CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

NEURAL NETWORKS APPLIED TO THE PREDICTION OF FLEXIBLE PIPES ARMOR WIRES CONFIGURATION UNDER PURE BENDING

Gabriel Mattos Gonzalez

João Paulo Ramos Cortina

José Renato Mendes de Sousa

Luis Volnei Sudati Sagrilo

gmattos@ laceo.coppe.ufrj.br

jpr.cortina@laceo.coppe.ufrj.br

jrenato@ laceo.coppe.ufrj.br

sagrilo@coc.ufrj.br

LACEO – Programa de Engenharia – COPPE/UFRJ - BRAZIL

Avenida Pedro Calmon s/nº, Cidade Universitária, 21941-972, Rio de Janeiro, RJ, Brazil

Abstract. *Flexible pipes are a feasible alternative for offshore oil and gas exploitation. High flexibility combined with high axial resistance compound their main characteristics. Local bending mechanics of these pipes usually treats their tensile armors as curves on a torus. Geodesic curves are one of the most utilized curves. It is not possible, however, to solve geodesic differential equations analytically, which imposes difficulties on the mechanical analysis. This work intends to train Artificial Neural Networks with data obtained from the numerical solution of these equations thus proposing an alternative solution for the stiff nonlinear differential equations for the special case of unbounded geodesic of a torus. The results showed good correlations for the whole set of cases presented.*

Keywords: *Flexible pipes, Geodesic, Artificial Neural Network*

1 INTRODUCTION

Flexible pipes are a consolidate concept for offshore oil exploitation in deep and ultra-deep water depths. These pipes are multilayered composite structures composed of concentric metallic and polymeric layers. Whereas metallic layers are mainly responsible for granting structural resistance; polymeric layers are either used to seal the pipe bore and its annulus or mitigating wear and friction between layers. A fundamental characteristic of these pipes is to combine high axial strength and stiffness with low bending stiffness and, therefore, high compliance to offshore motions. Figure 1 illustrates a typical flexible pipe cross-section and its main layers and components.

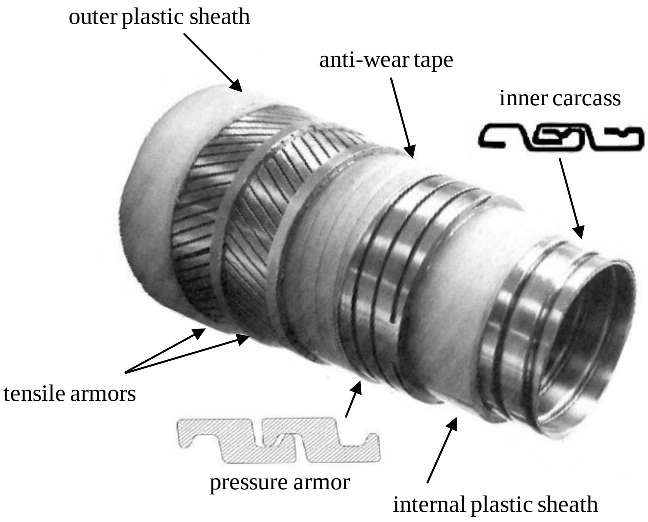

Figure 1. Typical flexible pipe (de Sousa et al., 2012).

Structural modelling of these pipes has been widely studied, especially in what concerns its axial behavior. In pure bending, however, some controversies arise, especially regarding the tendon slip path, the role played by friction and the pipe's hysteretic behavior. Even though many works have been published trying to solve this issue, a topic that still divides opinions around the world deals with the path undertaken by the tensile armor wires when the pipe is subjected to pure bending. While some authors assume that this path follows the geodesic of a torus (Féret and Bourzanel, 1986), others propose the loxodromic curve (Saevik, 1992).

A physical interpretation of these paths is given by Saevik (1992). The loxodromic represents the wire in a bent cylinder when no sliding occurs, whilst the wire should follow the geodesic path if no friction is considered and the wires are free to slide axially and transversally. Féret et al. (1995) proposed a function dependent on an empirical term that is capable of considering any curve in between these two limit paths, although little or no discussion about this function is fulfilled. Later, Leroy and Estrier (2001) followed the same approach, but disregarding this empirical parameter and attributing the function to be dependent on the friction and, therefore, on the variation of lateral displacement. A similar procedure is also adopted by Østergaard (2012), where six differential equations based on angular coordinates is proposed for describing the mechanics of a frictionlees wire laying on a torus surface. This approach is supposed to be very effective and showed that, even when no friction is considered, the equilibrium path followed by the tensile wire should be in between those two paths, the loxodromic and the geodesic.

Although this approach is a state-of-the-art procedure, it also contains some uncertainties and difficulties, such as the implementation process and the stability of the solution, which leads relevant authors to opt for still dealing with only one of these paths, such Saevik (2011). In this later work, sliding is assumed to occur only on axial direction while the wire is supposed to be adhered to its supporting surface. Saevik states that empirical evidence corroborates its assumption, at least for outer armor layers (2011). However, if the outer sheath is damaged, friction between the tensile armors can be significantly reduced and, therefore, the wire's path may be affected.

What we intended to show with this brief introduction is that both paths are significantly relevant for understanding the mechanics of flexible pipes under pure bending. Furthermore, it is important to highlight that while loxodromic curve equations are easily derived and solved from basic differential geometry concepts, geodesics are not that simple. This happens mainly because geodesic equations lead to stiff nonlinear differential equations of second order, which are assumed only to be solved by numerical methods (Out and Morgen, 1997). To allow future engineering analysis to also deal with these curve in an easier way, this work intend to present a closed form expression obtained by numerical regression for geodesic curves. In order to do so, inputs for the Artificial Neural Networks (ANNs) are obtained directly from the numerical solution of the geodesic boundary value problem. This work is divided in a brief description of the geometrical relations involved in the derivation of the geodesic differential equations followed by a discussion on the limit curves, an ANN review and the case study.

2 GEOMETRICAL RELATIONS

In structural analysis, spatial rods are geometrically represented as surface curves. The torus is a useful surface for this purpose, once it stands for a generic bent cylinder under constant curvature. Figure 1 illustrates the torus and the angles necessary for its parametrization, while Eq.1 provides its parametric equation.

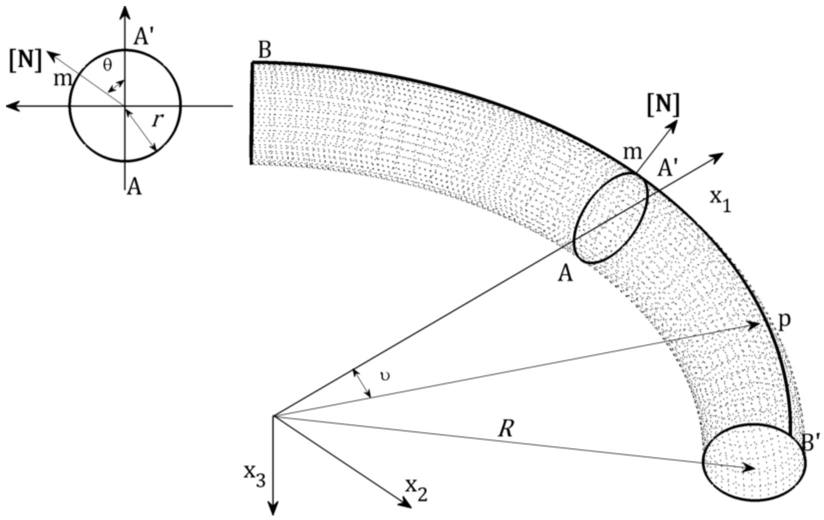

Fig. 2 – Geometry and parameters of a torus.

$$\mathbf{x} = \begin{bmatrix} (R + r\cos\theta)\cos(\upsilon) \\ (R + r\cos\theta)\,\mathrm{sen}(\upsilon) \\ r\sin\theta \end{bmatrix} \tag{1}$$

In order to orient the curve laying on this surface, a local orthogonal triad has to be established. A common description is given by the Darboux-Ribaucourt axes ($\mathbf{t}$,**N**,**B**), where [$\mathbf{t}$] is the curve unit tangent, [**N**] is the surface normal and [**B**] the surface binormal vectors. The variation of this triad with respect to the curve arc-length (s) gives the following relation:

$$\frac{d}{ds}\begin{bmatrix} \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} 0 & \kappa_n & \kappa_g \\ -\kappa_n & 0 & -\tau_g \\ -\kappa_g & \tau_g & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix} \tag{2}$$

where κ_n and κ_g are, respectively, the normal and geodesic curvatures and τ_g is the geodesic torsion of the curve. These components are essential for evaluating the mechanics of a pre-curved and twisted rod, very often related to wire ropes or flexible pipe armor tendons. The calculation of these curvatures components may be troublesome for a generic curve, thus, for the special case of flexible pipes' tendons, limit curves that may give simplified responses are chosen. These curves are named geodesic and loxodromic.

3 LIMIT CURVES

For bending mechanical analysis, the curvature components need to be calculated. From here, some assumptions and simplifications have to be admitted. As stated, the two limit curves, the geodesic or the loxodromic, are the most common paths utilized as simplifications. Figure 3 illustrates these limit curves.

Loxodromic curves are the curves that cross all the horizontal meridians of a given surface with the same constant angle. Helixes are not only loxodromes of the cylinder, but also geodesics. As the scope of this work is the geodesic curves, the loxodromic curves are no longer addressed.

Geodesic curves have their normal coincident with the surface normal. A more precise and mathematic accurate description of a geodesic is given by do Carmo (1976), for whom geodesic are constant speed curves with vanishing geodesic curvature. They are also the shortest curves that connect two points in a surface, in other words, they area curves that play analogous role to a straight line in a plane (Nutbourne and Martin, 1988). Many different parametrizations lead to many different differential equations of the geodesic of a surface. The first and more widespread parametrization was given by Leonhard Euler and Joseph Louis Lagrange (Gray et al., 1993), which consists of obtaining the minimization of the arc-length functional, Π, of a given surface, $\boldsymbol{x}$. This minimization problem recalls the classical Euler-Lagrange equation, as shows Eqs .(3) and (4):

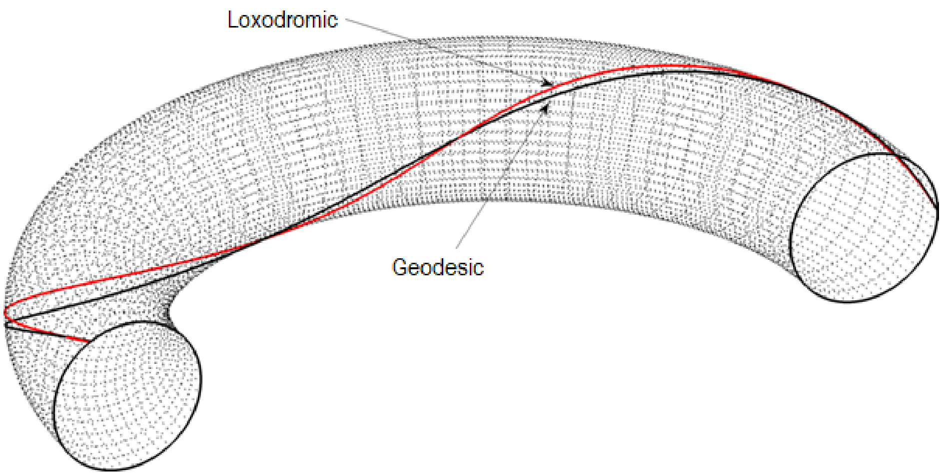

Fig. 3 – Limit curves.

$$s = \int \Pi dv = \int \sqrt{E + G(\theta_{,v})^{2}} dv \tag{3}$$

$$\frac{\partial \Pi}{\partial \theta} - \frac{d}{dv}\left(\frac{\partial \Pi}{\partial \theta_{,v}}\right) = 0^{1} \tag{4}$$

This problem was studied by Saevik (1991) and Out and Morgen (1997), among others. It is pointed out that the differential equations cannot be solved analytically (Out and Morgen, 1997). A less straightforward, but more systematic way, to obtain the differential equations of a geodesic of a generic surface is given by means of the set of differential equations illustrated by Eq (5) (Nutbourne and Martin, 1988).

$$\begin{aligned} &\frac{d^2 v}{ds^2} + \Gamma_{11}^{1}\left(\frac{dv}{ds}\right)^2 + 2\Gamma_{12}^{1}\left(\frac{dv}{ds}\right)\left(\frac{d\theta}{ds}\right) + \Gamma_{22}^{1}\left(\frac{d\theta}{ds}\right)^2 = 0 \\ &\frac{\partial^2 \theta}{\partial s^2} + \Gamma_{11}^{2}\left(\frac{dv}{ds}\right)^2 + 2\Gamma_{12}^{2}\left(\frac{dv}{ds}\right)\left(\frac{d\theta}{ds}\right) + \Gamma_{22}^{2}\left(\frac{d\theta}{ds}\right)^2 = 0 \end{aligned} \tag{5}$$

In surfaces parametrized by orthogonal coordinate system, such as the torus, the first fundamental form F is zero, thus the Christoffel symbols Γ_{jk}^{i} $(i, j, k = 1{,}2)$, which relates the first fundamental form of a surface and its derivatives, are given by Eq. (6).

$$\begin{aligned} &\Gamma_{11}^{1} = \frac{E_{,v}}{2E};\ \Gamma_{12}^{1} = \frac{E_{,\theta}}{2E};\ \Gamma_{22}^{1} = \frac{-G_{,v}}{2E}; \\ &\Gamma_{11}^{2} = \frac{-E_{,\theta}}{2G};\ \Gamma_{12}^{1} = \frac{G_{,v}}{2G};\ \Gamma_{22}^{1} = \frac{-G_{,v}}{G} \end{aligned} \tag{6}$$

where E,F and G are the first fundamental form given by classical modern differential geometry (Nutbourne and Martin, 1988), as follows:

$$E = \mathbf{x}_{,v}\mathbf{x}_{,v} = (R + r.\cos\theta)^2;\ F = \mathbf{x}_{,v}\mathbf{x}_{,\theta} = 0;\ G = \mathbf{x}_{,\theta}\mathbf{x}_{,\theta} = r^2 \tag{7}$$

Substituting Eqs. (6) and (7) into Eqs. (5), the set of differential equations for the geodesic of a torus can be written as:

---

[1] The subscript $(*)_{,i}$ in this work indicates partial derivative of the function generic function, $(*)$, with respect to the variable i.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

$$\frac{d^2 v}{ds^2} - \frac{2r\sin\theta}{R + r\cos\theta}\left(\frac{dv}{ds}\right)\left(\frac{d\theta}{ds}\right) = 0$$
$$\frac{d^2\theta}{ds^2} + \frac{1}{r}\sin\theta\,(R + r\cos\theta)\left(\frac{dv}{ds}\right)^2 = 0 \tag{8}$$

Do Carmo (1976), in his book, illustrates, using Liouville formula for binormal curvature, Eq. (9), how to obtain an alternative differential equation for the geodesic of any given surface.

$$\kappa_g = -\frac{E_{,\theta}}{2E\sqrt{G}}\cos\phi + \frac{G_{,v}}{2E\sqrt{G}}\sin\phi + \frac{d\phi}{ds} \tag{9}$$

As previously stated, geodesic curves are curves with null geodesic, or binormal, curvature, κ_g. Thus, by equating Eq. (9) to zero, a first order differential equation written in terms of lay angle $\phi(s)$ can be written. In order to completely predict the geodesic equations, a relation with any other torus angular parameter is necessary. This relation may be obtained from the tangent definition. As can be seen in Eq. (10), the tangent of a unit speed curve $\boldsymbol{\alpha}$ provides the following relations:

$$\mathbf{t} = \frac{d\boldsymbol{\alpha}}{ds} = \boldsymbol{t}_v\cos\phi + \boldsymbol{t}_\theta\sin\phi$$
$$\boldsymbol{t}_v = \frac{\mathbf{x}_v}{\|\mathbf{x}_v\|};\ \ \boldsymbol{t}_\theta = \frac{\mathbf{x}_\theta}{\|\mathbf{x}_\theta\|} \tag{10}$$

Through this equation one may obtain Eq. (11).

$$\frac{dv}{ds} = \frac{\cos\phi}{\|\mathbf{x}_v\|};\ \ \frac{d\theta}{ds} = \frac{\sin\phi}{\|\mathbf{x}_\theta\|} \tag{11}$$

Thus, a distinct set of differential equations of a geodesic may be obtained in terms of two functions of the arc-length s; the lay angle, ϕ; and radial angular parameter, θ, as shown in Eq. (12).

$$\frac{d\theta}{ds} = \frac{\sin\phi}{r} \tag{12}$$

$$\frac{d\phi}{ds} = -\frac{\kappa \sin\theta}{1 - r\kappa\cos\theta}\cos\phi$$

In order to obtain geodesic parameters, either one of the two set of differential equations, Eq. (5) and (12), need to be solved by finite differences, which was the case of this work through MATLAB® built-in *bvp4c* that applies the Lobatto-IIIa method (MATHWORKS, 2014)., or by imposing analytical constraints to the curve differential equations. From the last approach, the curvature components can, after troublesome analytical manipulations, be obtained without the need of solving the differential equation. It is worth mentioning that, with any of these solutions in hand, the curvature components of geodesic curves can be obtained.

4 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) constitute a form of mathematical processing of information with architecture and operation similar to the biological neural networks. The learning ability is the remarkable feature of the ANNs, and, in practice, this ability results in applications such as pattern recognition, classification, clustering, optimization and function approximation, among others (Haykin, 2005).

In this work, ANNs are employed as function approximations. In other terms, an ANN defines a relationship between the bent flexible pipe tensile wire geometrical parameters (network inputs) and the geodesic curve parameters: the radial angle θ and the laying angle φ given as functions of the wire arc-length s (network outputs). As an ANN needs sample data of the inputs and outputs to define the function relation between them (ANN training), here, this data sample will be provided by the numerical integration of the geodesic curve differential equation for some cases.

The main goal is training an ANN and checking whether its extrapolated results are or not valid. This is one of the most critical issues when it comes to ANN applications. In order to do so, the radial and the laying angles along the wire arc-length for a given configuration of pipe, which is not present in the training example, are compared to the numerical solution.

The most common ANN architecture used as a function approximation uses three layers: the input layer, the hidden layer and the output layer. This configuration is outlined in Figure 4. The first layer reads the input values of the network, whilst the output layer delivers the responses of the network. Both, hidden and output layers are composed by neurons (numerical functions). Each connection between elements in neighboring layers contains a weight (synapse weighting). Linked to the hidden and output layers, there are also bias parameters with unitary values.

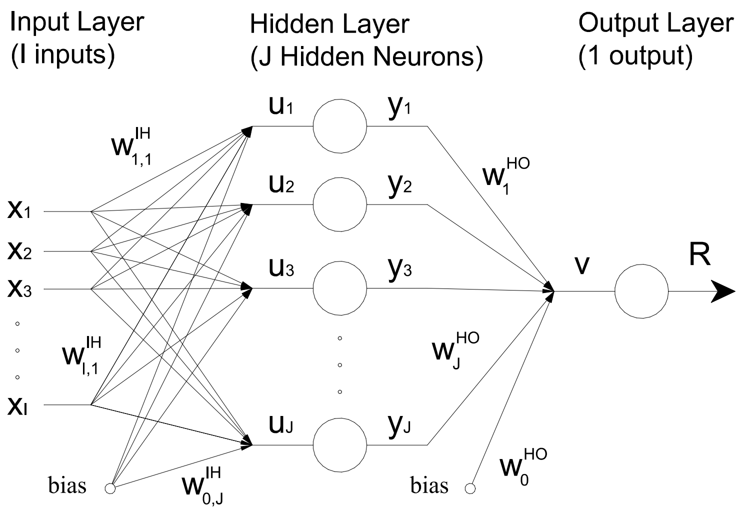

Figure 4. ANN architecture.

Generally, an ANN will work as follows:

a neuron j in the hidden layer receives an input u_j, given by:

$$u_j = w_{0,j}^{IH} \cdot 1 + \sum_{i=1}^{I} \left(w_{i,j}^{IH} \cdot x_i \right) \tag{13}$$

where I is the number of network inputs, $w_{0,j}^{IH}$ is the bias weight of the input layer, $w_{i,j}^{IH}$ is the weight between the input i and the respective neuron j in the hidden layer ($j = 1, \dots, J$, J is the total number of hidden layer neurons) and x_i is the i network input. Each neuron returns an output y_i denoted by:

$$y_j = \varphi_H(u_j) = \varphi_H\left(w_{0,j}^{IH} \cdot 1 + \sum_{i=1}^{I} \left(w_{i,j}^{IH} \cdot x_i \right) \right) \tag{14}$$

where $\varphi_H(.)$ is the so-called *activation function* or *transfer function*. The activation functions usually are sigmoid functions, like hyperbolic tangent or the logistic functions. Figure 5 presents the form of the hyperbolic-tangent and the logistic functions. The logistic function is defined by the following expression:

$$g_{Log}(u) \equiv \frac{1}{1+e^{-u}} \tag{15}$$

Its image is contained in an interval from zero to one. The hyperbolic tangent function is denoted by:

$$tanh(u) \equiv \frac{e^{u}-e^{-u}}{e^{u}+e^{-u}} \tag{16}$$

The range of output values from the hyperbolic tangent function belongs to the interval from -1 to +1.

Both functions are continuously increasing and differentiable over their domain as well as limited. These features are required to introduce a nonlinear mapping in the network (Smith, 1993). Due to the nature of the limit values, the logistic function is usually used in pattern recognition applications, while the hyperbolic tangent is employed to function approximation and time series prediction. The weights control the steepness and the offset function. As the in this work consists of a function fitting, the hyperbolic tangent is chosen for the hidden neurons of the networks of this study.

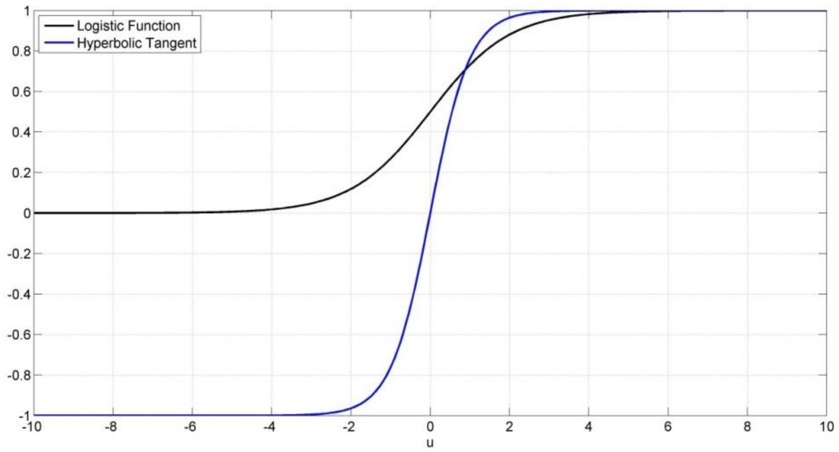

Figure 5. Activation functions.

For an ANN with a single output R, this parameter is determined by:

$$R = \varphi_O(v) = \varphi_O\left(w_0^{HO} \cdot 1 + \sum_{j=1}^{J}\left(w_j^{HO} \cdot y_j\right)\right) \tag{17}$$

where $\varphi_{O(.)}$ is the activation function of the neuron in the output layer, usually a linear function. The remaining parameters are defined similarly to those in the hidden layer (Eq. (13)).

For an ANN, as previously presented and considering the input (x) as a group of values parameters, the relation between these values and the network output R is:

$$R(\boldsymbol{W}, x) = f(\boldsymbol{W}, x) \tag{18}$$

where x is a vector containing the values parameters of the input and $\boldsymbol{W}$ is a matrix containing all ANN connecting weights.

The so-called *networking training* consists in the searching of a set of weights that propitiate a good fitting of the function that are being approximated. In this manner, a sample set of input and outputs is needed to adjust the network weights by an optimization method. The

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

optimization problem is expressed by the minimization of a function error, usually the mean square error E, defined by:

$$E(\boldsymbol{W}) \; = \frac{1}{N}\sum_{i=1}^{N}(R(\boldsymbol{W}, x_i) - t_i)^2 \tag{19}$$

where N is the length of the training sample, and t is the desired output sample.

An iterative optimization procedure is employed to obtain optimum values for the weights $\boldsymbol{W}$. In this work, the Levenberg-Marquardt algorithm is employed in the networking training. It is a variation of the Newton's method designed to minimize functions that are sums of squares of other nonlinear functions (HAGAN et al., ISBN 0-9717321-0-8), as the error function, Eq. (19). Through the variation of a parameter called learning rate, this algorithm provides a good equilibrium between the speed of the Newton's method and the convergence guaranteed by the steepest descent procedure. The Levenberg-Marquardt algorithm can be found in (HAGAN et al., ISBN 0-9717321-0-8). Each iteration in the optimization procedure is named *epoch*.

All the optimization methods (also named training algorithms) starts from an initial weights set (a point in the error surface, Eq. (19)). Usually, the weights are initialized using small random values (Smith, 1993). Hence, the outputs of the activation functions falls out of the saturation region (low magnitude derivatives region), turning the training procedure faster.

Since the input and output parameters can present different order of magnitude values, it must be scaled in some way. As in this work the parameters values will be in a defined range, the training data is scaled to fall in the $[-1; +1]$ range. This is done through the mathematical expression:

$$y_i \; = \frac{y_{max} \, - y_{min}}{x_{max} - x_{min}} \, (x_i - x_{min}) + \; y_{min} \tag{20}$$

where the minor subscripts max and min refer to the maximum and minimum values present in the training sample, respectively. For some applications (as time series), the normalization through the difference with the mean value and division for standard deviation of the data sample is employed. The data scale also aids to turn the training process quick. For a network trained with scaled data, the network application for new inputs will result in scaled output values, which must be reversed for an adequate use.

In general, two types of errors may occur during the networking training. The first is related to the stopping of training in the beginning of iterative procedure, far from a good set of weights, resulting in high approximation errors (even for trained data). The second is related to the over-fitting of the training sample: the network adapts too much to the particular training set. Both cases lead to poor estimation for new inputs (non-trained data) (Smith, 1993; HAGAN et al., ISBN 0-9717321-0-8). Therefore, there is an optimum level of training to be achieved. A commonly and simple procedure applied to avoid both errors is separate part of the training set for the training itself (weights optimization) and the other part for validation, with separated error function evaluations. The usual behavior of these errors is depicted in Figure 6. The training process must be stopped when the error of the validation set begins to increase, avoiding the over-fitting and ensuring a good network generalization.

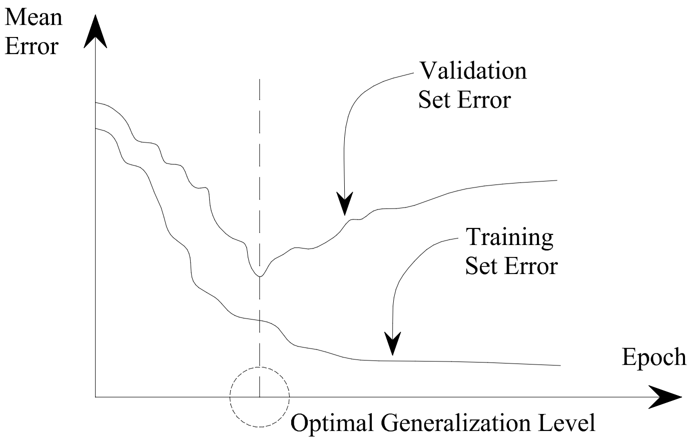

Figure 6. Usual behavior of the training and validation errors.

This work proposal consists in training ANNs capable of predicting the path of the geodesic curve followed by a tensile armor wire when the flexible pipe is under bending. In order to predict this path, geometrical parameters, such as the torus minor and major radius, initial laying angle, length of the wire and normalized arc-length, are taken as input parameters, whilst the angles defining geodesic differential equations, θ and φ, are the aimed outputs. This can be represented by the following expressions:

$$[\theta_i] = f([r, R, \varphi_0, S_{max}, s_i]) \quad (21)$$

$$[\varphi_i] = f([r, R, \varphi_0, S_{max}, s_i]) \quad (22)$$

where r is the internal cylinder radius, R is the bending radius (see Figure 2), φ_0 is the initial laying angle (before bending), S_{max} is the total arc length (corresponding to one helical pace), s_i is a point in the arc length (normalized, from 0 to 1) and θ_i and φ_i are the radial angle and the laying angle in the point s_i, respectively.

Thus, the radial angle θ and laying angle φ along the arc s can be easily determined for new combinations of geometrical wire parameters and bending radius.

5 CASE STUDY

Aiming to study the ability of the ANNs in defining the relations expressed by Eqs. (21) and (22), some cases are selected to compose the training dataset. These cases are constituted of variations in the wire geometrical parameters, imposed bending radius and the radial angle and laying angle responses, evaluated by numerical integration of geodesic differential equations. So, the training dataset can be represented as:

$$\begin{bmatrix} r_1, R_1,\ \varphi_{0_1},\ S_{max_1},\ s_{1,1}\ , \theta_{1,1}, \varphi_{1,1} \\ r_1, R_1,\ \varphi_{0_1},\ S_{max_1},\ s_{1,2}\ , \theta_{1,2}, \varphi_{1,2} \\ \vdots \\ r_1, R_1,\ \varphi_{0_1},\ S_{max_1},\ s_{1,I}\ , \theta_{1,I}, \varphi_{1,I} \\ r_2, R_2,\ \varphi_{0_2},\ S_{max_2},\ s_{2,1}\ , \theta_{2,1}, \varphi_{2,1} \\ \vdots \\ r_k, R_k,\ \varphi_{0_k},\ S_{max_k},\ s_{k,i}\ , \theta_{k,i}, \varphi_{k,i} \\ \vdots \\ r_K, R_K,\ \varphi_{0_K},\ S_{max_K},\ s_{K,I}\ , \theta_{K,I}, \varphi_{K,I} \end{bmatrix} \tag{23}$$

where K is the total number of selected cases, and I is the total number of points in the arc length employed for each case. The first five columns are the inputs of the network, while the sixth and seventh columns are the outputs, see Eqs. (21) and (22). In this manner, the inputs can be seen like $(I \cdot K)$ vectors. The outputs for training the network for the radial angle is a vector of $(I \cdot K)$ values of $\theta_{k,i}$, and the outputs for training the network for the laying angle is a vector of $(I \cdot K)$ values of $\varphi_{k,i}$.

The selection of cases must be done with caution. The limit values f the parameters must be present in the training set in order to cover the boundaries domain of the relations. Furthermore, cases with a combination of the values inside the domain are required to enable the network learning of the cross relationship between the parameters.

A total of 108 cases was selected with a discretization in the arc length s of 51 points (from 0 to 1, normalized), totalizing 5508 training data points. The range values of the inputs parameters are presented in Table 1.

Table 1. Range values of the input parameters in training

Parameter	Minimum Value	Maximum Value
r	0.04m	0.40m
R	1.0m	30.0m
φ_0	15.0º	55.0º
S_{max}	0.472m	5.870m
s	0	1

As presented previously, the network architecture consists in the number of inputs, outputs and nodes in the hidden layer. Since the number of inputs and outputs are defined by the problem, the quantity of hidden nodes is an open issue. The more hidden nodes are used, the more power to represent functions is given to the network. However, an excess of hidden nodes turns the network prone to the over-fitting and the weak generalization (Smith, 1993). One way to estimate the sufficient quantity of nodes is to perform a sensitivity analysis by iteratively varying this parameter and checking the fitting.

Once the networks weights are initialized randomly, the training procedure will stop in a different point of the error surface (a local minimum) every time the training is done. Hence, in the sensitivity analysis, the training was repeated 15 times for each number of nodes to compose a statistical view of the errors. The nodes in the hidden layer were varied from 5 to 70. In Figure 7, the result of the study is plotted, for both θ and φ. The mean value of the errors locates the point and the bar corresponds to the standard deviation. As can be seen, from 25 hidden nodes forward the errors present low values.

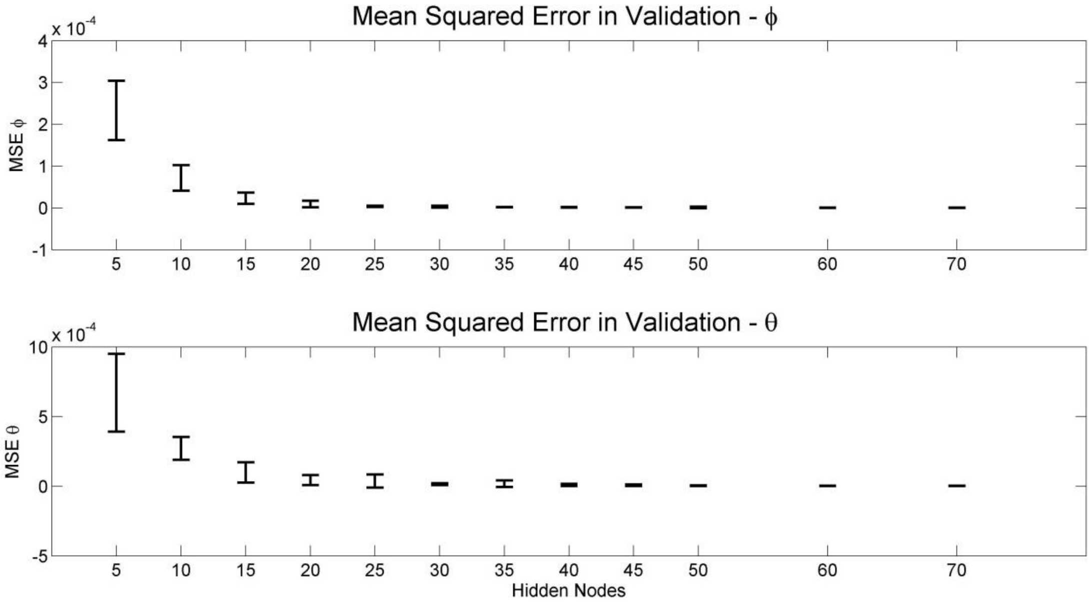

Figure 7. Hidden nodes sensitivity study results.

The results of Figure 7 are related to the validation part of the training sample. In this work, all ANNs are trained using 80% of the data for training itself and the remaining 20% is separated for validation. The selection is done randomly in the dataset, Eq. (11), and all the networks were trained using the Neural Network Toolbox®, from MATLAB® (MATHWORKS, 2014).

Summing up, two networks were trained, both with 25 hidden nodes, each one for the radial angle θ and for laying angle φ.

With the trained ANNs, new cases are selected to test the performance. These cases have parameters values in ranges between the values applied in the training of the networks, and these range values are presented on Table 2. Altogether, 49 cases for ANN testing were generated, which constitutes 2499 testing data points.

Table 2. Range values of the input parameters for testing ANNs

Parameter	Minimum Value	Maximum Value
r	0.07m	0.35m
R	8.0m	13.0m
φ_0	22.0º	42.0º
S_{max}	0.657m	5.870m
s	0	1

In Figure 8, the results of the ANN approximation is plotted for a case with the parameters: $r = 0.17m, R = 13m, \varphi_0 = 31°, S_{max} = 2.074m$. The ANN results agree with the ones provided by the numerical integration of differential equation of the angles. The behavior of the radial angle θ is simpler than the laying angle φ behavior, thus the ANN results for the radial angle are better than the results of the laying angle in this case, as also in the remaining testing cases.

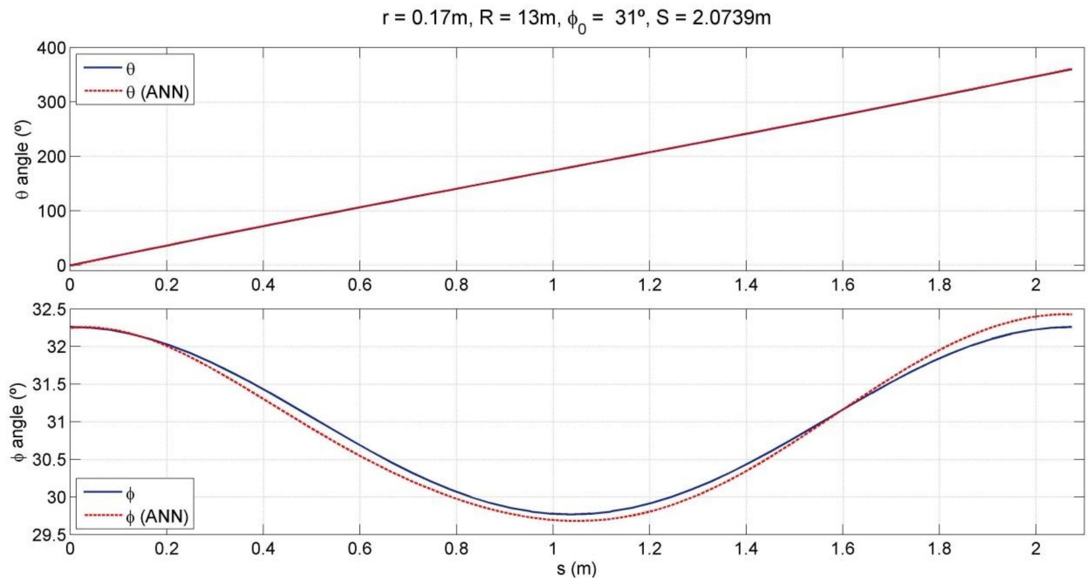

Figure 8. Case: $r = 0.17m, R = 13m, \varphi_0 = 31°, S_{max} = 2.074m$.

The results for a case with the following parameters: $r = 0.35m, R = 13m, \varphi_0 = 22°, S_{max} = 5.871m$ are plotted in Figure 8. Again, the results for the radial angle are better than those for the laying angle, that, in this case, presents worse results than the previous case (Figure 8). In Figure 10, the maximum absolute differences (found along the wire arc) for each case are plotted. The maximum absolute differences found for the radial angle are large, and this is due to the small values at the points in the beginning of the wire arc. The maximum difference for the laying angle is of 11.1%, and occurs for the case presented in Figure 9.

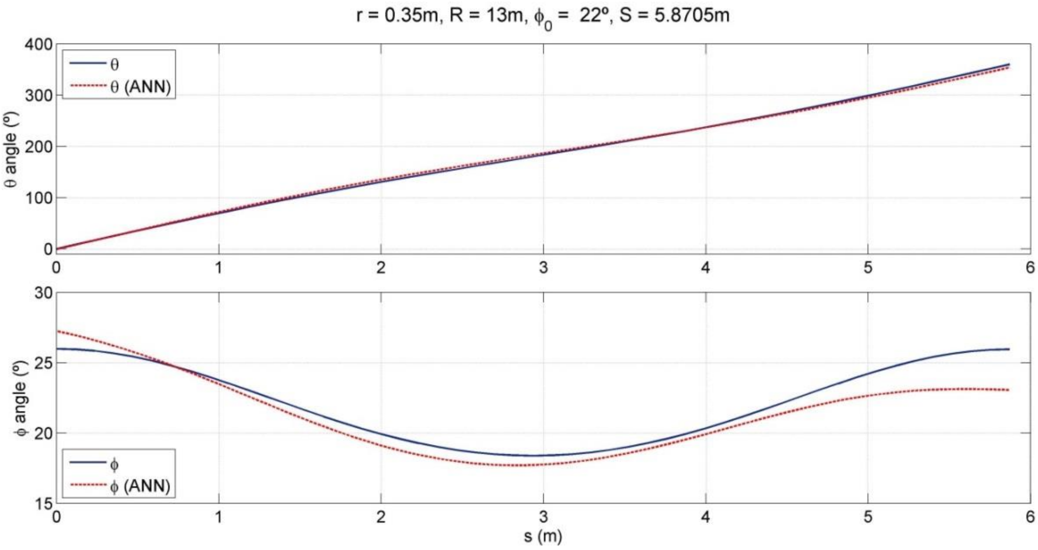

Figure 9. Case: $r = 0.35m, R = 13m, \varphi_0 = 22°, S_{max} = 5.871m$.

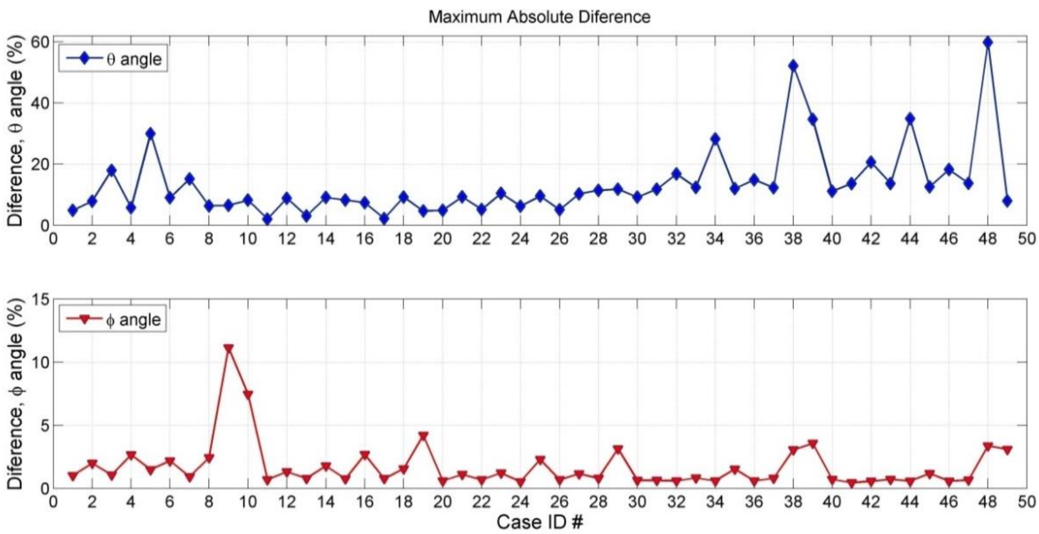

Figure 10. Maximum absolute difference (%) for each testing case.

The correlation coefficient for each case is presented in Figure 11. The values are very close to 1.0, especially for the radial angle θ. The case with the lower correlation for both angles is case 9, which presented the larger difference for the lay angle (Figure 10).

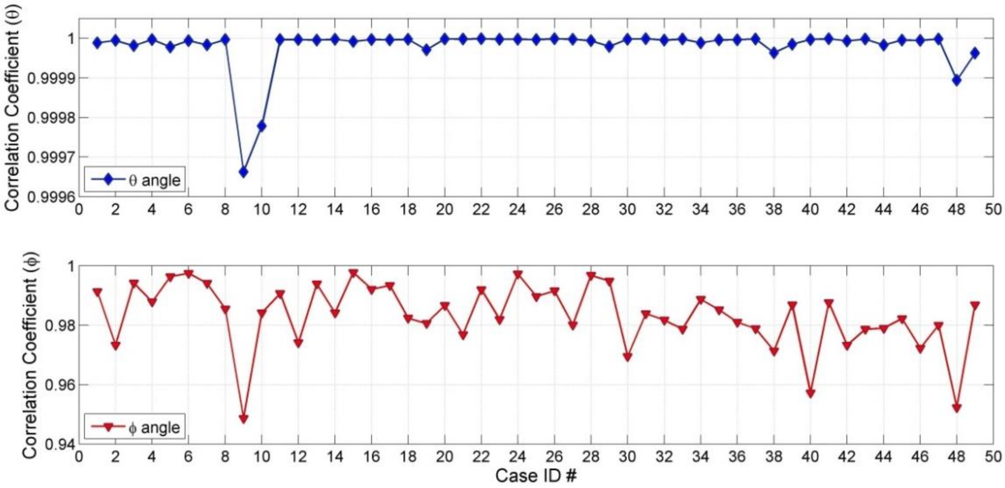

Figure 11. Correlation coefficient for each testing case.

In Figure 12, the cross correlation for all points of all testing cases are presented. This figure also presents, in the legend, the correlation coefficient (R) assessed using all cases. Once again, the results indicate a strong correlation between the ANNs response and the values obtained by numerical integration.

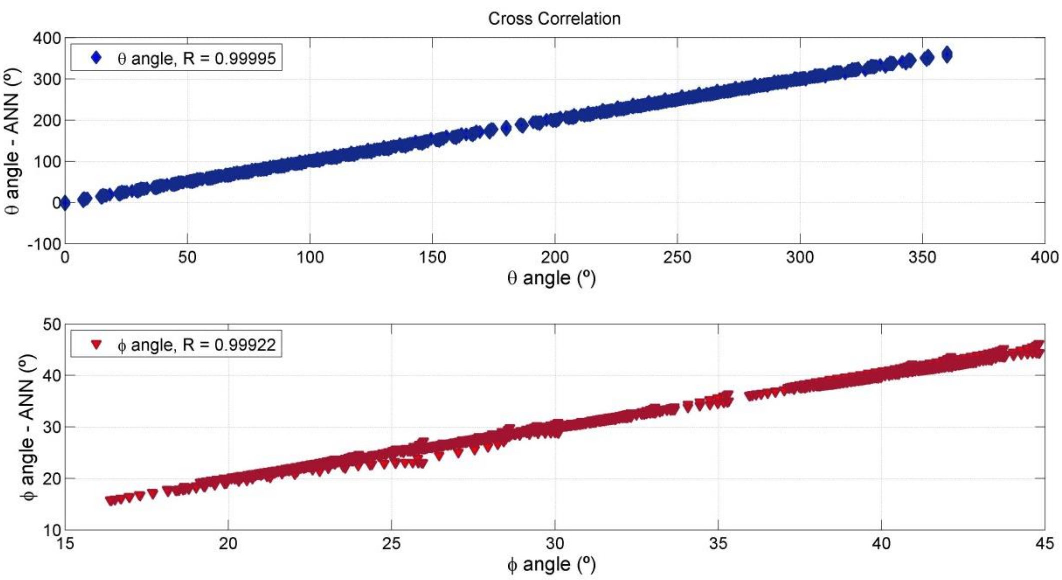

Figure 12. Cross correlation for all testing cases.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

6 CONCLUSIONS AND FINAL REMARKS

In the present work, the stiff nonlinear differential equations of the geodesic curve of a torus are presented and solved by numerical integration for the radial angle θ and for the laying angle φ. From this, a dataset of cases with different geometric wire parameters and imposed bending radius is generated for training ANNs to predict these angles. With the trained ANNs, a new set of cases is generated to test the ANNs performance for cases not present in the training.

The results showed a very good agreement between the ANNs outputs and the results provided by the numerical integration. In practical terms, one could use the ANNs to easily determine the angles of the riser wire for new geometrical configurations and imposed bending, without the need to solve numerically the geodesic differential equations. Moreover, the obtained results may also be utilized in the estimation of curvatures and torsion components, and, therefore, be used in the mechanical analysis of tensile armor wires.

Since the study presented here is in an early stage, some issues have to be analyzed in more detail. One of them is related to the validation data in the network training. In this work, a random selection is performed considering all the wire arc sample cases, while the validation data choice using some wire arc sample cases entirely should turn the validation error more representative. Another concern is related to the activation function employed in the hidden nodes. Here, the hyperbolic tangent was used, but the use of other activation functions (like wavelet functions) could facilitate the approximation in the beginning and in the end of the arc wire helix.

ACKNOWLEDGEMENTS

The authors would like to thank the FAPERJ (Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro) for financial support, as well as to the LACEO-COPPE/UFRJ.

REFERENCES

do Carmo, M. P., 1976. *Differential Geometry of Curves and Surfaces*. Englewwod Cliffs, New Jersey, USA: Prentice-Hall Inc..

Feret, J., Bournazel, C., and Rigaud, J., 1986. Calculation of stresses and slip in structural layers of unbonded flexible pipes. *Proceedings of the 1st OMAE*, New Orleans, USA.

Féret, J. J., Leroy, J. and Estrier, P., 1995. Calculation of Stresses and Slips in Flexible Armour Layers with Layers Interaction. *Proceedings of the 14th OMAE*, 205494, Copenhagen, Denamark.

Gray, A., Abbena, E. and Salamon, S., 1993. *Modern Differential Geometry of Curves and Surfaces with Mathematica*, 3rd Edition, CRC Press Online, 2005. Available at: http://www.crcpress.com

Hagan, M.T.; Demuth, H.B.; Beale, M.H.; de Jesús, O. ISBN 0-9717321-0-8, *Neural Network Design*.

Haykin, S., 2005, Neural Networks - A Comprehensive Foudation. Prentice Hall.

Leroy, J. and Estrier, P., 2001. Calculation of Stresses and Slips in Helical Layers of Dynamically Bent Flexible Pipes. *Oil and Gas Science and Technology*, vol. 56, pp. 545-554

Mathworks, 2014. Neural Network ToolboxTM User´s Guide. EUA, Volume 8.2.

Nutbourne, A. W. and Martin, R. R., 1988. *Differential Geometry Applied to Curve and Surface Design, Vol. 1*. Ellis Horwood Limited.

Out, J. M. M. and von Morgen, B. J., 1997. Slippage of Helical Reinforcing on a Bent Cylinder. *Engineering Structures*, vol. 19, pp. 507-515

Østergaard , N. H., 2012. *On Lateral Buckling of Armouring Wires in Flexible Pipes*. D.Sc. Thesis, Aalborg University, Denmark.

Smith, M., 1993, Neural Networks For Statistical Modeling. Van Nostrand Reinhold, EUA.

de Sousa J. M., Viero P. F., Magluta C., Roitman N., 2012. An Experimental and Numerical Study on the Axial Compression Response of Flexible Pipes. *Journal of Offshore Mechanics and Artic Engineering*, vol. 134, pp. 12.

Sævik, S., 1992. *On Stresses and Fatigue in Flexible Pipes*. D.Sc. thesis, University of Trondheim, Norway.

Sævik, S., 2011. Theoretical and Experimental Studies of Stresses in Flexible Pipes. *Computer & Structures*, vol. 89, pp. 97-106.