation forces that drive it. In the case of soft cable-driven manipulators, the direct model maps the cable tensions onto the tip position, thus we call it *direct kinetics model*, while the *inverse kinetics* (IK) *model* calculates the cable tensions in order to move the tip to the desired position. In our previous work [11], a preliminary study has been made on the effectiveness of the Jacobian method for solving the inverse kinetics of a non-constant curvature manipulator driven by two cables in two-dimensional space. The results have shown good performance in terms of convergence, however the computational cost of the Jacobian matrix and the accuracy in a real scenario should be improved by using a more sophisticated equipment. The computational cost of the inverse kinetics algorithm was improved by using a feed-forward neural network with a two-cable manipulator [21].

In this work, we tackle the inverse kinetics problem of a non-constant curvature manipulator driven by three cables. The cables are arranged at an angle of $2\pi/3$ rad. An exact geometrical model of this manipulator is employed. A deep analysis of the model equations is conducted in order to evaluate the feasibility of Jacobian methods. The non-linear terms in the equations of the mechanical model do no permit to find the analytical solutions required for the calculation of the Jacobian matrix elements. Therefore, a different approach based on a feed-forward neural network has been adopted for learning the IK. In particular, in this preliminary stage the data set for FNN training is generated in simulation by using the mechanical model of the conical shape manipulator. The mechanical model is also used to test the performance of the trained FNN. In conclusion, we demonstrate that FNNs are suitable models for solving accurately the IK problem of soft non-constant curvature manipulators driven by cables in three dimensional space.

II. Direct Kinetics Model

The kinetics model of the soft manipulator is developed exploiting a Cosserat approach, namely considering the arms as an infinite series of infinitesimal rigid bodies, which can rotate independently from the rotations of its closest neighbours.

A. Parametrization of the manipulator backbone

We refer to each rigid body by the parameter s. In order to describe the arm configuration, for each arclength parameter s a local reference frame $(\boldsymbol{t}(s), \boldsymbol{n}(s), \boldsymbol{b}(s))$ is considered (Fig.1) [22]. In particular, the unit vector $\boldsymbol{t}(s)$ is tangential to the manipulator backbone, it points in the direction of the tip, and it is perpendicular to the cross-section plane, because the Eulero-Bernoulli beam hypothesis has been adopted, i.e. the shear strains are neglected. The unit vector $\boldsymbol{n}(s)$ lies on the cross section and it is perpendicular to $\boldsymbol{t}(s)$. Finally, the unit vector $\boldsymbol{b}(s)$ is defined as $\boldsymbol{b}(s) = \boldsymbol{t}(s) \times \boldsymbol{n}(s)$. The local frame $(\boldsymbol{t}(s), \boldsymbol{n}(s), \boldsymbol{b}(s))$ and the position vector of the centre of mass $\boldsymbol{u}(s)$ respectively describe the orientation and position of the section s. A complete description of the manipulator configuration in 3-D space is obtained by using the following formulas [22] [23]:

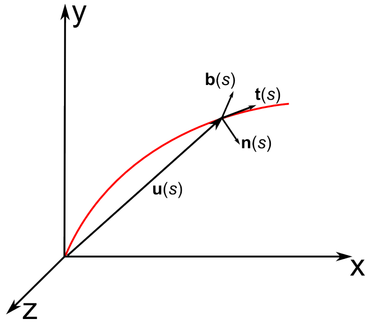

Fig. 1. The parametrization of the manipulator backbone, where $(\boldsymbol{t}(s), \boldsymbol{n}(s), \boldsymbol{b}(s))$ is the reference local frame at section s, (x, y, z) is the base reference frame, and $\boldsymbol{u}(s)$ is the position vector of the section s respect to the base frame.

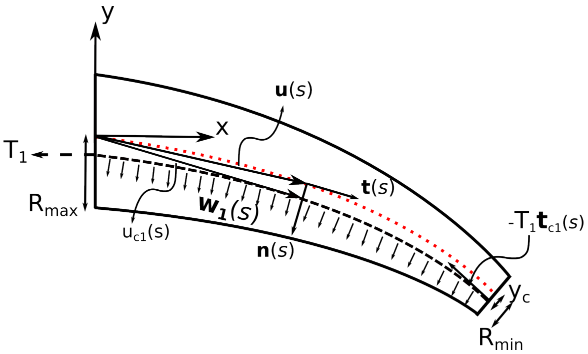

Fig. 2. The structure of the conical shape manipulator driven by three cables. To sake of simplicity only cable T_1 is shown. Pulling the cable T_1 the point load $-T_1\boldsymbol{t}_{c1}$ and the distributed load $\boldsymbol{w}_1(s)$ generate the bending movement.

$$\frac{d\boldsymbol{t}}{ds} = k(s)(1+q(s))\boldsymbol{n}(s) - \xi(s)(1+q(s))\boldsymbol{b}(s) \quad (1)$$

$$\frac{d\boldsymbol{n}}{ds} = -k(s)(1+q(s))\boldsymbol{t}(s) + \tau(s)(1+q(s))\boldsymbol{b}(s) \quad (2)$$

$$\frac{d\boldsymbol{b}}{ds} = \xi(s)(1+q(s))\boldsymbol{t}(s) - \tau(s)(1+q(s))\boldsymbol{n}(s) \quad (3)$$

$$\frac{d\boldsymbol{u}}{ds} = (1+q(s))\boldsymbol{t}(s) \quad (4)$$

Here the functions $k(s)$ and $\xi(s)$ are the curvatures with respect to $\boldsymbol{b}(s)$ and $\boldsymbol{n}(s)$, $\tau(s)$ is the torsion with respect to $\boldsymbol{t}(s)$, and $q(s)$ is the longitudinal strain along the arm.

In the following section we describe the statics model of the conical shape manipulator, i.e. the relation between the cable tensions T_1, T_2 e T_3 and the deformation variables $(k(s), \xi(s), \tau(s), q(s))$.

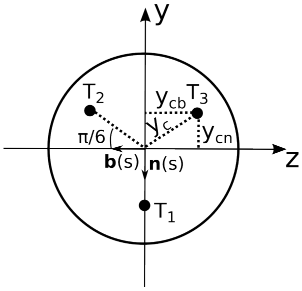

Fig. 3. Base view of the conical shape manipulator driven by three cables T_1,T_2 and T_3. The cables are arranged at an angle of $2\pi/3$ rad. They are parallel to the midline at the same distance y_c. The projections of the distance y_c onto the axis $\boldsymbol{b}(s)$ and $\boldsymbol{n}(s)$ are respectively y_{cb} and y_{cn}.

B. Statics of the conical shape manipulator

The conical shape manipulator has three cables embedded in the structure arranged at an angle of $2\pi/3$ rad (Fig. 3). When the cables are pulling, some stress are exerted on the structure realizing the deformations $k(s)$, $\xi(s)$, $\tau(s)$ and $q(s)$. Unlike the work [24], in this paper the cables are considered parallel to the midline at the same distance y_c (Fig. 3). This geometric change simplifies the mechanical model equations, without modifying the manipulator property of realizing a spiral-like configuration. The positions of the cables are defined as follows:

$$\begin{aligned}\boldsymbol{u}_{c1}(s) &= \boldsymbol{u}(s) + y_c\boldsymbol{n}(s) && (5)\\ \boldsymbol{u}_{c2}(s) &= \boldsymbol{u}(s) - y_{cn}\boldsymbol{n}(s) + y_{cb}\boldsymbol{b}(s) && (6)\\ \boldsymbol{u}_{c3}(s) &= \boldsymbol{u}(s) - y_{cn}\boldsymbol{n}(s) - y_{cb}\boldsymbol{b}(s) && (7)\end{aligned}$$

where y_{cn} and y_{cb} represent respectively the projection of the distance y_c onto the axis $\boldsymbol{n}(s)$ and $\boldsymbol{b}(s)$) (Fig. 3).

Moreover, we introduce the radius $R(s)$ of the section s:

$$R(s) = \left(\frac{R_{min} - R_{max}}{L}\right) s + R_{max} \tag{8}$$

where L is the arm length.

When the cables are pulled, they exert a point load at the spot where they are anchored, and a distributed load along the cable length. The point load is equal in magnitude to the cable tension and tangent to it $-T_i\boldsymbol{t}_{ci}(s)(Ti > 0)$ (Fig. 2), where $\boldsymbol{t}_{ci}(s)$ is the tangent vector to the cable equal to $d\boldsymbol{u}_{ci}/dS_c$, and S_c represents the arclength parametrization of the cable. The distributed load is centripetal and proportional to the curvature of the cable $T_i d\boldsymbol{t}_{ci}/dS_c$ [12] (Fig. 2), where $i \in \{1, 2, 3\}$ identifies one of the three cables.

As a consequence, the equilibrium equations for a cross section s are:

$$\begin{aligned}\boldsymbol{m}_i(s) &= -T_i\boldsymbol{t}_{ci}(L) \times [\boldsymbol{u}(s) - \boldsymbol{u}_{ci}(L)] + \\ &\quad + \int_s^L T_i \frac{d\boldsymbol{t}_{ci}}{d\sigma} \times [\boldsymbol{u}(s) - \boldsymbol{u}_{ci}(\sigma)]\, d\sigma && (9)\\ l_i(s) &= -T_i\boldsymbol{t}_{ci}(L) \cdot \boldsymbol{t}(s) + \\ &\quad + \boldsymbol{t}(s) \cdot \int_s^L T_i \frac{d\boldsymbol{t}_{ci}}{d\sigma}\, d\sigma && (10)\end{aligned}$$

where $l_i(s)$ and $\boldsymbol{m}_i(s)$ are respectively the internal longitudinal stress and the internal torque of the robot arm. Notice that, in order to deduce equations (9) (10) the following equivalence has been used:

$$\frac{d\boldsymbol{t}_{ci}}{dS_c} \times (...) dS_c = \frac{d\boldsymbol{t}_{ci}}{d\sigma}\frac{d\sigma}{dS_c} \times (...) \frac{dS_c}{d\sigma} d\sigma = \frac{d\boldsymbol{t}_{ci}}{d\sigma} \times (...) d\sigma$$

After some mathematical manipulation we derive the expression for $d\boldsymbol{m}_i(s)/ds$ and $l_i(s)$:

$$\begin{aligned}\frac{d\boldsymbol{m}_i}{ds} &= -T_i\boldsymbol{t}_{ci}(s) \times (1 + q(s))\boldsymbol{t}(s) + \\ &\quad - T_i\frac{d\boldsymbol{t}_{ci}}{ds} \times [\boldsymbol{u}(s) - \boldsymbol{u}_{ci}(s)] && (11)\\ l_i(s) &= T_i\boldsymbol{t}_{ci}(s) \cdot \boldsymbol{t}(s). && (12)\end{aligned}$$

In these equations $\boldsymbol{t}_{ci}(s)$ is equal to $d\boldsymbol{u}_{ci}/ds$, where we have assumed s to be a good approximation of the arclength parametrization of the cables (S_c) and of the midline. Another approximation introduced in the model is the linearity of the constitutive equations (13)(14), that guarantees a significant simplification:

$$EA(s)q(s) = l(s) \tag{13}$$

$$\begin{pmatrix} GI(s) & 0 & 0 \\ 0 & EJ(s) & 0 \\ 0 & 0 & EJ(s) \end{pmatrix} \begin{pmatrix} \tau(s) \\ \xi(s) \\ k(s) \end{pmatrix} = \boldsymbol{m}(s) \tag{14}$$

Here $l(s) = l_1(s) + l_2(s) + l_3(s)$, $m(s) = m_1(s) + m_2(s) + m_3(s)$, E is the Young modulus, $G(s) = E/2(1 + \nu)$ is the shear modulus and ν the Poisson ratio. $A(s)$ is the area of the section equal to $\pi R^2(s)$, $J(s) = (\pi/4)R^4(s)$ is the moment of inertia about the axis $\boldsymbol{n}(s)$ and $\boldsymbol{b}(s)$, and $I(s) = 2J(s)$ is the moment of inertia about the axis $\boldsymbol{t}(s)$.

Using the equilibrium equations (11) (12) and the constitutive equations (13) (14), in particular the derivative with respect to s of the (14), we derived the statics model of the conical shape manipulator, which maps the cable tensions T_1, T_2 and T_3 onto the deformation variables $(k(s), \xi(s), \tau(s), q(s))$:

$$\begin{aligned}\tau'(s) &= \frac{A_t(s)}{Z_t(s)}\tau + \frac{B_t}{Z_t(s)}k\xi + \frac{C_t}{Z_t(s)}\xi + \\ &\quad + \frac{D_t}{Z_t(s)}k + \frac{E_t}{Z_t(s)}(\xi^2 - k^2) && (15)\\ \xi'(s) &= \frac{A_\xi(s)}{Z(s)}\xi + \frac{B_\xi(s)}{Z(s)}k + \frac{C_\xi(s)}{Z(s)}\tau + \\ &\quad + \frac{D_\xi(s)}{Z(s)}\tau k + \frac{E_\xi(s)}{Z(s)}\tau\xi && (16)\\ k'(s) &= \frac{A_k(s)}{Z(s)}k + \frac{B_k(s)}{Z(s)}\xi + \frac{C_k(s)}{Z(s)}\tau + \\ &\quad + \frac{D_k(s)}{Z(s)}\tau\xi + \frac{E_k(s)}{Z(s)}\tau k && (17)\end{aligned}$$

$$\begin{aligned}
\tau(L) &= 0 && (18)\\
\xi(L) &= \frac{y_{cb}(T_3 - T_2)}{EJ(L)} && (19)\\
k(L) &= \frac{y_c T_1 - y_{cn}(T_2 + T_3)}{EJ(L)} && (20)\\
q(s) &= \frac{T_1(y_c k(s) - 1)}{EA(s)} + \\
&\quad + \frac{(T_2 + T_3)(y_{cn}k(s) - y_{cb}\xi(s) - 1)}{EA(s)} && (21)
\end{aligned}$$

where:

$$\begin{aligned}
A_t(s) &= GI^{'}(s)\\
B_t &= (y_{cb}^2 - y_{cn}^2)(T_2 + T_3) - y_c^2 T_1\\
C_t &= y_{cn}(T_2 + T_3) + y_c T_1\\
D_t &= y_{cb}(T_2 + T_3)\\
E_t &= y_{cn} y_{cb}(T_2 + T_3)\\
Z_t(s) &= GI(s) + y_c^2(T_1 + T_2 + T_3)\\
A_\xi(s) &= -EJ^{'}(s)[EJ(s) + y_c^2 T_1 + y_{cn}^2(T_2 + T_3)]\\
B_\xi(s) &= -EJ^{'}(s) y_{cn} y_{cb}(T_2 + T_3)\\
C_\xi(s) &= -y_{cn} y_{cb}^2 (T_2 + T_3)^2 - [y_{cn}(T_2 + T_3) + \\
&\quad + y_c T_1][EJ(s) + y_c^2 T_1 + y_{cn}^2(T_2 + T_3)]\\
D_\xi(s) &= y_{cn}^2 y_{cb}^2 (T_2 + T_3)^2 + \\
&\quad - EJ^{'}(s)[EJ(s) + y_c^2 T_1 + y_{cn}^2(T_2 + T_3)]\\
E_\xi(s) &= -EJ(s) y_{cn} y_{cb}(T_2 + T_3)\\
A_k(s) &= -EJ^{'}(s)[EJ(s) + y_{cb}^2(T_2 + T_3)]\\
B_k(s) &= -EJ^{'}(s) y_{cn} y_{cb}(T_2 + T_3)\\
C_k(s) &= -y_{cb}(T_2 + T_3)[EJ(s) + y_{cb}^2(T_2 + T_3)] + \\
&\quad - [y_{cn} y_{cb}(T_2 + T_3)][y_{cn}(T_2 + T_3) + y_c T_1]\\
D_k(s) &= [EJ(s) + y_{cb}^2(T_2 + T_3)][y_c^2 T_1 + \\
&\quad + y_{cn}^2(T_2 + T_3)] - y_{cn}^2 y_{cb}^2 (T_2 + T_3)^2\\
E_k(s) &= EJ(s) y_{cn} y_{cb}(T_2 + T_3)\\
Z(s) &= [EJ(s) + y_{cb}^2(T_2 + T_3)][EJ(s) + \\
&\quad + y_{cn}^2(T_2 + T_3) + y_c^2 T_1]
\end{aligned}$$

Here the symbol $'$ indicates the derivative with respect to s.

To find the direct kinetics model, i.e. the relation between the cable tensions $\boldsymbol{c}_t = (T_1, T_2, T_3)$ and the tip position $\boldsymbol{p}_{ee} = \boldsymbol{u}(L) = (x_{ee}, y_{ee}, z_{ee})$, firstly, the system of the ordinary differential equations (ODEs) (15)-(20) has to be solved numerically in order to find the deformation functions $(k(s), \xi(s), \tau(s), q(s))$, and then the arm configuration has to be calculated integrating the equations (1)-(4).

C. Mathematical Remarks on Jacobian Matrix

In order to use methods based on the Jacobian matrix:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{p}_{ee}}{\partial \boldsymbol{c}_t} \in \Re^{3\times 3} \qquad (22)$$

it is mandatory to have the analytical expression of the direct kinetics model: $\boldsymbol{p}_{ee} = \boldsymbol{F}(\boldsymbol{c}_t)$, as it occurs in two dimensional case [11]. In this manner each element of the Jacobian matrix (22) can be calculated, performing the partial derivatives of the tip position with respect to the cable tensions.

Supposing to have the deformation functions $(k(s), \xi(s), \tau(s), q(s))$, it is possible to find an analytical solution of the system of equations (1)-(3) [25], which is a system of *linear* ODEs with non constant coefficients. The resolution of that system permits to obtain the tip position through the integration of the equation (4):

$$\begin{aligned}
x_{ee} &= \int_0^L (1 + q(s)) t_x(s) ds && (23)\\
y_{ee} &= \int_0^L (1 + q(s)) t_y(s) ds && (24)\\
z_{ee} &= \int_0^L (1 + q(s)) t_z(s) ds && (25)
\end{aligned}$$

However, it is needed to solve analytically the system of *non-linear* ODEs (15)-(20) to find the deformation functions. To the best of our knowledge [25], there is no simple method that can be applied to this case for finding exact solutions. Therefore, the Jacobian methods seem to be not suitable in the case of non-constant curvature soft manipulators driven by three cables.

Considering the equations of the mechanical model (15)-(20) with more attention, the non linear terms can be neglected if the torsion $\tau(s)$ is approximately equal to zero for every section s. The torsion is not caused by the conical shape of the manipulator, but it is generated by the arrangement of the cables. The torsion of a cylindrical manipulator is not equal to zero, as follows from the equations (15)-(20) considering $I^{'}(s) = J^{'}(s) = 0$, but in many cases the torsion can be considered approximately equal to zero. However, for a non-constant curvature manipulator the effect of the torsion is not negligible. As a matter of fact, a simple example is performed with two manipulators: the first one has a conical shape an the second one has a cylindrical shape. We suppose they have the same geometrical and mechanical parameters, except for the diameters of the base and of the tip. In particular, the base of the conical shape manipulator has the same diameter of the cylindrical one, whereas the tip diameter is smaller of the 43% than the base diameter, as in the manipulator considered in this work (TABLE I). Using a torsion index defined as follows:

$$I_\tau = \frac{\sqrt{\int_0^L \tau^2(s) ds}}{L} \qquad (26)$$

and pulling two cables $T_1 = 1N$ and $T_2 = 3N$, it is possible to have a measure of the torsion which affects the two manipulators: the cylindrical one has $I_\tau = 0.051m^{-2}$ and the conical one has $I_\tau = 30m^{-2}$. This example demonstrated empirically how the torsion that effects the conical shape manipulator can not be neglected, however for the cylindrical one a simplification can be done in many case, as in [12].

Since analytical solutions are impossible to calculate, other solutions have been evaluated to solve the IK problem of this soft manipulator. In particular, a feed-forward neural

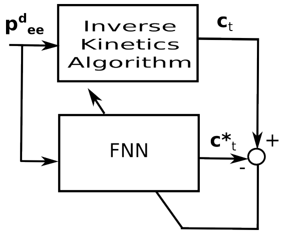

Fig. 4. The neural network learning the inverse kinetics model.

network has been used because it exhibits good performance in function approximation, as the theorem in [26] states.

TABLE I

MECHANICAL AND GEOMETRICAL PARAMETERS OF THE MODEL OF THE CONICAL SHAPE MANIPULATOR

$E[kPa]$	110	ν	0.5	$G[kPa]$	36.67
$yc[mm]$	2.5	$Rmax[mm]$	15	$Rmin[mm]$	6.5
$L[mm]$	310				

III. FEED-FORWARD NEURAL NETWORK

The neural network implemented is a fully-connected FNN with one hidden layer and with the hyperbolic tangent as the activation function ($a(x) = tanh(x)$). The FNN takes the tip position $\boldsymbol{p}^d_{ee}$ as input, where d indicates a *desired* position, and provides the cable tensions $\boldsymbol{c}^*_t$ (Fig. 4) as output. The data set has been generated by means of the direct kinetics model described above, using the parameters in TABLE I. In particular, the direct model is fed by cable tension samples $\boldsymbol{c}_t$, which are generated by a uniform distribution from $a = 1N$ to $b = 5.5N$. The size of data set is 500 samples. The data set has been divided first of all in training set (80%) and test set (TS)(20%). The training set is also split in estimation set (ES)(80%) and validation set (VS)(20%) [27]. The training set is used during learning phase, whereas the test set is only employed to evaluate the performance of the FNN. Moreover during the learning phase, the validation set is used to avoid the overfitting [27].

The input data set has been pre-processed, firstly, removing the mean value, secondly, realizing the decorrelation by using the PCA (principal component analysis), finally, performing the covariance equalization [27]. The output data set has been, also, normalized, dividing by their maximum values. The learning of the FNN is realized using the scheme in Fig. 4, where the synaptic weights are determined so that the output $\boldsymbol{c}^*_t$ of FNN comes close to the output of the inverse model $\boldsymbol{c}_t$. In order to minimize the mean square error (MSE) calculated on the ES, the synaptic weights are adjusted by using the back-propagation algorithm, which implements the gradient descent method including the momentum term. However, for a good generalization of the FNN and to avoid overfitting, the *early-stopping method* for training is implemented [27]. The method requires that after a period of training (N_t epochs) using the samples of the ES, the synaptic weights and bias levels of the FNN are fixed, and the network is operated in forward mode using the samples of the VS. When the validation phase is finished, the mean square error on the VS ($\mathcal{E}_V$) is calculated, and the training is resumed for another N_t epochs. The process is repeated until $\mathcal{E}_V$ reaches its minimum value. In particular, we compare the $\mathcal{E}_V(p)$ relative to the p^{th} iteration with the previous one $\mathcal{E}_V(p-1)$. When $\mathcal{E}_V(p) > \mathcal{E}_V(p-1)$ we increment a variable $counter$, whereas if $\mathcal{E}_V(p) < \mathcal{E}_V(p-1)$ the counter is reinitialized to zero. The algorithm is stopped when $counter$ is greater than $MaxCounter$. The initial values of the synaptic weights are selected from a uniform distribution with a mean value of zero and a variance equal to the reciprocal number of synaptic connections of the neurons [27].

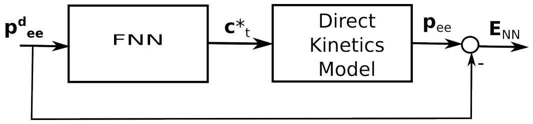

Fig. 5. Scheme employed during test phase of the FNN. After the FNN has been trained, its performance are measured on test set using the output of the FNN as input of the direct kinetics model.

A. Model Selection

The *universal approximation theorem* [26] for a nonlinear input-output mapping states that a FNN is able to approximate any continuous function defined on compact subset of $\Re^n$, by using a single hidden layer, which contains a finite number of neurons, and choosing non-constant, bounded, and monotone-increasing continuous function, as $a(x) = tanh(x)$. Therefore, the model selection consists in finding the right number of neurons N^*_H in the hidden layer. The procedure used in our work is based on the *cross-validation* [27]. The *model selection algorithm* has the following instructions:

```
• Initialize $N_H = 1$
• while ($N_H < MaxN_H$)
    – Random initialization of the FNN weights
    – Train the FNN for $N_{epochs}$ by using the ES
    – Calculate the mean square error on the VS: $\mathcal{E}_V$
    – Fill vector $v_{mse}[N_H] = \mathcal{E}_V$
    – $N_H = N_H + 1$
• end
• $N^*_H = argmin(v_{mse})$
```

In particular, we repeat the model selection algorithm N_{ms} times, because the algorithm is based on the random initialization of the synaptic weights of the FNN.

IV. RESULTS

A. Model selection of the FNN

The model selection algorithm described above is employed to set the number of hidden neurons for the FNN of the conical shape manipulator. In order to execute the algorithm a choice of three parameters has to be made: the maximum number of neurons in the hidden layer $MaxN_H$, the number of epochs N_{epochs} for training, and the number of times N_{ms} that we repeat the algorithm:

- $MaxN_H = 43$
- $N_{epochs} = 15000$
- $N_{ms} = 40$

In TABLE II the results of the model selection algorithm are shown. The first row presents the number of hidden neurons N_H, and the second row shows the number of times N_T that the model selection algorithm chooses N_H as a number of hidden neurons. The best number of hidden neurons is in the range [33-35], therefore we choose $N_H^* = 34$ as the optimum value for the FNN learning the IK model.

TABLE II
MODEL SELECTION OF THE FNN FOR THE CONICAL SHAPE MANIPULATOR DRIVEN BY THREE CABLES

N_H	30-32	33-35	36-38	39-41	42-23
N_T	6	11	8	8	7

B. Test of the FNN

The FNN has been trained as described in the previous section and the trends of the mean square error on the ES are shown in Fig. 6, where the MSE curves of four different trainings are plotted. The parameters employed in the back-propagation algorithm, which implements the early-stopping method, are the maximum number of epochs $MaxN_{Epoc}$, the number of epochs (N_t) in which the ES samples are used, the learning rate η, the momentum term α, and the maximum number of time ($MaxCounter$) that $\mathcal{E}_V(n)$ is greater than $\mathcal{E}_V(n-1)$:

- $MaxN_{Epoc} = 25000$
- $N_t = 20$
- $\eta = 0.01$
- $\alpha = 0.015$
- $MaxCounter = 250$

In Fig. 6 we observe that in every case the training is stopped before reaching the $MaxN_{Epoc}$, because the BP algorithm has found the best generalization of the FNN on the training data, thanks to the early-stopping method. In general the performances of the FNNs are as good as the performance of the FNNs relative to the curves MSE-1, MSE-2 and MSE-3. In these cases the FNNs reaches the same results in terms of generalization as shown for the third neural network (FNN-3) represented by the curve MSE-3 (Fig. 6). This curve reaches its minimum value of 0.47mN after ~18120 epochs, that is after ~70s. However, in some case the the early stopping method falls into local minimum, as in the case represented by curve MSE-4.

A clear representation of the FNN-3 performance is obtained by using the scheme in Fig. 5 on the test set. The error between the tip position reached by the direct kinetics model $\boldsymbol{p}_{ee}$ and the desired tip position $\boldsymbol{p}_{ee}^d$ has been calculated:

$$E_{NN} = \parallel \boldsymbol{p}_{ee} - \boldsymbol{p}_{ee}^d \parallel \tag{27}$$

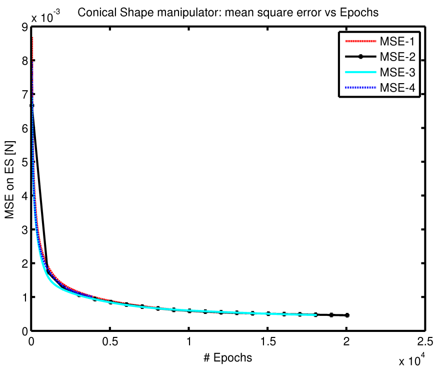

Fig. 6. Conical shape manipulator driven by three cables: trends of the mean square errors (MSE) calculated on the ES during four different training. The trained FNNs represented by the curve MSE-1, MSE-2, MSE-3 have the same performance. The FNN relative the curve MSE-4 falls to a local minimum during the learning, therefore it has worse perfomances than the other cases.

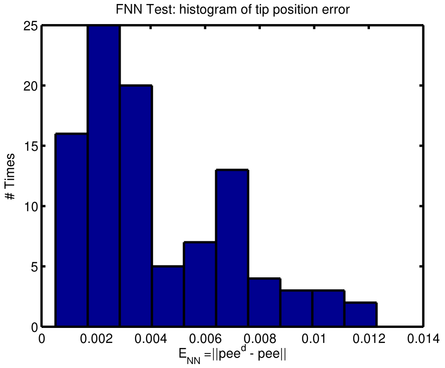

Fig. 7. Conical shape manipulator driven by three cables: histogram of the FNN-3 tip errors (in m) calculated using the test set.

In Fig. 7 the distribution of the tip errors (27) relative to TS samples is shown. In order to appreciate the error distribution in Fig. 7, some statistics values have been calculated for the FNN-3, as shown in TABLE III. The FNN-3 shows good performance in the 89.8% of the cases, when the tip error is less than 8mm. Moreover, the average tip error is 4.2mm, and the maximum error reached in two cases is 12.3mm. The relative average error calculated with respect the total length of the manipulator is 1.36%, whereas the relative maximum error is 3.96%.

The computational cost of the FNN is $7N_H^* + 3 = \mathcal{O}(N_H^*)$, therefore it takes 0.162ms by using a Intel® Core™ i7-2670QM CPU at 2.20GHz.

V. CONCLUSION

The inverse kinetics problem of non-constant curvature soft manipulators driven by three cables cannot be solved by using the classical mathematical methods based on the calculation of the Jacobian matrix. The differential equations of the static model of the manipulator are indeed non-linear, hence the exact solutions cannot be calculated easily. Since the analytical solutions of the mechanical model are not available, the elements of the Jacobian matrix cannot be

TABLE III

PERFORMANCE OF THE FNN-3 ABLE TO LEARN THE IK OF A NON-CONSTANT CURVATURE SOFT MANIPULATOR DRIVEN BY THREE CABLES

	Absolute (mm)	Percentage (%)
mean	4.2	1.36
std	2.8	0.91
max	12.3	3.96
$p_{\%}$		89.8

calculated. For this reason, a FNN has been proposed for its effectiveness at approximating functions

The feasibility of the method based on the FNN learning has been tested on data generated by the mechanical model of the manipulator. The FNN shows remarkable performance, because in the 89.8% of the cases the tip reaches the desired position with an error of less than 8mm. The relative average error is 1.36% with respect to the total length of the arm.

In the future, we will validate the method also with a real mechanical prototype of the arm. To investigate the effectiveness of the approach, it will be used to generate paths in the task space in order to perform grasping.

REFERENCES

[1] D. Trivedi, C. R. D, W. M. Kier, and I. Walker, “Soft robotics: Biological inspiration, state of the art, and future research,” *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

[2] S. Kim, C. Laschi, and B. Trimmer, “Soft robotics: a bioinspired evolution in robotics,” *Trends in biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.

[3] G. Robinson and J. Davies, “Continuum robots - a state of art,” in *Proc. IEEE Int. Conf. on Robot. and Automat.*, Detroit, USA, May 1999, pp. 2849–54.

[4] M. Calisti, M. Giorelli, G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario, “An octopus-bioinspired solution to movement and manipulation for soft robots,” *Bioinspir. Biomim.*, vol. 6, (2011) 036002.

[5] B. A. Jones and I. D. Walker, “Kinematics for multisection continuum robots,” *IEEE Trans. on Robot.*, vol. 22, no. 1, pp. 43–57, 2006.

[6] N. Cheng, M. Lobovsky, S. Keating, A. Setapen, K. Gero, A. Hosoi, and K. Iagnemma, “Design and analysis of a robust, low-cost, highly articulated manipulator enabled by jamming of granular media,” in *Proc. IEEE Int. Conf. on Robot. and Automat.*, St Paul, Minnesota, USA, May 2012, pp. 4328–4333.

[7] M. Cianchetti, A. Arienti, M. Follador, B. Mazzolai, P. Dario, and C. Laschi, “Design concept and validation of a robotic arm inspired by the octopus,” *Material Science and Engineering: C*, vol. 31, no. 6, pp. 1230–1239, 2011.

[8] R. J. Webster and B. A. Jones, “Design and kinematic modeling of constant curvature continuum robots: A review,” *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[9] G. S. Chirikjian and J. W. Burdick, “A modal approach to hyper-redundant manipulator kinematics,” *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, pp. 343–354, 1994.

[10] N. Simaan, R. Taylor, and P. Flint, “A dexterous system for laryngeal surgery,” in *Proc. IEEE Int. Conf. on Robot. and Automat.*, New Orleans, USA, 2004, pp. 351–357.

[11] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, “A two dimensional inverse kinetics model of a cable driven manipulator inspired by the octopus arm,” in *Proc. IEEE Int. Conf. on Robot. and Automat.*, St Paul, Minnesota, USA, May 2012, pp. 3819–3824.

[12] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, “Mechanics modelling of tendon driven continuum manipulators,” *IEEE Trans. on Robot.*, vol. 24, no. 6, pp. 1262–1273, 2008.

[13] Y. Kuroe, Y. Nakai, and T. Mori, “A new neural network learning of inverse kinmatics of robot manipulator,” in *in Proc. IEEE Int. Conf. on Neural Network*, Orlando, USA, 1994, pp. 2819–2824.

[14] A. Hassan, N. Usmail, A. Hamouda, I. Aris, N. Marhaban, and H. Al-Assadi, “Artificial neural network-based kinematics jacobian solution for serial manipulator passing through singular configurations,” *Advanced in Engineering Software*, vol. 41, pp. 359–367, 2010.

[15] J. Martin, J. de Lope, and M. Santos, “A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers,” *Neurocomputing*, vol. 72, pp. 2806–2814, 2009.

[16] E. Oyama, A. Agah, K. MacDorman, T. Maeda, and S. Tachi, “A modular neural network architecture for inverse kinematics model learning,” *Neurocomputing*, vol. 38-40, pp. 787–805, 2001.

[17] E. Dermatas, A. Nearchou, and N. Aspragathos, “Error-back-propagation solution to the inverse kinematic problem of redundant manipulator,” *Robotics and Computer-Integrated Manufacturing*, vol. 12, no. 4, pp. 303–310, 1996.

[18] Y. Zhang, Z. Tan, K. Chen, Z. Yang, and X. Lv, “Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulators straight-line example,” *Robot. and Auton. Syst.*, vol. 57, pp. 645–651, 2009.

[19] D. Gorinevsky and T. Connolly, “Comparison of some neural network and scattered data approximations: the inverse manipulator kinematics example,” *Neural Computation*, vol. 6, pp. 521–542, 1994.

[20] D. Braganza, D. Dawson, I. Walker, and N. Nath, “A neural network controller for continuum robots,” *IEEE Trans. on Robot.*, vol. 23, no. 5, pp. 1270–1277, 2007.

[21] M. Giorelli, F. Renda, G. Ferri, and C. Laschi, “A feed-forward neural network for solving the inverse kinetics of non-constant curvature soft manipulators driven by cables,” in *6th ASME Dynamical Systems and Control Conference*, Stanford University, Palo Alto, CA, USA, 2013, (accepted).

[22] S. S. Antman, *Nonlinear problems of elasticity*. New York, NY: Springer, 2005.

[23] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, “A 3d steady state model of a tendon-driven continuum soft manipulator inspired by octopus arm,” *Bioinspir. Biomim.*, vol. 7, no. 2, (2012) 025006.

[24] F. Renda and C. Laschi, “A general mechanical model for tendon-driven continuum manipulators,” in *Proc. IEEE Int. Conf. on Robot. and Automat.*, St Paul, Minnesota, USA, May 2012, pp. 3813–3818.

[25] D. Zwillinger, *Handbook of differential equations*. London: Academic Press, INC., 1992.

[26] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematichs of Control, Signals, and Systems*, vol. 2, pp. 303–314, 1989.

[27] S. Haykin, *Neural network and learning machine, 3rd edition*. New Delhi: PHI learning, 2010.