

# ANN Based Inverse Dynamic Model of the 6-PGK Parallel Robot Manipulator

L. Moldovan, H.-S. Grif, A. Gligor

**Liviu Moldovan\***, **Horatiu-Stefan Grif**, **Adrian Gligor**

Petru Maior University of Tirgu-Mures

Romania, 540088 Tirgu-Mures, Nicolae Iorga, 1

liviu.moldovan@ing.upm.ro, horatiu.grif@ing.upm.ro, adrian.gligor@ing.upm.ro

\*Corresponding author: liviu.moldovan@ing.upm.ro

**Abstract:** This paper presents an inverse dynamic model estimation based on an artificial neural network of a complete new parallel robot manipulator prototype 6-PGK with six degrees of freedom, built at Petru Maior University of Tirgu-Mures. The model estimation of the parallel robot manipulator is performed with a feedforward artificial neural network. In the control engineering domain there are control structures that need the direct or inverse model of the process for ensuring the process control at the imposed performances. Usually, the determination of the direct/inverse mathematical model is a difficult or impossible task to be achieved. In these cases different non-parametric or parametric, off-line or on-line identification methods are used. A solution that may support the on-line parametric methods is represented by the feedforward artificial neural networks. By implementing feedforward artificial neural networks as a nonlinear autoregressive model with exogenous inputs, the authors investigate the possibility of choosing the optimum parameters that characterize the neural network so that it approximates as better as possible the model of the 6-PGK prototype robot. Finally an innovative algorithm is developed for obtaining the optimal configuration parameters set of the feedforward artificial neural network. The proposed algorithm helps in setting the optimal parameters of the neural network that offer high opportunities to provide satisfactory identification of the robot model. Experimental results obtained by a structure derived from the proposed solution demonstrate a good approximation related to the studied system, which is characterized by nonlinearities and high complexity.

**Keywords:** 6-DOF parallel robot manipulator, inverse dynamics, nonlinear model, unmodeled dynamics, feed-forward artificial neural network.

## 1 Introduction

In robotics, the class of parallel robots represent a constructive solution suitable for many industrial, medical or domestic applications, like flight simulators and entertainment rides [1], micromanufacturing [2], tool machine [3], pick and place operation [4], earthquake motion simulator [4], medical haptic devices [6], laparoscopic surgery [7], or even space docking technology [8].

Even if parallel robots have a reduced workspace [9], their benefits over their serial counterparts are various: stability and rigidity of contacts during haptic interaction [10], high accuracy, high stiffness, high payload capability, low moving inertia [11], good dexterity, compact size, large power to weight ratio [12]. Along with these advantages, there still exist many difficulties in the actual control process.

A classic approach in controlling robots consists in the employment of the dynamic model. Various methods have been proposed to derive the dynamic model of the parallel manipulator, like the generalized momentum approach [13], principle of virtual work [14] screw theory [15] and Hamilton's principle [16], the recursive matrix method [17] or the Lagrangian approach [11], which describes the dynamics of mechanical system from the concepts of work and energy [13].

All these approaches are equivalent as they are describing the same physical system, and lead to equivalent dynamic equations, which present different levels of complexity and associated computational loads.

The analytical calculus involved in the dynamic equation is very tedious, thus presenting an elevated risk for errors. Furthermore, the duration of numerical computations is getting longer when the number of mechanism's degrees of freedom is increased.

The dynamic model based control arises numerous difficulties because of the time-varying and coupling of the equations. Unfortunately, the present-day commercial controllers cannot provide satisfying performance for control.

In recent years an important credit in obtaining new enhancements is granted to the support offered by artificial intelligence techniques.

The role of artificial intelligence related to the parallel robots aims with predilection the modeling and control domain in terms of kinematics or dynamics. Moreover these strategies are designed to improve the classical control strategies, in terms of increasing the current performances, to add new properties such as robustness and improvement of the real-time behavior using fuzzy systems or different types of neural networks (NNET).

Compared to the mathematical model of the kinematics, neural networks provide significantly better mapping between manipulator's task space and joint space, as demonstrated by Uzunovic et al. [2].

Achili et al. in [18] proposed an adaptive force/position controller for parallel robot with constrained motions based on multi-layer perceptron neural network. The solution presents the major advantage of obtaining the control law without the prior knowledge of the inverse dynamic model and being able to take into account the endogenous disturbance (uncertainties and nonlinearities related to the robot dynamics) and to compensate exogenous disturbances.

Peng et al. [19] design a control system for a parallel mechanism consisting from a combination of linear controllers and two neural networks. The aim of this control system is to compensate the nonlinearities from the model of the system and the dynamic model estimation for improved accuracy of trajectory.

Le et al. [11] investigate the enhancement of the classical nonlinear PD torque controller by using neural networks for on-line self gain tuning. The proposed solution consists in a self gain tuning control law for minimizing the error in tracking trajectories of the parallel manipulators characterized by simple structure suitable for low computation effort in real-time implementations.

Yu and Weng in [20] propose a robust control structure for parallel manipulators based on  $H_\infty$  tracking adaptive fuzzy integral sliding mode control scheme mainly able to handle the nonlinear unmodeled dynamics and to compensate external disturbances.

All these solutions benefit from the advantages of various artificial intelligence techniques, but underline the difficulty of implementing them, starting from the the structure, the different parameters values or choice of training methods, and finally reaching the problem of required computational effort.

Recent research in the field of robot control is conducted in new directions based on bio-inspired approach that proved their efficiency on other fields of robotics, like the simplified fuzzy controller using neural network based on genetic algorithm learning for mobile robots [21], the communications and social structure of whale pods applied to cooperative robot structures [22] or fast genetic algorithm for robot soccer and planet exploration citelee. We consider that this direction of research may find applicability in the case of parallel robots, for the potential offered in solving many complex problems such as those in the case of parallel robot control discussed in this paper.

From the above explored literature we conclude that in the case of parallel robots there is a

need to find adequate solutions for the real time control which involve increasing the trajectory following performance and low computational effort.

The objective of the paper is to build and implement an artificial neural network based on the inverse dynamic model estimation of a complete new parallel robot manipulator 6-PGK type with 6 degrees of freedom built at Petru Maior University of Tirgu-Mures.

## 2 The 6-PGK Parallel Robot Dynamics

### 2.1 The 6-PGK Parallel Robot Manipulator Design

The six controlled degrees of freedom (DOF) parallel robot configuration is a structure proposed by Stewart and Hunt and widely referred as Stewart platform [23–26]. The parallel kinematic link system proposed by D. Stewart is one of a 6-DOF robot manipulator used as a flight simulator [25, 26].

The basic geometric structure of the 6-PGK parallel robot manipulator developed at Petru Maior University consists in a mobile platform (supporting the end-effector) connected to the adjacent links at six distinct points ( $i=1,2,\dots,6$ ) at the level III by cardanic kinematic pairs (Figure 1) [27]. Six legs connect the platform to the base. Each leg has an actuated translational

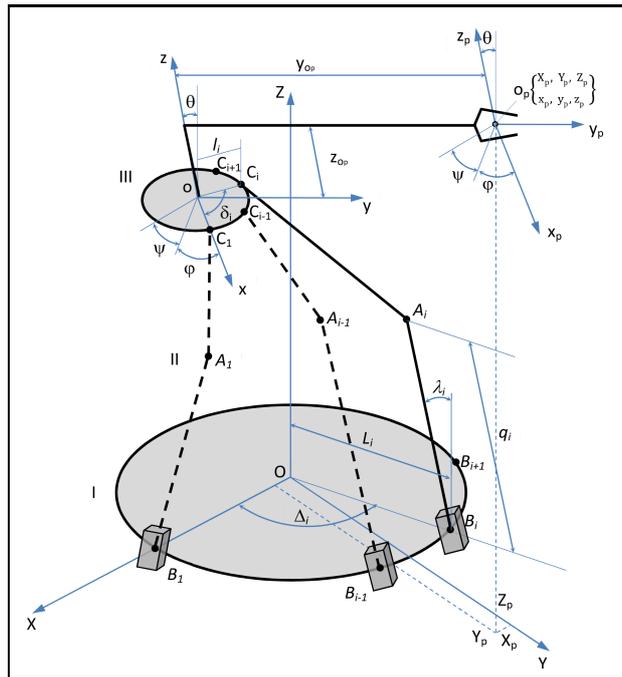


Figure 1: The structure of the 6-PGK parallel robot [27]

kinematic pair at level I and a spherical kinematic pair at level II. The translational kinematic pairs form the angles  $\lambda_i$  with the vertical axis in points  $B_i$ . The location (position and orientation) of the end-effector and of the manipulated object are expressed in the fixed reference frame  $OXYZ$  by the generalized coordinates of the manipulated object  $X_p, Y_p, Z_p, \psi, \phi, \theta$  denoted  $q_{p_i}$  ( $i=1,2,\dots,6$ ), and in the mobile frame  $oxyz$  by the  $x_p, y_p, z_p$  coordinates. The displacements in the actuated translational kinematic pairs (actuator displacements) are the generalized coordinates of the parallel robot  $q_i$  ( $i=1, \dots, 6$ ).

## 2.2 Parallel Robot Manipulator Dynamics Modeling

The dynamics modeling in case of robot manipulators refers to establishing a mathematical model that relates the time evolution of its end-effector in terms of positions, velocity and acceleration with the forces and moments acting upon it [28].

In order to get the position of the mobile platform with respect to generalized coordinates of the robot, the in-out equations are employed, which solve the positional problem of the robot. They relate the generalized coordinates of the robot to the generalized coordinates of the manipulated object [27]:

$$q_i^2 - 2b_i q_i + c_i = 0, \quad (i = 1, \dots, 6) \quad (1)$$

where  $b_i, c_i$  are coefficients according to the robot mechanical parameters [27].

Determination of the velocity and acceleration of the mobile platform  $[\dot{q}_p]$  with respect to generalized coordinates of the robot is performed using the kinematics equations:

$$[\dot{q}_p] = [J][\dot{q}] \quad (2)$$

obtained by deriving the in-out equations (1) with respect to time, where  $[J]$  is the Jacobian matrix [27].

The dynamic modeling determines the generalized forces  $[Q_j]$  that are required by the actuators to balance the forces applied to the robot.

In the case of the 6-PGK parallel robot, by employing the Lagrange method, the dynamic equations are obtained [27]:

$$\left\{ \frac{1}{2} \sum_{k=1}^3 \sum_{l=1}^3 J_{kl} \left( \frac{\partial \omega_k}{\partial \dot{q}_{p_j}} \dot{\omega}_l + \omega_k \frac{\partial \omega_l}{\partial \dot{q}_{p_j}} \right) + M \sum_{k=1}^3 \ddot{q}_{p_k} \frac{\partial \dot{q}_{p_k}}{\partial \dot{q}_{p_j}} + \sum_{i=1}^6 m_i \ddot{q}_i \frac{\partial \dot{q}_i}{\partial \dot{q}_{p_j}} \right\} + \left\{ \begin{array}{l} \frac{1}{2} \sum_{k=1}^3 \sum_{l=1}^3 J_{kl} \left( \frac{\partial \omega_k}{\partial \dot{q}_{p_j}} \dot{\omega}_l + \omega_k \frac{\partial \omega_l}{\partial \dot{q}_{p_j}} \right) - \frac{1}{2} \sum_{k=1}^3 \sum_{l=1}^3 \frac{\partial J_{kl}}{\partial q_{p_j}} \omega_k \omega_l + \\ \frac{1}{2} \sum_{k=1}^3 \sum_{l=1}^3 J_{kl} \left[ \frac{d}{dt} \left( \frac{\partial \omega_k}{\partial \dot{q}_{p_j}} \right) \dot{\omega}_l + \omega_k \frac{d}{dt} \left( \frac{\partial \omega_l}{\partial \dot{q}_{p_j}} \right) - \frac{\partial \omega_k}{\partial q_{p_j}} \dot{\omega}_l - \omega_k \frac{\partial \omega_l}{\partial q_{p_j}} \right] + \\ \sum_{i=1}^6 m_i \left[ \dot{q}_i \frac{d}{dt} \left( \frac{\partial \dot{q}_i}{\partial \dot{q}_{p_j}} \right) - \dot{q}_i \frac{\partial \dot{q}_i}{\partial \dot{q}_{p_j}} \right] \end{array} \right\} + \left\{ Mg \frac{\partial Z_p}{\partial q_{p_j}} + \sum_{i=1}^6 m_i g \frac{\partial q_i}{\partial q_{p_j}} C \lambda_i - m_0 g \left( x_p \frac{\partial \gamma'}{\partial q_{p_j}} + y_p \frac{\partial \gamma''}{\partial q_{p_j}} + z_p \frac{\partial \gamma'''}{\partial q_{p_j}} \right) \right\} = Q_j \quad (3)$$

( $j = 1, \dots, 6$ )

where  $Q_j$  are the generalized forces due to the non-conservative forces,  $m_i$  are masses of the motor links,  $J_{kl}$  are the elements of the inertia matrix of the mobile platform of mass  $m_0$  and manipulated object of mass  $m_p$ , their sum is  $M = m_0 + m_p$ , then  $\alpha', \dots, \gamma'''$  are terms of the rotational matrix that relates the coordinates fixed in mobile frame to the base coordinates and  $\omega_x, \omega_y, \omega_z$  are the angular velocity terms in  $OXYZ$  system.

The generalized motor forces are deduced from equation [27]:

$$Q_j = \sum_{i=1}^6 l_{ij} Q_{m_i} \quad (4)$$

where  $l_{i,j}$  are terms of the inverse Jacobian matrix.

The inverse dynamic model is useful for the robotic system control. It consists in determination in real time of the generalized motor forces  $Q_{m_i}$  from equations (4), by solving the positional problem from equation (1), the kinematic problem from equation (2) and the dynamic model from equation (3).

Parallel robots with a high number of degrees of freedom, in particular six degrees like the 6-PGK parallel robot, are modeled by complex dynamic equations, comprising a large number of parameters. Solving them requires crossing a high number of iterations and selection of admissible real solutions. The complexity of the program, the large number of loops, testing all solutions, selecting the nearest current configuration solution, can induce errors in configuration identification. Therefore in current industrial exploitation, parallel robots control requires the use of simple and effective control methods.

### 3 FANN Based Dynamic Model of the 6-PGK Parallel Robot Manipulator

#### 3.1 FANN Structure for Robot Dynamic Modeling

For the 6-PGK parallel robot manipulator modeling a feed-forward artificial neural network (FANN) with one hidden layer has been designed. The FANN is implemented as a nonlinear autoregressive model with exogenous inputs (NARX) model. The NARX model is based on the linear ARX model, which is commonly used in time-series modeling. [30]

The numbers of the inputs of the used FANN is determined by the numbers of the generalized coordinates of the robot, the number of the generalized forces and the number of theirs considered past values. The numbers of the outputs of the used FANN is given by the numbers of the generalized forces from the dynamical model of the robot.

By using the compact structure of an artificial neuron presented in [29] we can summarize the mathematical processing that occurs inside of an artificial neuron from the used FANN as:

$$a = F\left(\sum_{j=1}^n \omega_j x_j + \theta\right) \quad (5)$$

where are denoted by:  $x_j$  ( $j = 1, \dots, n$ ) - the inputs of the neuron,  $\omega_j$  ( $j = 1, \dots, n$ ) - the weights of the inputs  $x_j$ ,  $\theta$  - the weight of the offset input,  $n$  - the numbers of the inputs of the neuron,  $F$  is the activation function,  $a$  - the activation of the neuron.

The type of the activation function and the numbers of the inputs corresponding to each neuron depend on the layer to which it belongs. For the neurons from the hidden layer are used two types of the activation function: the hyperbolic tangent function (HTF) type and the radial basis function (RBF) type [30]. The numbers of the inputs of the neurons from this layer is given by the maximum lags associated with robot output ( $n_Q$ ) and input ( $n_{q_p}$ ) signals described by relation  $6(n_{q_p} + n_Q)$ , which also represents the number of the inputs of the FANN. The activation function of the neurons from the output layer is the linear type function and the number of the inputs of these neurons equals the number of the neurons from the hidden layer of the FANN. For the training of the FANN the gradient descent backpropagation algorithm was used [30]. The training of the used FANN was achieved by presenting to their inputs and outputs the training set of the  $N$  pairs of the vectors:

$$\{(\underline{x}^i(1), \underline{a}_d^o(1)), (\underline{x}^i(2), \underline{a}_d^o(2)), \dots, (\underline{x}^i(N), \underline{a}_d^o(N))\}$$

where  $\underline{x}^i(k)$  is an input training vector ( $k = 1, \dots, N$ ) with  $6(n_{q_p} + n_Q)$  elements and  $\underline{a}_d^o(k)$  is the desired target output vector (with 6 elements) of the FANN corresponding to the input training

vector  $x^i(k)$ .

### 3.2 FANN Optimal Parameters Determination

Analytical determination of the FANN parameters is an uncommon task, thus the empirical approach is more often used. The optimum determination however is a more complex task, so in this section a tested methodology for determination of these parameters is proposed.

Usually, the investigation of a system starts from the searching of the parameters that influence the behavior of the studied system. The same approach was established for the present work. In the case of the proposed structure of neural network the following four parameters were considered: learning rate ( $\gamma$ ), the number of the neurons on the hidden layer ( $n_h$ ), the number of the past values of the generalized forces ( $n_Q$ ) and the number of the past values of the generalized coordinates ( $n_q$ ). In order to study the influence of these parameters on the neural network behavior for building the FANN training data sets the generalized coordinates trajectory of the robot and the corresponding trajectories for generalized motor forces will be used.

An adequate solution for the process of searching through the experimental set can be performed by visual analysis of all the available results (large sets of plotted trajectories). But this approach presents a major drawback from time needed for analysis and required storage resources. In order to facilitate and automatize the evaluation process, quality measures can be used. An option is represented by the mean squared error or the mean absolute error. However, both measures present major difficulty in the case of establishing of a threshold value to aid in deciding the quality of the robot model identification. In order to overcome the mentioned drawbacks for the evaluation process the authors propose an automated method and the usage of a quality measure based on signal-to-noise ratio (SNR). For the 6-PGK prototype robot the SNR is given by the following equation:

$$SNR_i = 10 \cdot \log_{10} \frac{\sum_{j=1}^N \left( Q_{mi}(j) \right)^2}{\sum_{j=1}^N \left( Q_{mi}(j) - Q_{mi_{FANN}}(j) \right)^2} [dB], \quad (i = 1, \dots, 6) \quad (6)$$

where  $Q_{mi}$  represents the generalized motor trajectory forces given by the inverse dynamical model of the robot,  $Q_{mi_{FANN}}$  represents the generalized motor trajectory forces given by the FANN and  $N$  is the number of the trajectory samples.

The proposed SNR quality measured was derived from the signal-to-noise ratio used for evaluation of unidimensional electrical signals.

The algorithm for FANN optimal parameters determination is presented below.

#### General Algorithm for Choosing the Structure of the FANN

Let us consider the following parameters which influence the behavior of the FANN: learning rate ( $\gamma$ ), the number of the neurons ( $n_h$ ) from hidden layer, the number of the past values for the input ( $n_{q_p}$ ) and output ( $n_Q$ ) variables employed to build the robot regressors vectors (input vectors of the FANN) used in training and simulation of the FANN. These parameters are referred as "FANN parameters". The steps of the proposed algorithm for choosing the structure of the FANN are:

*Step 0.* Select (experimentally): the number of the initial sets of values for the FANN weights and the corresponding weights values (that are random real numbers in the range  $[-0.5, 0.5]$ ), having built a collection of initial sets of values for the FANN weights, the

signal-to-noise ratio threshold  $SNR_{thold}$ , the sets of values for the FANN parameters. To each set of values of the FANN parameters is attached a set of counter variables of the same size as the size of the value of the considered set. Thus for a FANN parameter to each value (from the set of values of the considered FANN parameter) it corresponds a counter variable. The value of the counter variables is initialized with zero;

*Step 1.* Set the following values of the FANN parameters with start values  $n_h = n_{hStart}$ ,  $n_Q = n_{QStart}$ ,  $n_{qp} = n_{qpStart}$ . These values are drawn from the sets of the FANN parameter values;

*Step 2.* Keeping constant the values of the FANN parameters  $n_h = n_{hStart}$ ,  $n_Q = n_{QStart}$ ,  $n_{qp} = n_{qpStart}$  establish the optimal learning rate  $\gamma_{opt}$  based on the Algorithm for Evaluation and Determination of the Optimum Value for the FANN parameters;

*Step 3.* Keeping constant the value of the parameters  $\gamma = \gamma_{opt}$ ,  $n_Q = n_{QStart}$ ,  $n_{qp} = n_{qpStart}$  establish the optimal number of hidden layer neurons ( $n_{hopt}$ ) based on the Algorithm for Evaluation and Determination of the Optimum Value for the FANN parameters;

*Step 4.* Keeping constant the value of the FANN parameters  $gamma = \gamma_{opt}$ ,  $n_h = n_{hopt}$  establish optimal values of the number of the past samples for the input and output variables of the robot model ( $n_{Qopt}$ ,  $n_{qpopt}$ ) based on the Algorithm for Evaluation and Determination of the Optimum Value for the FANN parameters to which the following changes are made corresponding to the steps *Step 4''*, *Step 6''* and *Step 7*;

*Step 5.* Keeping constant the value of the FANN parameters  $gamma = \gamma_{opt}$ ,  $n_h = n_{hopt}$  and  $n_q = n_{qopt}$  it is tested the estimation and identification algorithm considering other trajectories for input and output variables of the robot (or model of the robot).

### Algorithm for Evaluation and Determination of the Optimum Value for the FANN

For each value from the set of values corresponding to the FANN parameter set to be evaluated, the following steps will be executed separately for the each set of the initial weights:

*Step 1'.* Initialize the weights of the FANN with the corresponding initial selected set of weights;

*Step 2'.* Run the estimation and identification algorithm calculating the signal-to-noise ratio (SNR) based on the relation (6) for each output path (trajectory corresponding to a generalized forces), thus corresponding to each set of initial weights, a set of six values of the noise-to-signal ratio will be obtained;

*Step 3'.* Compute the average of the signal-to-noise ratio  $SNR_{med}$  as mean value of the six signal-to-noise ratio determined in *Step 2*;

*Step 4'.* Compare the value of  $SNR_{med}$  with the threshold value  $SNR_{thold}$ . If the average is greater than the threshold then increment the counter variable corresponding to the current value of the FANN parameter and go to *Step 5'*;

*Step 4''.* Compare the value of the  $SNR_{med}$  with the threshold value  $SNR_{thold}$ . If the average value is greater than the threshold value, then increment the counter variable corresponding to the current values combination for the two FANN parameters;

*Step 5I.* If the sets of initial values for weights are not exhausted select a new set of values for weights and jump to *Step 1I*. If the sets of initial values for weights were exhausted continue with *Step 6I*;

*Step 6I.* Using the values of the counter variables, the success rates corresponding to each FANN parameter value is computed by

$$SR_i = \frac{\text{counter}_i}{\text{no. of the initial weights sets}} \cdot 100 [\%]$$

where  $i = 1, \dots, \text{no. of the FANN parameter values}$ . Go to *Step 7I*;

*Step 6II.* Using the values of the counter variables, the success rates corresponding to each combination of the FANN parameters values are computed and go to *Step 7II*;

*Step 7I.* Choose the FANN parameter value for which the success rate is the largest. Finally this value will be considered as the optimal value for the evaluated FANN parameter;

*Step 7II.* Choose a pair of values of the two FANN parameters for which the success rate is the largest. Finally the values of the selected pair will be considered as the optimal for the two parameters of the FANN evaluated.

### 3.3 Experimental Results and Discussions

For the implementation and testing of the parallel robot modeling, Matlab platform and its Neural Network Toolbox were used. In order to investigate the performance of the FANN based modeling the model of the parallel robot based on equations 1 and 2 was implemented.

In Figure 2 six desired trajectories are presented for the generalized coordinates of the manipulated object ( $X_p, Y_p, Z_p, \psi, \phi, \theta$  denoted  $q_{p_i}, i = 1, \dots, 6$ ). These trajectories are placed in parallel surfaces having the same 3D spatial shape at different elevation. These trajectories represent the input trajectories for the Inverse Dynamic Model of the 6-PGK Parallel Robot Manipulator.

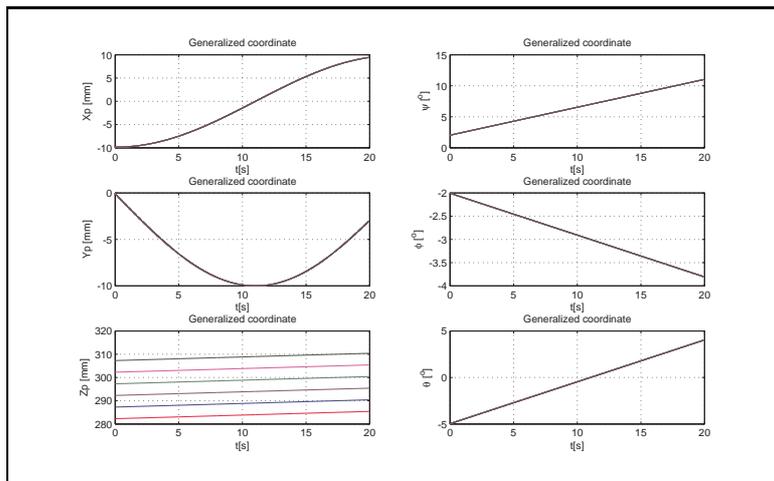


Figure 2: The generalized coordinates trajectories of the 6-PGK parallel robot

In order to study the influence of the FANN parameters on the neural network behavior for building the FANN training data sets, one of the spatial generalized coordinates trajectory

was used, which has the red  $Z_p$  line from Figure 2 and the corresponding red trajectories for generalized motor forces from Figure 3.

In Figure 3 six desired trajectories for the generalized motor forces ( $Q_{mj}$ ,  $j = 1, \dots, 6$ ) are presented. These trajectories represent the output trajectories of the Inverse Dynamic Model of the 6-PGK Parallel Robot Manipulator corresponding to the input trajectories from Figure 2.

The training data are set up using the trajectories presented in Figures 2 and 3, trajectories that consist from 2001 points each.

The estimation and identification algorithm is applied in each time step and consists of two phases:

- the estimation phase - based on the current input vector, the FANN computes the current estimated vector of generalized forces;
- the identification phase - based on current input/output training vectors the FANN is trained.

An input training vector ( $\underline{x}_i(k)$ ,  $k = 0, \dots, N - 1$ ) and an output training vector ( $\underline{a}_d^o(k)$ ,  $k = 0, \dots, N - 1$ ), from the training vectors set, are constructed as follow:

$$\begin{aligned} \underline{x}^i(k) &= [x_1^i(k-1) \dots x_6^i(k-1) \dots x_{6(n_{qp}+n_Q-1)+1}^i(k-n_Q) \dots x_{6(n_{qp}+n_Q)}^i(k-n_Q)] \\ &= [X_p(k-1) \dots \theta(k-1) \dots X_p(k-n_{qp}) \dots \theta(k-n_{qp}) \\ &\quad \dots Q_{m1}(k-1) \dots Q_{m6}(k-1) \dots Q_{m1}(k-n_Q) \dots Q_{m6}(k-n_Q)] \end{aligned} \quad (7)$$

$$\underline{a}_d^o(k) = [a_{d1}^o \quad a_{d2}^o \quad \dots \quad a_{d6}^o] = [Q_{m1}(k) \quad Q_{m2}(k) \quad \dots \quad Q_{m6}(k)] \quad (8)$$

The order of the vectors in the vectors training set corresponds to the order of the time sample from the time axis. Taking into account this observation, the input training vector from (7) and the training output vector from (8) can be expressed in relation with time:

$$\underline{x}^i(t) = [x_1^i(kT - T) \quad \dots \quad x_6^i(kT - T) \quad \dots \quad x_{6(n_{qp}+n_Q-1)+1}^i(kT - n_Q T) \\ \dots \quad x_{6(n_{qp}+n_Q)}^i(kT - n_Q T)] \quad (9)$$

$$\underline{a}_d^o(t) = \underline{a}_d^o(kT) = [Q_{m1}(kT) \quad Q_{m2}(kT) \quad \dots \quad Q_{m6}(kT)] \quad (10)$$

where  $T$  is the sampling time.

The results obtained by using the above algorithms are presented as follows. The experimental study was started from the following FANN parameters values:  $n_h = 5$ ,  $n_Q = 1$ ,  $n_{qp} = 1$ , 20 initial weights sets values,  $SNR_{threshold} = 44.5[dB]$ ,  $\gamma = 0.01, 0.05, 0.1, \dots, 0.98, 1$ . During the *Step 2* of the general algorithm the success rates of  $\gamma$  were computed and then synthesized in Figure 4. The analysis of Figure 4 led to the selecting of the value of 0.25 as the optimal value for the  $\gamma$  parameter. During *Step 3* the success rates of  $n_h$  were computed and then synthesized in Figure 5. The analysis of Figure 5 led to the selecting of the value of 7 as the optimal value for the  $n_h$  parameter. Further, after the evaluation of *Step 4* the results are synthesized in Table 1. The values of the lag for  $n_Q$  are taken successively from 1 to 5. The values for  $n_{qp}$  are taken successively from 1 to  $n_Q$ .

Analyzing the experimental results presented in Table 1 were selected as optimal values  $n_{qp} = 1$  and  $n_Q = 1$ .

In case of the FANN with RBF the proposed algorithm generate the following parameters:  $\gamma = 0.25$ ,  $n_h = 6$ ,  $n_Q = 1$  and  $n_{qp} = 1$ .

Comparing data from Table 1 and Table 2, Figures 4, 5, 6 and 7 shows that FANN with RBF offer a better chance of achieving a good estimation even if the learning rate hasn't a 100% success rate.

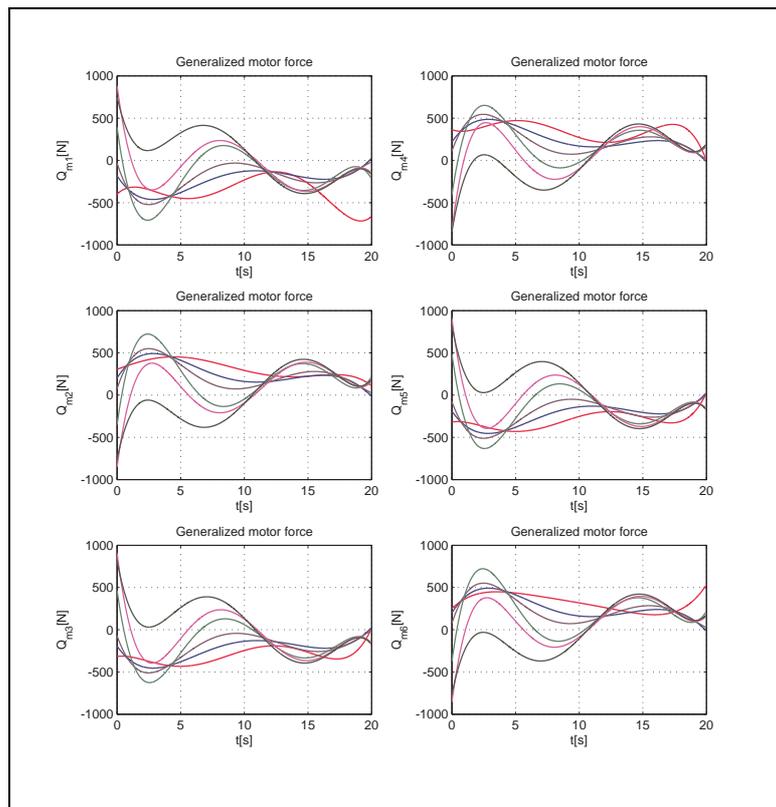


Figure 3: The generalized motor forces of the 6-PGK parallel robot

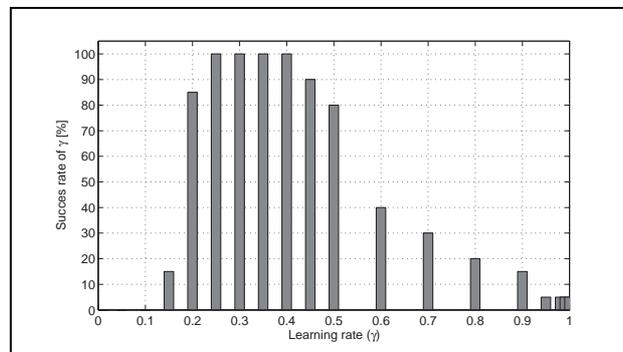
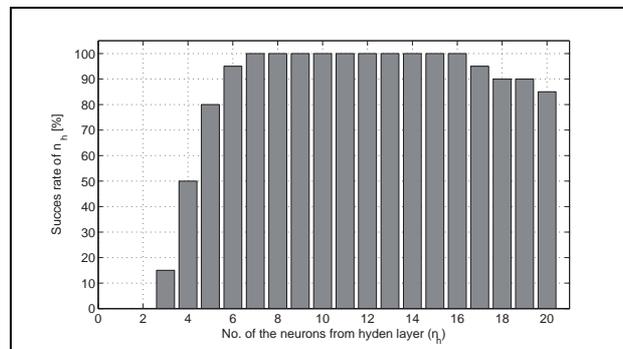
Figure 4: Success rates of  $\gamma$  parameter in case of using HTF in hidden layerFigure 5: Success rates of  $n_h$  parameter in case of using HTF in hidden layer ( $\gamma = 0.25$ )

Table 1: The success rates corresponding to  $(n_Q, n_{qp})$  (FANN: HTF,  $n_h = 7, \gamma = 0.25$ )

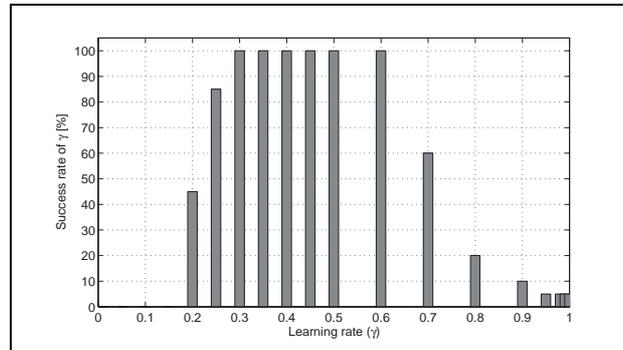
Success rates [%]		$n_{qp}$				
		1	2	3	4	5
$n_Q$	1	100	-	-	-	-
	2	100	75	-	-	-
	3	90	70	35	-	-
	4	60	55	15	20	-
	5	45	35	20	20	10

Table 2: The success rates corresponding to  $(n_Q, n_{qp})$  (FANN: RBF,  $n_h = 6, \gamma = 0.25$ )

Success rates [%]		$n_{qp}$				
		1	2	3	4	5
$n_Q$	1	100	-	-	-	-
	2	95	85	-	-	-
	3	85	85	65	-	-
	4	75	70	45	40	-
	5	60	65	40	30	15

In order to evaluate the behavior of the built FANN, in Figure 8 a test motor forces trajectories of the parallel robot and the estimated trajectory generated by the FANN are presented comparatively.

In Figure 9 the estimation error trajectories related to the trajectories from Figure 8 is shown. The experimental results demonstrate a good approximation related to the studied system which is characterized by nonlinearities and high complexity.

Figure 6: Success rates of  $\gamma$  parameter in case of using RBF in hidden layer

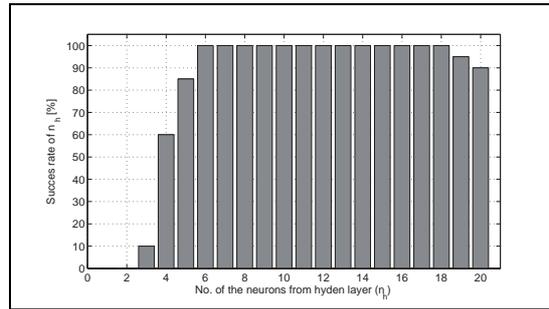


Figure 7: Success rates of  $n_h$  parameter in case of using RBF in hidden layer ( $\gamma = 0.25$ )

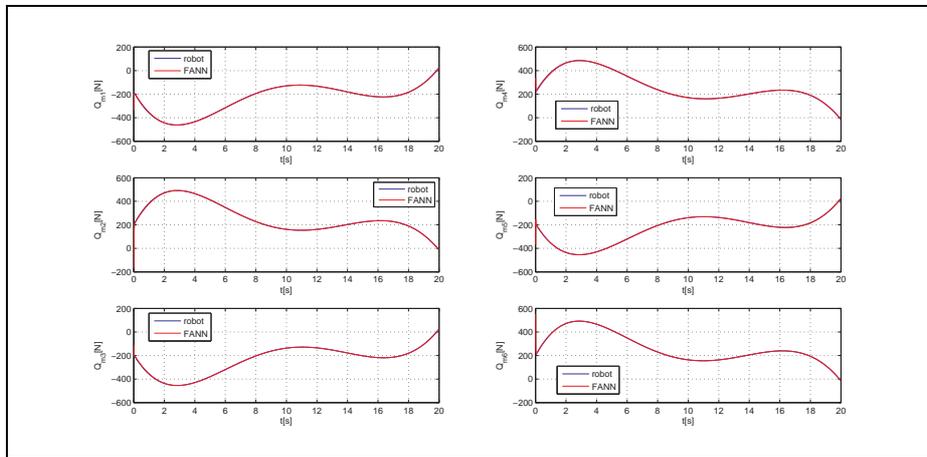


Figure 8: The FANN estimated motor forces trajectory and the motor forces trajectory of the 6-PGK parallel robot (FANN: RBF,  $n_h = 6$ ,  $\gamma = 0.25$ )

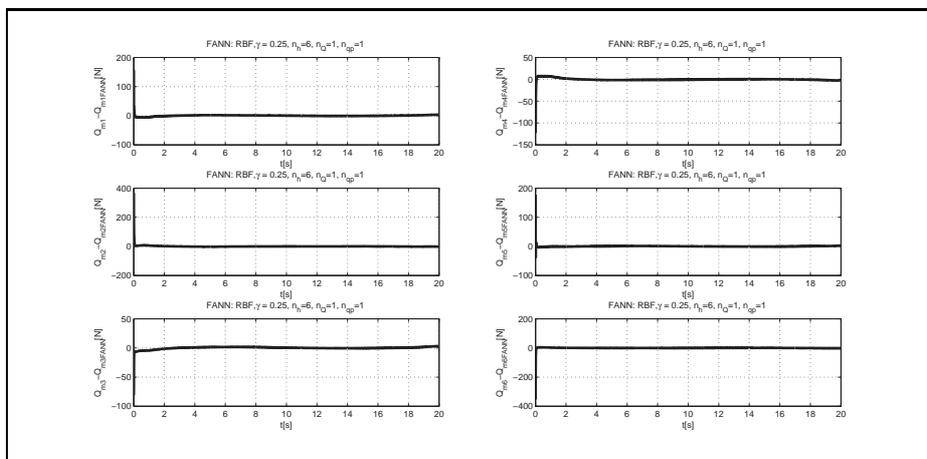


Figure 9: Estimation error trajectory corresponding to Figure 8

## 4 Conclusions

This paper proposed to build an on-line feedforward artificial neural network with the aim of estimating the inverse dynamic model of the 6-PGK prototype parallel robot.

The presented solution was adopted in order to perform an on-line parametric identification of the inverse dynamic model of the robot. The chosen solution was adopted mainly for the use when the robots are characterized by the nonlinearities and high mathematical model complexity.

The implementation of the feedforward artificial neural networks allowed to obtain a nonlinear autoregressive model with exogenous input behavior, for which a new method for finding optimum parameters that approximate as better as possible the model of the 6-PGK robot was obtained. In addition, the proposed solution offers, with the best estimation results, the minimal computational load structure of the neural network.

Even if the FANN has a minimal internal structure, a slower training learning method, it offers a robust and efficient estimation method of the parallel robot motor trajectories, despite its complex and nonlinear mathematical model.

## Bibliography

- [1] Hamlin, G.J.; Sanderson, A.C. (1997); TETROBOT: a modular approach to parallel robotics, *IEEE Robotics & Automation Magazine*, ISSN 1070-9932, 4(1): 42-50.
- [2] Uzunovic, T.; Golubovic, E.; Baran, E. A.; Sabanovic, A. (2013); Configuration space control of a parallel Delta robot with a neural network based inverse kinematics, *IEEE Proc. of 2013 8th Intl. Conf. on EEE (ELECO)*, 497-501.
- [3] Abdellatif H.; Heimann, B. (2010); Experimental identification of the dynamics model for 6-DOF parallel manipulators, *Robotica*, 28:359-368. doi:10.1017/S0263574709005682.
- [4] Yang, Z.; Wu, J.; Mei, J.; Gao, J.; Huang, T. (2008); Mechatronic model based computed torque control of a parallel manipulator, *International Journal of Advanced Robotic Systems*, 5(1):123-128.
- [5] Zhang, J.; Yu, H.; Gao, F.; Zhao, X.; Ma, C.; Huang, X. (2010); Application of a novel 6-DOF parallel robot with redundant actuation for earthquake simulation, *IEEE Proc. on Intl. Conf. on Robotics and Biomimetics (ROBIO)*, 513-518.
- [6] Najafi, F.; Sepehri N. (2008); A novel hand-controller for remote ultrasound imaging, *Mechatronics*, 18: 578-590.
- [7] Ibrahim, K.; Ramadan, A.; Fanni, M.; Kobayashi, Y.; Abo-Ismael, A.A.; Fujie, M.G. (2012); Design and workspace analysis of a new endoscopic parallel manipulator, *IEEE Proc of 2012 12th Intl. Conf. on Control, Automation and Systems (ICCAS)*, 688-693.
- [8] Jian, X.; Zhiyong, T.; Zhongcai, P.; Shao, H.; Lanbo, L. (2013); Adaptive controller for 6-DOF parallel robot using TS fuzzy inference, *International Journal of Advanced Robotic Systems*, 10(119):1-9.
- [9] Moldovan, L. (2008); Geometrical Method for Description of the 6-PGK Parallel Robot's Workspace, *IEEE Proc. of First Intl. Conf. on Complexity and Intelligence of the Artificial and Natural Complex Systems, Medical Applications of the Complex Systems, Biomedical Computing*, 2008. CANS'08, 45-51.

- [10] Constantinescu, D.; Salcudean, S.E.; Croft E.A. (2005); Haptic Rendering of Rigid Contacts Using Impulsive and Penalty Forces, *IEEE Transactions On Robotics*,21(3):309-323.
- [11] Le, T. D.; Kang, H. J.; Suh, Y. S.; Ro, Y. S. (2013); An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator, *Neurocomputing*, 116: 53-61.
- [12] Lu, Y.; Li, X.P. (2014); Dynamics analysis for a novel 6-DoF parallel manipulator I with three planar limbs, *Advanced Robotics*, 28(16): 1121-1132. DOI: 10.1080/01691864.2014.908743.
- [13] Lopes, A.M. (2010); Complete dynamic modelling of a moving base 6-dof parallel manipulator, *Robotica*, 28:781-793, doi:10.1017/S0263574709990506.
- [14] Staicu, S.; Liu X.-J.; Wang, J. (2007); Inverse dynamics of the HALF parallelmanipulator with revolute actuators, *Nonlinear Dyn.*, 50: 1- 12.
- [15] Gallardo J.; Rico, J.; Frisoli, A.; Checcacci, D.; Bergamasco M. (2003); Dynamics of parallel manipulators by means of screw theory, *Mech. Mach. Theory*, 38: 1113-1131.
- [16] Miller, K. (2004); Optimal Design and Modeling of Spatial Parallel Manipulators, *The Int. J. of Robotics Research*, 23(2): 127-140.
- [17] Staicu, S.; Zhang, D.; Rugescu, R. (2006); Dynamic modelling of a 3-DOF parallel manipulator using recursive matrix relations, *Robotica*, 24:125-130, doi:10.1017/S0263574705001670.
- [18] Achili, B.; Daachi, B.; Amirat, Y.; Ali-Cherif, A.; DaËchi, M. E. (2012); A stable adaptive force/position controller for a C5 parallel robot: a neural network approach, *Robotica*, doi:10.1017/S0263574711001354, 30(7):1177-1187.
- [19] Peng, Z.; Liu, F.; Yang, L. (2010); Control based on double neural networks-PI for parallel mechanism, *Robotics and Computer-Integrated Manufacturing*, 26(3): 250-252.
- [20] Yu, W. S.; Weng, C. C. (2014);  $H^\infty$  tracking adaptive fuzzy integral sliding mode control for parallel manipulators; *Fuzzy Sets and Systems*, ISSN 0165-0114, 248: 1-38.
- [21] Obe, O.; Dumitrache, I. (2012); Adaptive Neuro-Fuzzy Controller With Genetic Training For Mobile Robot Control, *International Journal od Computers Communications & Control*, 7(1): 135-146.
- [22] Resceanu, I. C.; Resceanu, C. F.; Bizdoaca, N. G. (2012); Cooperative Robot Structures Modeled After Whale Behavior and Social Structure, *International Journal od Computers Communications & Control*, 7(5):945-956.
- [23] Liu, X.; Wang, Q.; Malikov, A.; Wang, H. (2012); The design and dynamic analysis of a novel 6-DOF parallel mechanism, *International Journal of Machine Learning and Cybernetics*, 3(1): 27-37.
- [24] Pei, Z.; Zhang, Y.; Tang; Z. (2007); Model reference adaptive PID control of hydraulic parallel robot based on RBF neural network, *IEEE Proc. of Intl. Conf.on Robotics and Biomimetics, ROBIO 2007*, 1383-1387.
- [25] Beji, L.; Pascal, M. (1999); The kinematics and the full minimal dynamic model of a 6-DOF parallel robot manipulator, *Nonlinear Dynamics*, 18(4): 339-356.

- [26] Ayas, M. S.; Sahin, E.; Altas, I. H. (2014); Trajectory tracking control of a Stewart platform, *Proc. of IEEE 2014 16th Intl. Conf. and Exposition on Power Electronics and Motion Control (PEMC)*, 720-724.
- [27] Moldovan, L. (2008); Trajectory Errors of the 6-PGK Parallel Robot, *IEEE Proc. of First Intl. Conf. on Complexity and Intelligence of the Artificial and Natural Complex Systems, Medical Applications of the Complex Systems, Biomedical Computing*, 2008. CANS'08, 31-37.
- [28] Mendes Lopes, A.; Almeida, F. (2009); The generalized momentum approach to the dynamic modeling of a 6-dof parallel manipulator, *Multibody System Dynamics*, 21(2): 123-146.
- [29] Grif, S. (2014); Automatic daylight control system based on neural estimator, *Procedia Technology*, 12: 759-765.
- [30] Demut, H.; Beale, M.; Hagan, M. (2007); *Neural Network Toolbox For Use with Matlab. User's Guide. Version 5*, 2007, The MathWorks, Inc.