Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2015, Article ID 906126, 16 pages
http://dx.doi.org/10.1155/2015/906126

*Research Article*

# Enhanced Dynamic Model of Pneumatic Muscle Actuator with Elman Neural Network

**Alexander Hošovský, Ján Pitel', and Kamil Židek**

*Faculty of Manufacturing Technologies with a Seat in Prešov, Technical University of Košice, Bayerova 1, 080 01 Prešov, Slovakia*

Correspondence should be addressed to Alexander Hošovský; alexander.hosovsky@tuke.sk

Received 16 October 2014; Accepted 22 January 2015

Academic Editor: Mathiyalagan Kalidass

To make effective use of model-based control system design techniques, one needs a good model which captures system's dynamic properties in the range of interest. Here an analytical model of pneumatic muscle actuator with two pneumatic artificial muscles driving a rotational joint is developed. Use of analytical model makes it possible to retain the physical interpretation of the model and the model is validated using open-loop responses. Since it was considered important to design a robust controller based on this model, the effect of changed moment of inertia (as a representation of uncertain parameter) was taken into account and compared with nominal case. To improve the accuracy of the model, these effects are treated as a disturbance modeled using the recurrent (Elman) neural network. Recurrent neural network was preferred over feedforward type due to its better long-term prediction capabilities well suited for simulation use of the model. The results confirm that this method improves the model performance (tested for five of the measured variables: joint angle, muscle pressures, and muscle forces) while retaining its physical interpretation.

## 1. Introduction

Pneumatic artificial muscle (PAM) is a special type of actuator that converts energy of pressurized air within the muscle into mechanical energy in the form of its contraction. The actuator has some interesting properties that make it attractive for certain specific applications, which might require powerful actuators with clean operation. As such, it remains one of the few nonconventional actuators that were transformed from a laboratory prototype into fully fledged reliable and viable commercial product (e.g., FESTO fluidic muscles). The issue of PAM modeling was pivotal in its research since the very beginning and it resulted in several different model types which, more or less, characterize the modeling approaches in all works regarding PAMs, namely, geometrical model, empirical model, and phenomenological model [1]. The geometrical model represents a traditional approach to PAM modeling, which relies on geometrical parameters that can be difficult to measure accurately. This modeling approach was used in enhanced form in [2, 3], where the dependence of fiber length on muscle pressure and the friction between fibers themselves were taken into account. Researchers in [4] presented a detailed analytical model of PAM derived using its geometrical and mechanical properties without need for any experimentally determined parameters. A different approach was chosen in [5], where a dynamic phenomenological model of PAM was derived using the analogy with Hill's model of biological muscle. In [6] a combined empirical-geometric model of PAM based on the nonlinear spring-damper analogy intended for a robotic application was developed. Solely empirical model was derived in [7] and this was used in [8] for the analysis of mechanical properties of PAM in mechatronic systems. Some works included the development of PAM model within the task of model-based control design where the particular approach selected for PAM modeling differed based on the application needs and experiment conditions. In [9] an empirical PAM model with improved force modeling was presented as a part of work on cascaded control concept for two degrees of freedom pneumatic muscle robot. Similar problem of model-based control design for PAM-based manipulator control was dealt with in [10], where researchers used feedforward compensation of actuator hysteresis based on the Preisach hysteresis model. Based on the empirical model of PAM, researchers in [11] developed a state-space model used for a model-based nonlinear control algorithm design. An interesting approach is presented in [12], where a novel model based on Bouc-Wen differential equation was used to describe the hysteretic behavior of PAM actuator and, on the basis of this model, a new cascaded control concept was tested.

Our previous research concentrated on developing a dynamic model of pneumatic artificial muscle with application to PAM-based actuator modeling with one degree of freedom (one rotational joint) driven by a pair of FESTO fluidic muscles. The basic dynamic model of this system intended for industrial applications was developed in [13], where preliminary experiments in open-loop mode were carried out. In addition to this, works [14, 15] were dedicated to the analysis of fundamental static and dynamic properties of PAM-based actuator and its operating modes. In [16], the analytical model of PAM-based actuator developed in [13] was used for model-based adaptive fuzzy control design using genetic algorithms. Also in [17], the properties of pneumatic artificial muscle were analyzed from mathematical point of view with their description given in the form of nonlinear differential equations.

Our enhanced dynamic model is based on the analytical model derived in [13], where mechanical part of the PAM model is treated as a parallel combination of nonlinear spring and nonlinear damper in accordance with [6]. It is shown that the model itself, in general, is capable of capturing the nature of actuator's dynamics, yet its accuracy suffers from quite a number of simplifications made in the description of rather complex physical phenomena. For the comparison of model and actuator responses, we selected five variables for which appropriate sensors were available and which were considered important in view of the further research: joint angle, muscle pressures, and muscle forces. It was also considered important to take into account the effect of varying moment of inertia which is, in general, unpredictable and proposed control system should be robust enough to provide satisfactory performance under this condition. Results of open-loop mode model validation helped to uncover deviations between the actuator and model performance originating from the unmodeled effects in physical description of the system with the magnitude of their manifestation (at least in terms of oscillations detected in the measured responses) being proportional to the moment of inertia represented by a load weight attached to the actuator arm. In view of these effects' nature, it was proposed here to use powerful approximation capabilities of neural networks to tackle the issue of complex physical phenomena description. It has to be noted that even though it might have been possible to abandon analytical approach to PAM-based actuator modeling completely, it was deemed advantageous to retain it and only make use of neural network in combination with primary (analytical) model. In this way, the physical interpretation of the model was preserved and unmodeled effects which were to be approximated by neural network were treated as external disturbance. As to the neural network type, recurrent network was preferred over the series-parallel (nonlinear autoregressive model with exogenous input, NARX) configuration due to its better suitability for simulation tasks. The model was intended to be used for control system design, where no input in the form of actual variable value (as in series-parallel configuration without feedback) was expected. The results confirmed usefulness of this approach in predicting actuator's dynamic characteristics even in closed-loop mode, where unmodeled effects caused larger deviations of the model performance compared to the actuator performance.

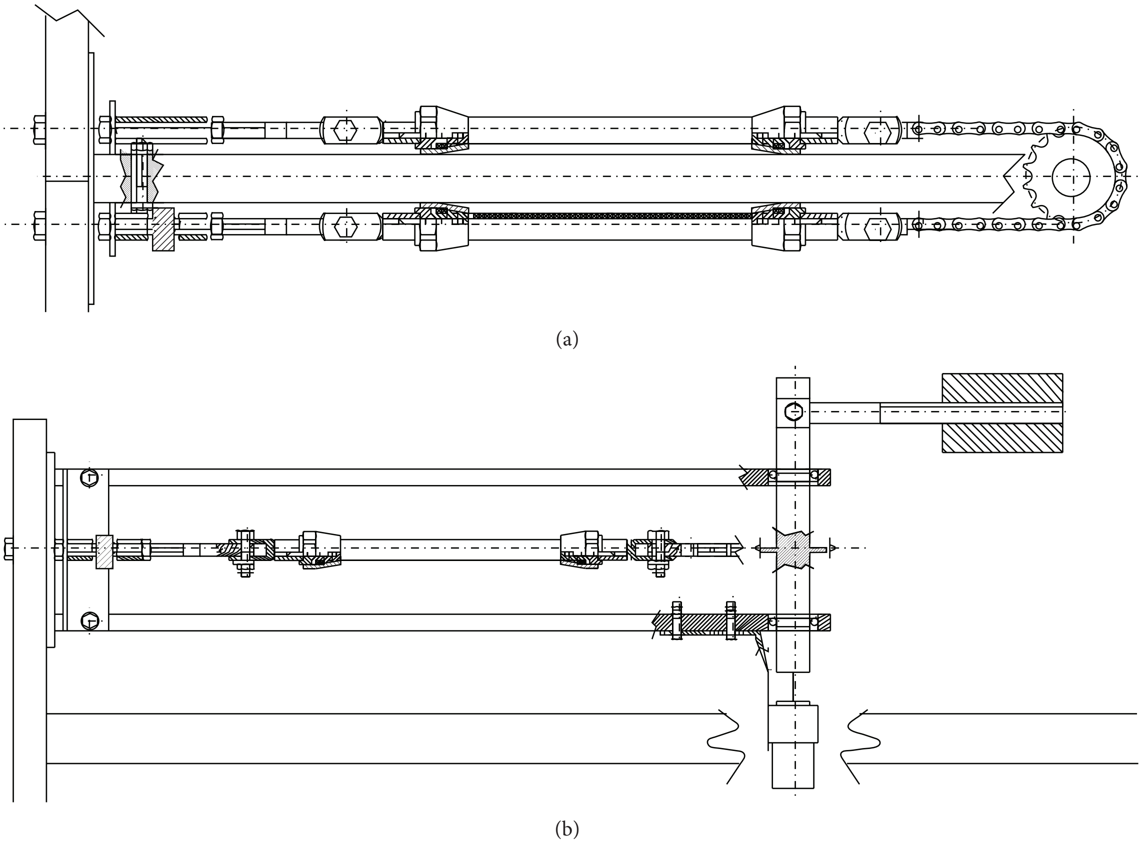
## 2. Materials and Methods

The main object of interest was pneumatic muscle-based actuator with a pair of pneumatic artificial muscles in antagonistic connection joined with a chain led through a sprocket (Figure 1). The muscles themselves were fixed to a metal plate forming a rigid base at the upper part of the actuator. The sprocket was welded to a steel shaft (visible in the side view of the actuator in Figure 1), which passed through supporting columns with bearings. A rotating arm was attached to one end of the shaft to which a load weight (representing varying moment of inertia) could be fixed.

From physical point of view, PAM-based actuator is a multidomain system consisting of electrical, pneumatic, and mechanical part (Figure 2). The electrical part of the system (denoted by EPAM, electropneumatic action module) represents the power module needed for valve control and contains four power transistors (BUZ11) together with a power supply. As the effect of this part on the relevant dynamic characteristics of the system was considered negligible, it was not included in modeling process.

By neglecting the electrical part, the system model could be divided into two parts: pneumatic part represented by a compressor and two 3/3 on-off valves and mechanical part represented by two parallel connections of nonlinear spring and nonlinear damper and rotational joint. In Figure 2, all the variables between which the dependencies expressed in the form of differential equations had to be determined as a part of modeling process are shown. The interactions among these variables characterize the physical phenomena taking place within the system during its operation, from the inflow of pressurized air into the muscles to the transfer of force action of the muscles to a load. The principle of operation of the examined actuator, which is important for developing its dynamic model, can be clarified using Figure 3.

The operation of actuator always starts from the set-up position in which both muscles are inflated to the same pressure (550 kPa in our experiments) and are at the same initial length denoted by $l_0$. This corresponds to position 1 in Figure 3, where equilibrium with the highest force occurs. Starting from this point, only one of the muscles is active (i.e., being deflated) in one half of the joint movement range and the muscles reverse their roles with arm passing through the zero position. The joint movement is stopped whenever the force equilibrium is found (corresponding to consecutive equilibrium points $1, 2, 3, \ldots$), which happens at different length of the muscles (longer for active muscle, up to nominal length) and different pressures (lower for active muscle, down to atmospheric pressure, when it is completely deflated). It is of note that the actuator operates well below the upper force limit for both muscles (1200 N for muscles with force compensator), due to a relatively large clearance between the



(a)

(b)

Figure 1: Mechanical configuration of pneumatic muscle actuator with two muscles in antagonistic connection joined through a chain gear ((a) front view and (b) side view).

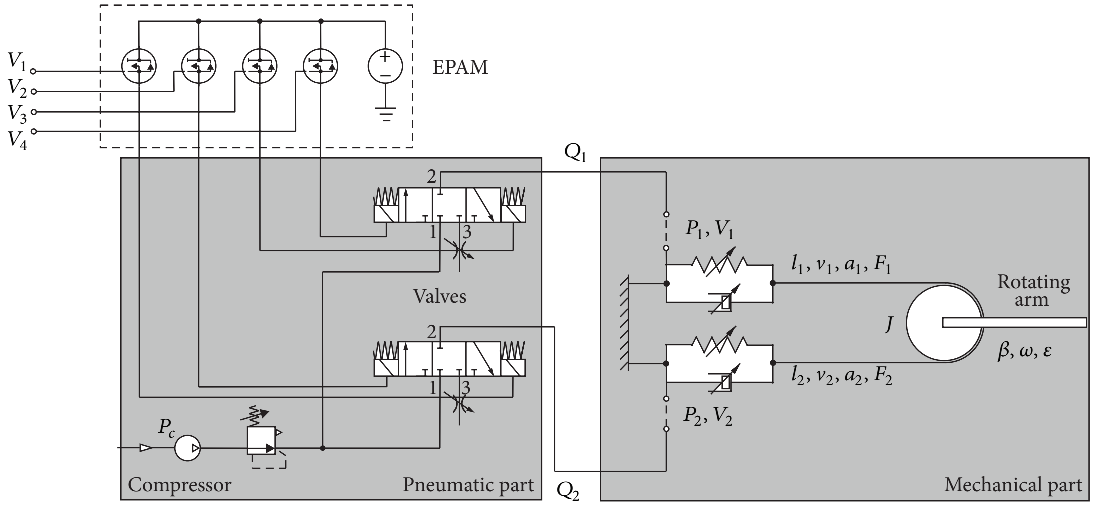


Figure 2: Physical representation of PAM-based actuator with modeled parts being shown in grey color (in the scheme, $P_c$ is compressor pressure [Pa], $Q_i$ is volume flow rate [$\text{m}^3 \cdot \text{s}^{-1}$], $P_i$ is muscle pressure [Pa], $V_i$ is muscle volume [$\text{m}^3$], $l_i$ is muscle length [m], $v_i$ is muscle velocity [$\text{m} \cdot \text{s}^{-1}$], $a_i$ is muscle acceleration [$\text{m} \cdot \text{s}^{-2}$], $i$ is muscle index, $\beta$ is joint angle [deg], $\omega$ is joint angular velocity [$\text{rad} \cdot \text{s}^{-1}$], and $\varepsilon$ is joint angular acceleration [$\text{rad} \cdot \text{s}^{-2}$]).

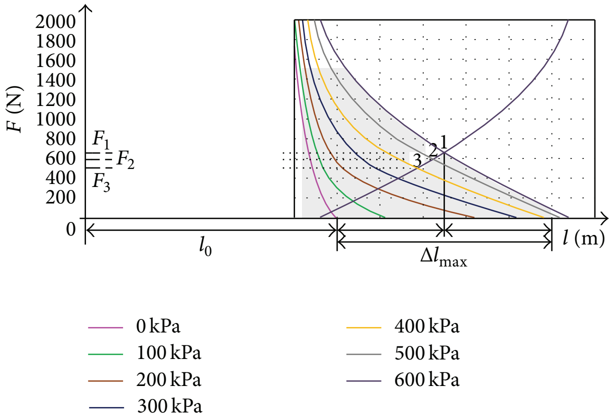


Figure 3: Static characteristics (force-length relationship) of PAM-based actuator. Equilibrium points are located at the intersections of force curves for given muscle pressures. As is shown in the figure, one of the muscles remains inflated to the initial pressure and is not active in one half of the operating range of rotational joint movement (in the figure, $F$ is muscle force, $l$ is muscle length, $l_0$ is initial muscle length, $\Delta l_{\max}$ is maximum range of muscle length change, and $P$ is muscle pressure).

chain and the sprocket in deflated state of both muscles (giving initial contraction of approximately 14%).

*2.1. Derivation of Analytical Model of Pneumatic Muscle Actuator.* As was mentioned above, a dynamic model of pneumatic muscle actuator consists of mechanical and pneumatic part. The former one is based on the mechanical model of pneumatic artificial muscle, represented as a parallel connection of nonlinear spring and nonlinear damper [6]. According to this model, PAM can be modeled as a nonlinear mass-spring-damper system described by the following nonlinear differential equation:

$$\ddot{\xi} = \left(\frac{1}{m}\right)\left[F_E - F_{nd}(\dot{\chi}, P) - F_{ns}(\chi, P)\right], \tag{1}$$

where $\xi$ is muscle displacement [m], $m$ is moved mass [kg], $F_E$ is external force [N], $F_{nd}$ is nonlinear damper force [N], $F_{ns}$ is nonlinear spring force [N], $\chi$ is muscle contraction [-], and $P$ is absolute muscle pressure [Pa].

Nonlinear spring force term ($F_{ns}$) depends on both, muscle contraction and muscle pressure, and its approximate shape can be estimated from Figure 3. This term was approximated using the fifth-order polynomial (using Surface Fitting Tool in Matlab) based on the data read from the force-length relationship given in [18] (shown in Figure 4).

The approximation equation could be expressed in the following form:

$$F_{ns}(P, \chi) = \sum_{k=0}^{l} \sum_{m=0}^{n} a_{km} P^k \chi^m, \tag{2}$$

where $a_{km}$ is polynomial coefficients. Totally 329 points (blue dots in Figure 4), taken from the experimental force-length curves, were used for the function approximation with the values of polynomial coefficients shown in Table 1. The resulting RMSE obtained for the spring force predictor using (2) was 44.416 and coefficient of determination ($R^2$) was 0.9912.

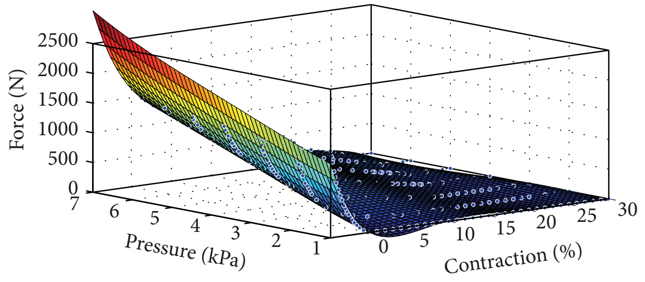


Figure 4: Muscle force function approximation using fifth-order polynomial. Blue circles correspond to experimental data given in [18]. Even though it is not shown in the figure, the force approximator used saturation function at its output to limit the force into (0, 1200) N interval.

Table 1: Values of polynomial coefficients used for approximation of nonlinear spring term (all other coefficients were equal to zero).

| Coefficient | Value |
|---|---|
| $a_{00}$ | −90.84 |
| $a_{10}$ | −79.63 |
| $a_{01}$ | 6.508 |
| $a_{20}$ | 23.5 |
| $a_{11}$ | −23.1 |
| $a_{02}$ | 115 |
| $a_{30}$ | −1.883 |
| $a_{21}$ | −0.7284 |
| $a_{12}$ | 3.715 |
| $a_{03}$ | −22.82 |
| $a_{40}$ | 0.062 |
| $a_{31}$ | 0.081 |
| $a_{22}$ | −0.145 |
| $a_{13}$ | −0.185 |
| $a_{04}$ | 2.088 |
| $a_{50}$ | $-73.5e-5$ |
| $a_{41}$ | $-96.8e-5$ |
| $a_{32}$ | $-31.1e-4$ |
| $a_{23}$ | $19.9e-3$ |
| $a_{14}$ | $-17.4e-3$ |
| $a_{05}$ | $-67.2e-3$ |

The second term in brackets on the right side of (2) corresponds to the force of (nonlinear) damper. This term is commonly proportional to the velocity of mass-spring-damper system through the damping coefficient. In our model we use a nonlinear damper, the force of which depends not only on the velocity but also on muscle pressure (as was presented in [6]). This can be expressed as follows:

$$F_{nd}(P, \dot{\chi}) = -\gamma P \dot{\chi}, \tag{3}$$

where $\gamma$ is modified damping coefficient [m·s], $P$ is muscle pressure [Pa], and $\dot{\chi}$ is muscle velocity [m·s$^{-1}$]. Due to the antagonistic connection of the muscles, one muscle can be considered an agonist and the other one an antagonist, which helps to explain the role of external force term in (2) ($F_E$) as being equal to the force of opposing muscle.

Derivation of the pneumatic part of the model was based on a simplified description of the dynamics of fluid flow within the system. The differential equation describing the dynamics of muscle pressure can be derived from Boyle-Mariotte's ideal gas law [6, 19]:

$$P_0 V_0 = P V_m, \tag{4}$$

$$\frac{d}{dt}\left(\frac{P_0 V_0}{V_m}\right) = P_0\left(\frac{\dot{V}_0 V_m - \dot{V}_m V_0}{V_m^2}\right) = P_0 \frac{\dot{V}_0}{V_m} - P_m \frac{\dot{V}_m}{V_m}, \tag{5}$$

where $P_0$ is atmospheric pressure [Pa], $V_0$ is volume of pressurized air within the muscle [m$^3$], $V_m$ is muscle volume [m$^3$], and $P$ is muscle pressure [Pa]. In order to use (5) in actuator model, one has to define the terms $V_m$, $\dot{V}_m$, and $\dot{V}_0$. The last term actually represents the volume flow of pressurized air through the valve, which can be modeled using a standard equation given, for example, in [19]:

$$\dot{V}_0 = \begin{cases} P_1 C \sqrt{\dfrac{T_0}{T_1}} \sqrt{1 - \left(\dfrac{P_2/P_1 - \psi}{1-\psi}\right)^2}, & \text{if } \dfrac{P_2}{P_1} > \psi \\ P_1 C \sqrt{\dfrac{T_0}{T_1}}, & \text{if } \dfrac{P_2}{P_1} \leq \psi, \end{cases} \tag{6}$$

where $P_1$ is absolute upstream pressure [Pa], $C$ is sonic conductance [m$^3$·s$^{-1}$·Pa$^{-1}$], $P_2$ is absolute downstream pressure [Pa], $T_0$ is ambient air temperature at reference conditions [K], $T_1$ is upstream pressure [K], and $\psi$ is critical ratio [-].

The muscle volume generally depends on geometrical parameters of the muscle, but it can be approximated using a polynomial in the following form:

$$V_m(\chi) = \sum_{k=0}^{m} b_k \chi^k, \tag{7}$$

where $b_k$ is $k$th coefficient of the approximation polynomial. In our case $m = 3$ (third-order polynomial approximation) with the following values of polynomial coefficients: $b_3 = -11 \times 10^{-4}$, $b_2 = -18 \times 10^{-6}$, $b_1 = 48 \times 10^{-5}$, and $b_0 = 79 \times 10^{-6}$. After taking time derivative of (7) for $m = 3$, we get

$$\dot{V}_m(\chi, \dot{\chi}) = 3 b_3 \chi^2 \dot{\chi} + 2 b_2 \chi \dot{\chi} + b_1 \dot{\chi}. \tag{8}$$

Up to this point, only the model of one pneumatic artificial muscle has been derived. The actuator itself uses two pneumatic muscles working as an opposing pair. From the kinematic point of view, actuator consists of one degree of freedom in the form of a rotational joint and it was therefore necessary to develop the model of joint dynamics driven by a pair of muscles described previously. Of note is also the fact that the actuator frame was placed in horizontal direction, allowing for a simpler model with neglected effect of gravity.

To develop the joint dynamics model, Lagrangian mechanics approach can be used. Hence, the following equation holds [20]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i, \tag{9}$$

where $L$ is Lagrangian and $L = K_E - P_E$, $K_E$ is kinetic energy, $P_E$ is potential energy, $q_i$ is $i$th generalized variable, and $\tau_i$ is $i$th generalized force. In case we are dealing with a rotational joint, generalized variable corresponds to a joint angle ($\beta$) and generalized force corresponds to a torque ($T$) developed by a joint actuator (in our case a pair of muscles). Thus, we get

$$q = [\beta], \quad \dot{q} = [\dot{\beta}], \quad \tau = [T], \quad K = \frac{1}{2} J \dot{\beta}^2. \tag{10}$$

Having the actuator placed in horizontal direction resulted in the fact that potential energy term in Lagrangian could be set to zero because the reference point can be chosen arbitrarily and only changes in $P_E$ are relevant [21]. Taking this fact into account, we can write the following:

$$\frac{d}{dt}\left(\frac{\partial\left((1/2) J \dot{\beta}^2\right)}{\partial \dot{\beta}}\right) - \frac{\partial\left((1/2) J \dot{\beta}^2\right)}{\partial \beta} = T \tag{11}$$

$$J \ddot{\beta} = T - T_L, \quad \text{where } T = (F_1 - F_2) r, \tag{12}$$

where $F_1, F_2$ are muscle forces [N], $r$ is sprocket diameter [m], and $T_L$ is load torque [N·m]. It is also of note that the friction term in (12) was neglected.

*2.2. Calculation of Moment of Inertia for Given Loads.* As was mentioned in the introduction, the effect of varying moment of inertia was supposed to be taken into account when designing a controller for actuator. Different moment of inertia was represented by a detachable weight that could be fixed to the actuator arm. The moment of inertia for the load weights was calculated using the following formula for compound objects:

$$J_{\text{tot}} = \sum_{k=1}^{m} J_k, \tag{13}$$

where $J_{\text{tot}}$ is total moment of inertia and $J_k$ is moment of inertia for $k$th object. This formula can be used on the condition that all particular objects have the same rotational axis. In our case, the compound object consists of a steel shaft with length $L_S$, diameter $D_S$, and mass $M_S$ and a steel arm with length $L_A$, diameter $D_A$, and mass $M_A$ and both objects rotate around the shaft's longitudinal axis. All three particular objects (shaft, arm, and load weight) were treated as solid cylinders for which moments of inertia can be calculated using standard formulas (as in, e.g., [22]):

$$J_S = \frac{1}{2} M_S R_S^2 \qquad J_A = \frac{1}{4} M_A R_A^2 + \frac{1}{3} M_A L_A^2, \tag{14}$$

where $J_S$ is shaft moment of inertia [kg·m$^2$], $M_S$ is shaft mass [kg], $R_S$ is shaft radius [m], $J_A$ is arm moment of inertia [kg·m$^2$], $R_A$ is arm radius [m], and $L_A$ is arm length [m].

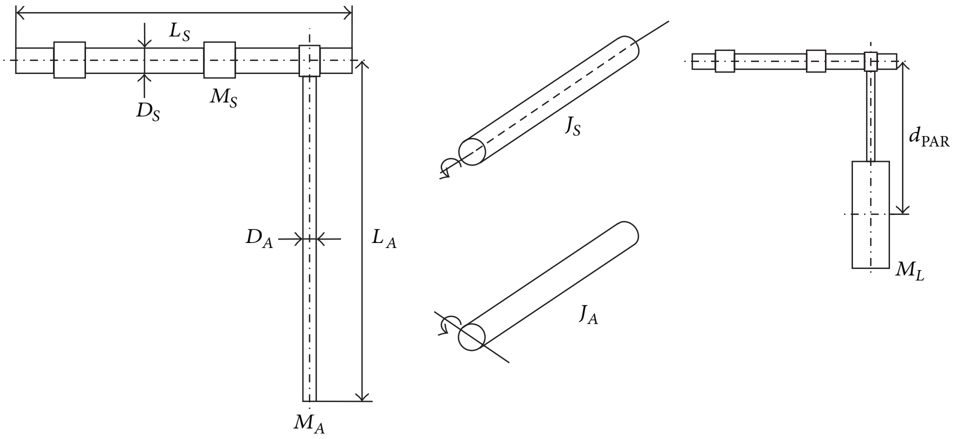


Figure 5: Arm and shaft dimensions for moment of inertia calculations (in the scheme, $L_S$ is shaft length [m], $D_S$ is shaft diameter [m], $M_S$ is shaft mass [kg], $L_A$ is arm length [m], $D_A$ is arm diameter [m], $M_A$ is arm mass [kg], $J_S$ is shaft moment of inertia [kg·m$^2$], $J_A$ is arm moment of inertia [kg·m$^2$], $M_L$ is load weight [kg], and $d_{\text{PAR}}$ is parallel axis distance [m]).

Table 2: Mechanical parameters of actuator used for moment of inertia calculation. In the table, $\rho$ is steel density, $J_n$ is calculated moment of inertia for no load, and $M_L$ is load weight (all other variables as described in Figure 5).

| Parameter | Value | Unit |
|---|---|---|
| $\rho$ | 7850 | kg·m$^{-3}$ |
| $D_S$ | 37.5 | mm |
| $L_S$ | 285 | mm |
| $M_S$ | 2.47 | kg |
| $D_A$ | 18.13 | mm |
| $L_A$ | 272 | mm |
| $M_A$ | 0.55 | kg |
| $d_{\text{PAR}}$ | 201.1 | mm |
| $M_L$ | 3.34 | kg |
| $J_n$ | 0.014 | kg·m$^2$ |
| $J_1$ | 0.155 | kg·m$^2$ |

To calculate the moment of inertia of the actuator with the load weight attached, the parallel axis theorem in the following form was used:

$$J_{\text{PAR}} = J + Md_{\text{PAR}}^2, \quad (15)$$

where $J$ is moment of inertia calculated for rotation axis passing through the centre of gravity and $d_{\text{PAR}}$ is parallel distance between the actual rotation axis and rotation axis passing through the centre of gravity. All dimensions relevant to the calculations of moment of inertia (together with the illustration of a parallel axis distance for the load weight) are shown in Figure 5. Values of given parameters and calculated moment of inertia can be found in Table 2.

*2.3. Elman Network for Unmodeled Dynamics Identification.* The model presented above neglects the description of a number of effects associated with complex phenomena that occur in the system. This resulted in deviations of the model responses from the actuator responses, whether in steady-state parts (more significant for nominal moment of inertia) or transient parts (more significant for increased moment of inertia). To improve the model performance, it is proposed here to use the powerful approximation capabilities of neural networks. The network itself should act as a disturbance term identifier, where disturbance term incorporates all the effects of unmodeled dynamics that cause the deviation of model performance from the performance of the actuator. To put it mathematically, one can resort to the expression of basic equation for the dynamics of $n$-link serial manipulator [23]:

$$\begin{gathered}\sum_{j=1}^{n} D_{ij}(q)\ddot{q}_j + \sum_{k=1}^{n}\sum_{m=1}^{n} H_{ikm}\dot{q}_k\dot{q}_m + G_i = Q_i, \\ H_{ijk} = \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial D_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial D_{jk}}{\partial q_i}\right),\end{gathered} \quad (16)$$

where $q$ is generalized variable, $D_{ij}$ is elements of inertial matrix, $H_{ikm}$ is velocity coupling vector, $G_i$ is gravity vector, and $Q_i$ is generalized force vector. This equation can be rewritten in the more compact vector form as follows:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{Q}. \quad (17)$$

According to [21], this equation can be extended with the disturbance term denoted by $\tau_P$ which includes, for instance, the effects of unmodeled dynamics. Thus, the extended form of (17) can be expressed as follows:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_{\mathbf{P}} = \mathbf{Q}, \quad (18)$$

where $\mathbf{F}(\dot{\mathbf{q}})$ is friction term and $\boldsymbol{\tau}_{\mathbf{P}}$ is disturbance term. The basic principle behind the use of neural network for disturbance term identification is shown in Figure 6. The scheme depicted in the figure consists of the actuator in open-loop mode, the analytical model of the actuator, and a neural network. The difference between measured actuator output

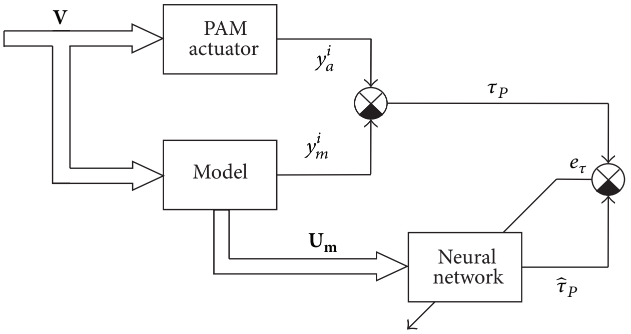


FIGURE 6: Proposed identification scheme for the estimation of disturbance term $\tau_P$ using a neural network. In the scheme, $\mathbf{V}$ is valve control signal vector, $y_a^i$ is $i$th actuator output variable, $y_m^i$ is $i$th model output variable, $\tau_P$ is disturbance term, $\hat{\tau}_P$ is disturbance term estimation, $e_\tau$ is disturbance term estimation error, and $\mathbf{U_m}$ is neural network input vector.

variable ($y_a$) and model output variable ($y_m$) corresponds to the magnitude of model error caused by unknown disturbance effects, expressed in the form of disturbance term ($\tau_P$—we drop the vector denomination in bold because we treat this variable as scalar). The main role of the neural network was to estimate this term based on the information from the model fed as an input vector $\mathbf{U_m}$. This estimation is denoted by $\hat{\tau}_P$ in the scheme, and the difference between the measured disturbance term and its estimation ($e_\tau$) was used as an error signal, which should have been minimized by the network learning algorithm.

It was decided to prefer recurrent neural network over the purely feedforward type due to its better long-term memory capabilities [24]. These capabilities mainly stem from its internal memory representation through the delay feedback from the hidden layer back to the input (Elman neural network, Figure 7(b)). The input into the context layer is actually one-step prediction of the state vector $\mathbf{x}(k+1)$, which is further processed by the output linear layer to obtain one-step output vector prediction $\mathbf{y}(k+1)$ (Figure 7(a)). To express the relevant relationships mathematically, one can first define state and output equations for Elman network:

$$\begin{gathered}\mathbf{x}(k+1)=\mathbf{f}\left(\mathbf{CWx}(k)+\mathbf{IWu}(k)+\mathbf{b}^{(1)}\right)\\ y(k)=\mathbf{LWx}(k)+b^{(2)},\end{gathered}\tag{19}$$

where $\mathbf{x}(k)$ is $n$-dimensional state vector, $\mathbf{u}(k)$ is $m$-dimensional input vector, $\mathbf{b}^{(1)}$ is hidden layer bias vector, $b^{(2)}$ is output layer bias (only one neuron was used in the output layer), $\mathbf{CW}$ is context layer weight matrix, $\mathbf{LW}$ is hidden layer weight matrix, $\mathbf{IW}$ is input layer weight matrix, $\mathbf{f}$ is nonlinear activation function vector in hidden layer, and $k$ is $k$th sample.

All the hidden layer neurons used the same activation function (hyperbolic tangent) in the following form:

$$f(x)=\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}.\tag{20}$$

By using (19), we can express the disturbance term estimation using Elman network as

$$\tau_P=\mathbf{LW}f\left(\mathbf{CWx}(k)+\mathbf{IWu}(k)+\mathbf{b}^{(1)}\right)+b^{(2)}+e_\tau,\tag{21}$$

where $e_\tau$ is disturbance term estimation error.

Since we assumed offline identification task, the strategy used for network learning was epochwise training with weight updates after the presentation of the whole sample set. In order to use conventional backpropagation method for network training, it is necessary to unfold the network in time so that a new layer with updated state is added for each time step. Then, the network is trained in two passes, forward and backward, to obtain the input, state, and output data and to calculate local gradients using the following formula [25]:

$$\delta_j(k)=-\frac{\partial E_{\text{tot}}(k_0,k_1)}{\partial l_j(k)}\tag{22}$$

for all $j\in\Im$ and $k_0<k\le k_1$. In this formula, $j$ is index of neuron for which desired response is computed, $E_{\text{tot}}(k_0,k_1)$ is total error function for time period from $k_0$ to $k_1$, and $l_j(k)$ is induced local field for $j$th neuron. The local gradients are calculated backwards, starting from the last time step $k_1$ and then using this value in computation of local gradient(s) in previous time step using the following formula:

$$\delta_j(k)=\begin{cases}f'\left(l_j(k)\right)e_j(k) & \text{for } k=k_1\\ f'\left(l_j(k)\right)\left[e_j(k)+\sum\limits_{j\in\Im}w_{jm}\delta_k\right.\\ \qquad\qquad\qquad\left.\cdot(k+1)\right] & \text{for } k_0<k<k_1,\end{cases}\tag{23}$$

where $f'(l_j(k))$ is derivation of activation function of $j$th neuron and $w_{jk}$ is weight between $j$th and $m$th neuron. The weights are then updated as follows:

$$\Delta w_{ji}=-\eta\frac{\partial E_{\text{tot}}(k_0,k_1)}{\partial w_{ji}}=\eta\sum_{k=k_0+1}^{k_1}\delta_j(k)\,x_i(k-1),\tag{24}$$

where $\eta$ is learning rate and $x_i(k-1)$ is input applied to $i$th synapse of $j$th neuron in time step $k$.

Actual network training was carried out using the Levenberg-Marquardt algorithm with Bayesian regularization to improve the generalization properties of resulting neural model. Thus, the update of whole parameter vector $\boldsymbol{\theta}$ (containing all weights and biases) had the following form [26, 27]:

$$\begin{gathered}\boldsymbol{\theta}_{\text{new}}=\boldsymbol{\theta}_{\text{current}}-\left[\mathbf{J}^T\mathbf{J}+\mu\mathbf{I}\right]^{-1}\mathbf{J}^T\mathbf{e}\text{ where}\\ \mathbf{J}^T\mathbf{J}\approx\mathbf{H}\quad\mathbf{J}^T\mathbf{e}=\mathbf{g},\end{gathered}\tag{25}$$

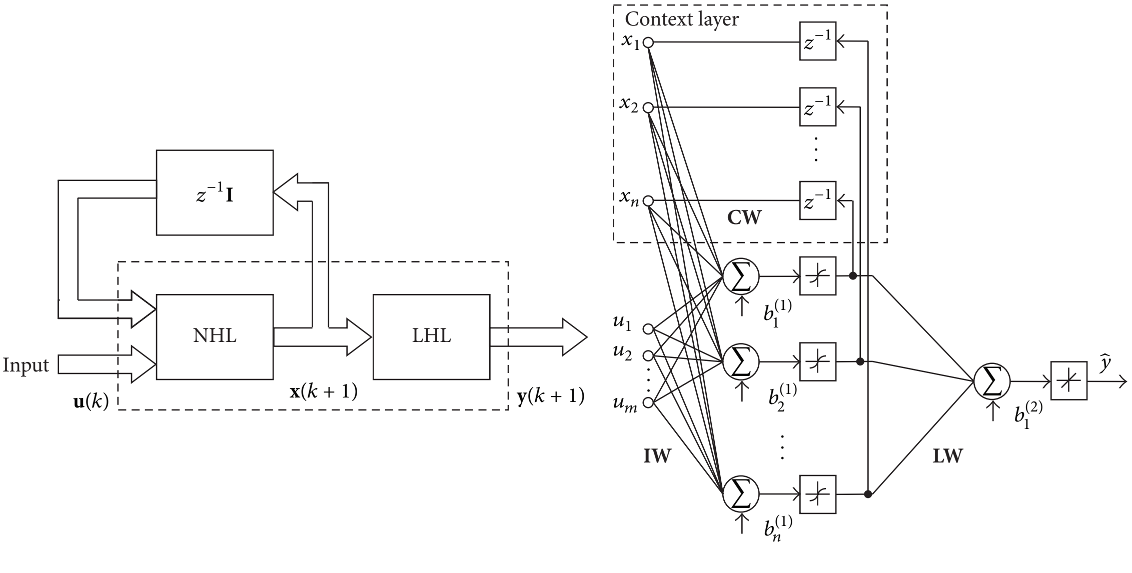


Figure 7: State diagram of Elman network (a) and network structure (b) where $\mathbf{u}(k)$ is input vector, $\mathbf{x}(k+1)$ is state vector, $\mathbf{y}(k+1)$ is output vector, NHL is nonlinear hidden layer, LHL is linear hidden layer, $z^{-1}$ is unit delay, **IW** is input layer weight matrix, **CW** is context layer weight matrix, **LW** is hidden layer weight matrix, $b_l^{(k)}$ is $l$th neuron bias in $k$th layer, and $\hat{y}$ is network output.

where $\mathbf{J}$ is Jacobian matrix for network error with respect to $\boldsymbol{\theta}$, $\mathbf{I}$ is identity matrix, $\mathbf{H}$ is Hessian matrix, $\mathbf{g}$ is gradient vector, and

$$\mathbf{e} = e_i(\boldsymbol{\theta}) = t_i - y_i \quad i = 1, \ldots, n \tag{26}$$

is error vector containing error values for all $n$ samples in data set. The parameter $\mu$ in (25) allows altering the direction of movement between the steepest descent direction and Newton direction and its value is modified in every step according to the success of this step with regard to the change in error function value. Its limit value was set to $\mu = 1e^{10}$ and breaching this limit was the main criterion for network training convergence.

## 3. Results and Discussion

In the first step, the responses of the model derived in Section 2.1 were compared to actuator responses as a part of model validation process. The variables used in this process were selected based on the availability of sensors for their measurement. These included joint angle $\beta$ [deg], left muscle force $F_1$ [N], right muscle force $F_2$ [N], left muscle pressure $P_1$ [kPa], and right muscle pressure $P_2$ [kPa]. The muscles are denoted by left and right when viewed from front in upright position of the actuator. To estimate how well the model response matches the response of the actuator, mean squared error (MSE) criterion was used in the following form:

$$\mathrm{MSE} = \frac{1}{n} \sum_{k=1}^{n} \left(y_k - \hat{y}_k\right)^2, \tag{27}$$

where $n$ is number of samples, $y_k$ is actuator variable in $k$th time step, and $\hat{y}_k$ is model variable in $k$th time step. In order to decrease the level of noise in analog signals from pressure and force transducers, a simple moving average (MA) method was used:

$$\mathrm{MA} = \frac{\sum_{i=1}^{m} y_i}{m}, \tag{28}$$

where $y_i$ is $i$th component of filtered signal and $m$ is time window for filtering ($m = 21$).

To compare how well the model matches the responses of the actuator, excitation signal shown in Figure 8 was used. This signal contained several randomly chosen valve control pulses, which corresponded to random changes of joint angle within its operating range. It can be observed that the pulse sequences are paired (thus forming two sets of distinctive sequences) in order to ensure smooth transition of the joint angle through the reference (zero) position. Since it was considered important to take into account the effect of changing moment of inertia as a representation of varying system parameter, the test was also carried out for 11-multiple of nominal moment of inertia calculated in Section 2.2 ($J = 11J_n$). Initial (relative) muscle pressure was selected to be $p_0 = 550$ kPa at which the muscles contracted by approximately 14% (these values were also set in the model for simulation). The results of both tests, expressed in the quantitative form, are shown in Table 3. In this table, the mean squared error for given test responses calculated using (27) is shown.

Qualitatively, the results can be evaluated using Figure 9 ($J = J_n$) and Figure 10 ($J = 11J_n$), where the comparison of responses between the model and the actuator for all variables can be seen. If one inspects all the responses, several aspects can be observed in relation to the performance of both, model and actuator. At first, the model, in general, captures the salient points of the actuator's dynamics in both

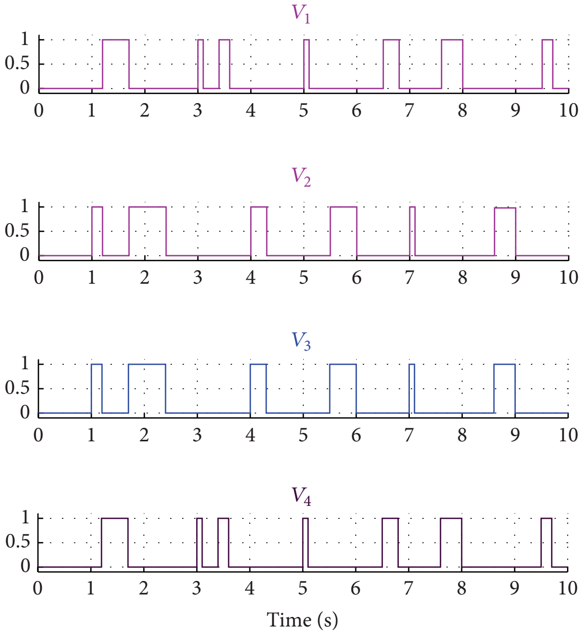


Figure 8: Excitation signal used in model validation. The pulses shown in figure correspond to logic signals used for actuator control (0 is closed valve, 1 is open valve, and $V_1$, $V_2$, $V_3$, and $V_4$ are valves).

cases. When the joint angle for nominal case is considered (Figure 9), it can be seen that up to time of 2 seconds the responses are quite similar while from that time on, the model predicts higher values of joint angle. This is more pronounced in steady-state part of the response with the maximum difference of 6.78° at 6.5 seconds. The situation is similar for all other variables (muscle forces and muscle pressures), where the first part of the model response matches the actuator response quite well with subsequent accumulation of the prediction error. This should not come as a surprise, as the dynamics of the joint angle depends on the torque being determined by the difference in muscle forces, and muscle forces themselves depend on the muscle pressure (as defined in Section 2.1). The deviations of the model responses from the actuator responses expressed through MSE in Table 3 represent the accumulation of errors caused by modeling inaccuracies and simplifications—compressor pressure drop, errors in force function approximation, errors caused by simplifications of modeling equations, and so forth. Still, the responses of the model might offer usable insight into the dynamics of the actuator.

In Figure 10, the responses for 11-multiple of nominal moment of inertia are shown. It is readily observable that, in contrast to the responses with nominal moment of inertia, the oscillations are now much more pronounced. Similar to the previous case, after 2 seconds the model predicts higher values of joint angle with generally lower amplitude of oscillations.

The responses of muscle force in actuator are also markedly oscillatory, with the amplitude of these oscillations being on average higher than predicted by the model. On the other hand, muscle pressure responses in actuator do not show significant oscillations; however, it has to be noted that the pressure sensors were placed approximately 1 meter down the pressure line from the muscles and this could have caused more dampened responses. Quantitative evaluation of the results for $J = 11J_n$ shows that the relative errors for this case are lower than for $J = J_n$, which can be explained by the more pronounced oscillations and the way in which mean absolute error is calculated.

Table 3: Results of all the tests for given moments of inertia, calculated as mean squared error between both the actuator and the model response ($\beta$ is joint angle [deg], $P_1$ is left muscle relative pressure [kPa], $P_2$ is right muscle relative pressure, $F_1$ is left muscle force [N], and $F_2$ is right muscle force [N]).

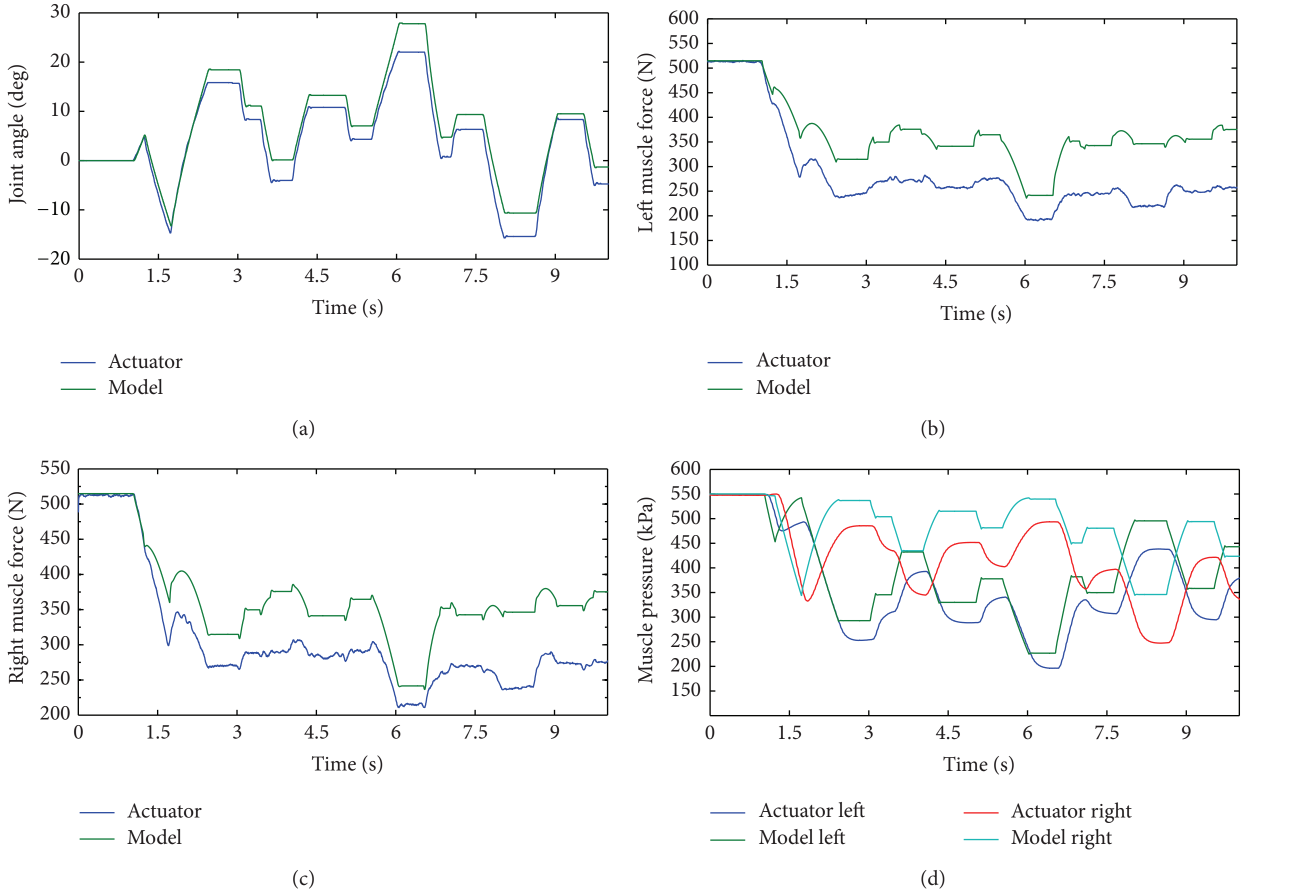
| Variable | MSE<br>$J = J_n$ | MSE<br>$J = 11J_n$ |
|---|---|---|
| $\beta$ | 10.024 | 10.2917 |
| $F_1$ | 7663.2 | 5728.6 |
| $F_2$ | 4561.8 | 3003.7 |
| $P_1$ | 2298.9 | 1807.5 |
| $P_2$ | 5160.2 | 3919.9 |

The responses shown in Figures 9 and 10 reveal the inaccuracies of derived analytical model, but the information contained in them is hardly useful for neural network training and validation due to its lack of excitation richness, which is very important in experimental modeling. In order to obtain data better suited for disturbance identification purposes, pseudobinary random sequences depicted in Figure 11 were used.

Using the signals shown in Figure 11 as input to both the actuator and the model, it was possible to obtain sufficiently representative data sets, which could be used in neural network training process. Each of the training data sets contained 80000 samples (sampling period was set to 0.003 s), and another 5000 samples not included in these sets were used as test data for checking the generalization capabilities of trained networks. The number of neurons was selected to be 15 and the maximum number of iterations was limited to 1000. It is of note that not all of the network trainings necessarily converged to local optimum within this number of iterations (the training was considered to have converged if $\mu \geq 10^{10}$).

The results of neural network training process can be evaluated using Table 4. In this table, both normalized and unnormalized values of MSE criterion are used. The reason for this distinction is the fact that, in case of the training, Bayesian regularization was used which performs best when data are normalized while in case of testing, unnormalized MSE offers better guess of the overall error in terms of operating range for a given variable. In addition to the resulting errors, the table shows also convergence status for all network trainings. Thus, one can readily see that in four cases the network training did not converge and the results probably could have been improved if the training continued.

On the other hand, the number of maximum iterations (1000) was considered sufficient to give the results close to



(a)

(b)

(c)

(d)

Figure 9: Comparison of model and actuator performance for the excitation signal shown in Figure 8 with nominal moment of inertia ((a) joint angle response, (b) left muscle force response, (c) right muscle force response, and (d) muscle pressure response for both muscles).

the optimum, and also the final values of training parameters in nonconverged trainings did not imply that large changes in resulting MSE could have been anticipated. Moreover, it should be noted that even for converged trainings the resulting MSE was very likely only a local optimum and different value could have been obtained for a training restarted with new initial weights.

The composition of $\mathbf{U}_\mathbf{m}$ vector (the number and selection of input variables) was found using trial and error approach. As can be observed from Table 4, it consists of the same input variables for given type of a variable (e.g., force or pressure), with the one exception in right muscle force ($J = 11J_n$), where an additional variable (i.e., joint angle) had to be used to obtain better enhanced model performance. One of the most important results of the presented approach can be illustrated through comparison of the resulting MSE on test data set for enhanced and nonenhanced model (given in the third and the fourth column of Table 4). These results can be evaluated also visually using Figures 12 and 13, where the responses of enhanced and nonenhanced model for $J = J_n$ and $J = 11J_n$ are shown. This evaluation can be further complemented by the inspection of error histograms obtained for the test data set using enhanced model shown in Figure 14. The addition of error histograms should help to better evaluate the contribution of neural network to improvement of the model accuracy as well as pointing to possibly problematic parts of the identification process.

It can be seen in Figure 12 ($J = J_n$) that the most significant error for joint angle prediction using nonenhanced model occurs in steady-state parts of the responses. Using the enhanced model, it was possible to reduce MSE to 1.08% of its original value. According to the histogram (Figure 14(a)), the higher number of instances (>300) occurs for errors in the range of −2° to −0.5° and most error instances are negative meaning that the model predicts very slightly higher values compared to the actuator. The performance of the enhanced model for the force prediction is comparable (Figures 12(b) and 12(c)). In these cases, the error reduction for left muscle and right muscle was to 1.52% and 1.58%, respectively. As can be seen from the figure, the nonenhanced model captures the nature of the force dynamics in both cases, but its errors are as high as 150 N in some cases. Error histograms reveal that most error instances also have negative sign, meaning that the model predicts higher values. This fact may be caused by the composition of training and test data sets and/or the results of neural network training. In case of muscle pressure prediction, the error reduction for enhanced model was to 1.9% (left muscle) and 1.82% (right muscle). The analytical model generally predicts the higher values of muscle pressures compared to the actuator. By comparing

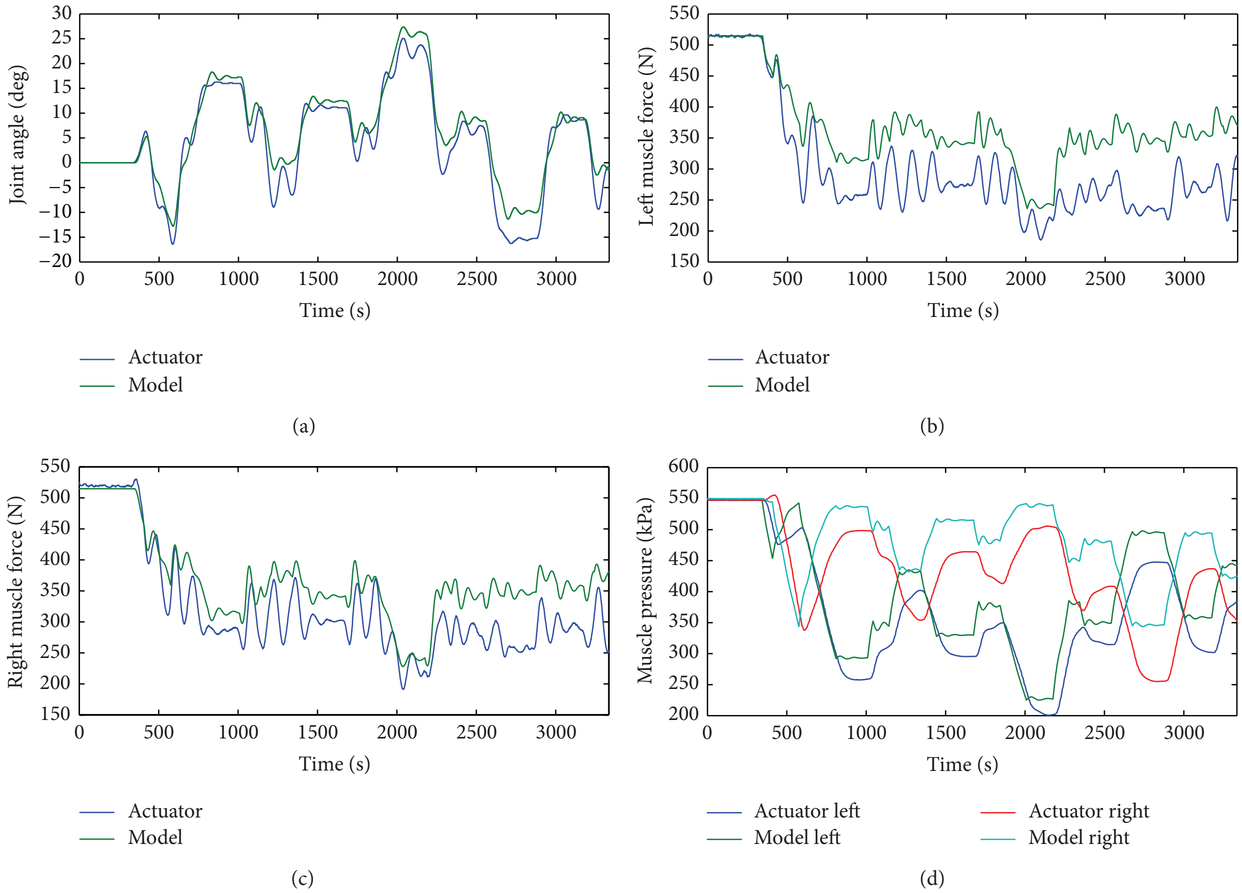


FIGURE 10: Comparison of model and actuator performance for the excitation signal shown in Figure 8 with 11-multiple of moment of inertia ((a) joint angle response, (b) left muscle force response, (c) right muscle force response, and (d) muscle pressure response in both muscles).

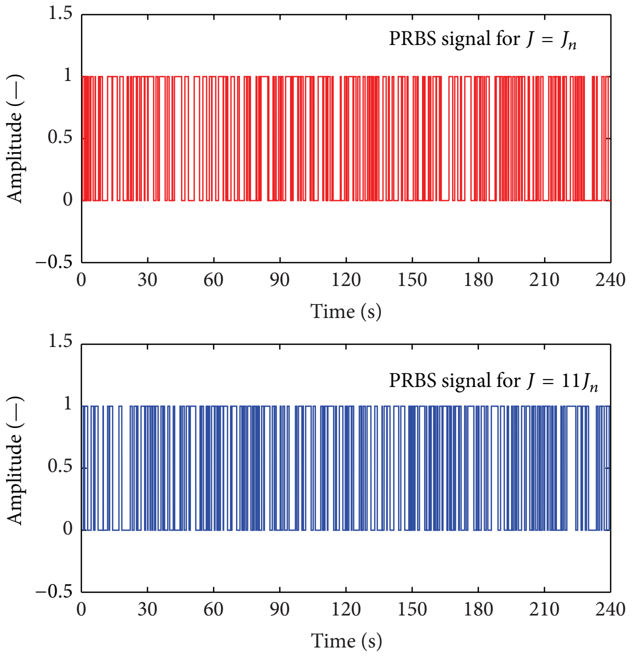


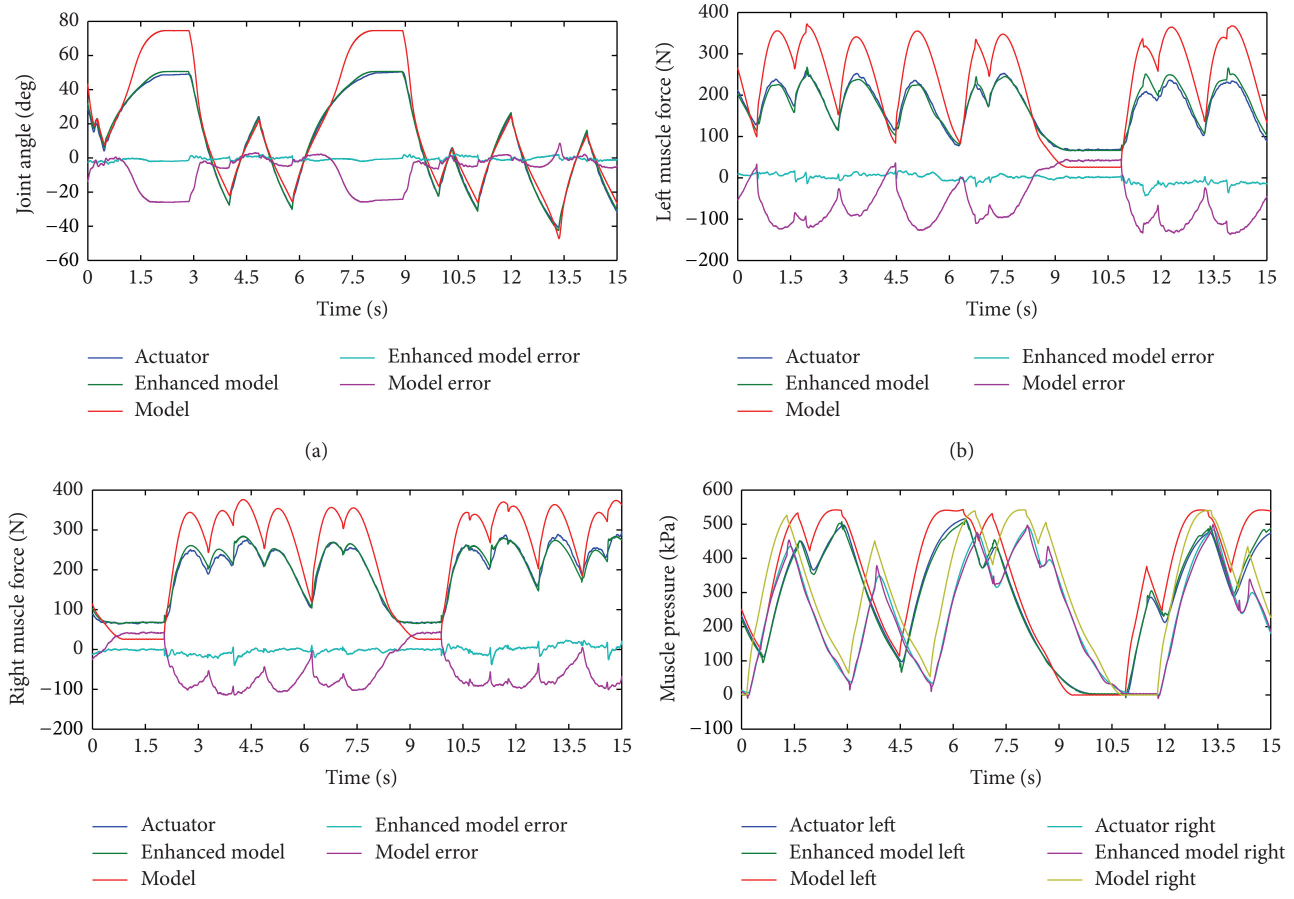
FIGURE 11: Excitation (pseudorandom binary sequence) signals used for training and test data acquisition for neural network training and validation (different signals were used for nominal and 11-multiple moments of inertia).

respective error histograms (Figures 14(d) and 14(e)), it can be concluded that most instances of the error occur in the range from −3 kPa to 12 kPa for left muscle and from −4.5 kPa to 14 kPa which is very good result given the operating range of this variable.

As was illustrated in the first part of this section, the change of nominal moment of inertia affected the actuator's dynamics in the way that was not accurately described in the original analytical model. This can be further demonstrated using Figure 13, where the responses for test excitation signal with $J = 11J_n$ are shown. In case of joint angle prediction, MSE value was reduced to 5.47% of its original value. It can be observed that the analytical model does not describe less pronounced oscillations of the transient parts of the response, which can be significant later in closed-loop tests. The remaining errors of the enhanced model are (similar to the nominal case) mostly in the range of −2.5° to −0.5°. The responses of muscle forces for 11-multiple of nominal moment of inertia probably show the most complex dynamic patterns of all measured responses. Even in this case, the networks reduced MSE to 2.54% (left muscle) and to 5.4% (right muscle). These responses reveal the intricate interplay of muscles' dynamics and structural dynamics of the actuator, which give rise to such complex dynamic patterns. When force error histograms (Figures 14(g) and 14(h)) are checked, one can

Table 4: The results of neural network training for $J = J_n$ and $J = 11J_n$. In the table, $\mathrm{MSE}_n$ is normalized mean squared error, MSE is mean squared error, EM is enhanced model, M is model, $\mathbf{U_m}$ is neural network input vector, $\beta_m$ is model joint angle, $\omega_m$ is model angular velocity, $\varepsilon_m$ is model angular acceleration, $F_{1m}$ is left model muscle force, $F_{2m}$ is right model muscle force, $P_{1m}$ is left model muscle pressure, and $P_{2m}$ is right model muscle pressure.

| Variable | $\mathrm{MSE}_n$ training | MSE testing EM | MSE testing M | Convergence | $\mathbf{U_m}$ |
|---|---|---|---|---|---|
| Joint angle, $J = J_n$ | 0.00522 | 1.607 | 148.896 | Yes (444 it.) | $\beta_m, \omega_m, \varepsilon_m$ |
| Left muscle force, $J = J_n$ | 0.004 | 105.925 | 6960.1 | No | $F_{1m}, F_{2m}, \omega_m, \varepsilon_m$ |
| Right muscle force, $J = J_n$ | 0.00688 | 83.412 | 5278.5 | Yes (164 it.) | $F_{1m}, F_{2m}, \omega_m, \varepsilon_m$ |
| Left muscle pressure, $J = J_n$ | 0.00595 | 102.757 | 5401.6 | Yes (88 it.) | $P_{1m}, P_{2m}, \omega_m, \varepsilon_m$ |
| Right muscle pressure, $J = J_n$ | 0.006 | 113.651 | 6247.5 | Yes (66 it.) | $P_{1m}, P_{2m}, \omega_m, \varepsilon_m$ |
| Joint angle, $J = 11J_n$ | 0.00237 | 4.146 | 80.721 | No | $\beta_m, \omega_m, \varepsilon_m$ |
| Left muscle force, $J = 11J_n$ | 0.00644 | 220.236 | 8679.9 | No | $F_{1m}, F_{2m}, \omega_m, \varepsilon_m$ |
| Right muscle force, $J = 11J_n$ | 0.009 | 316.135 | 5857.7 | No | $F_{1m}, F_{2m}, \omega_m, \varepsilon_m, \beta_m$ |
| Left muscle pressure, $J = 11J_n$ | 0.00292 | 100.206 | 3862.9 | Yes (535 it.) | $P_{1m}, P_{2m}, \omega_m, \varepsilon_m$ |
| Right muscle pressure, $J = 11J_n$ | 0.00391 | 123.295 | 7022.3 | Yes (449 it.) | $P_{1m}, P_{2m}, \omega_m, \varepsilon_m$ |



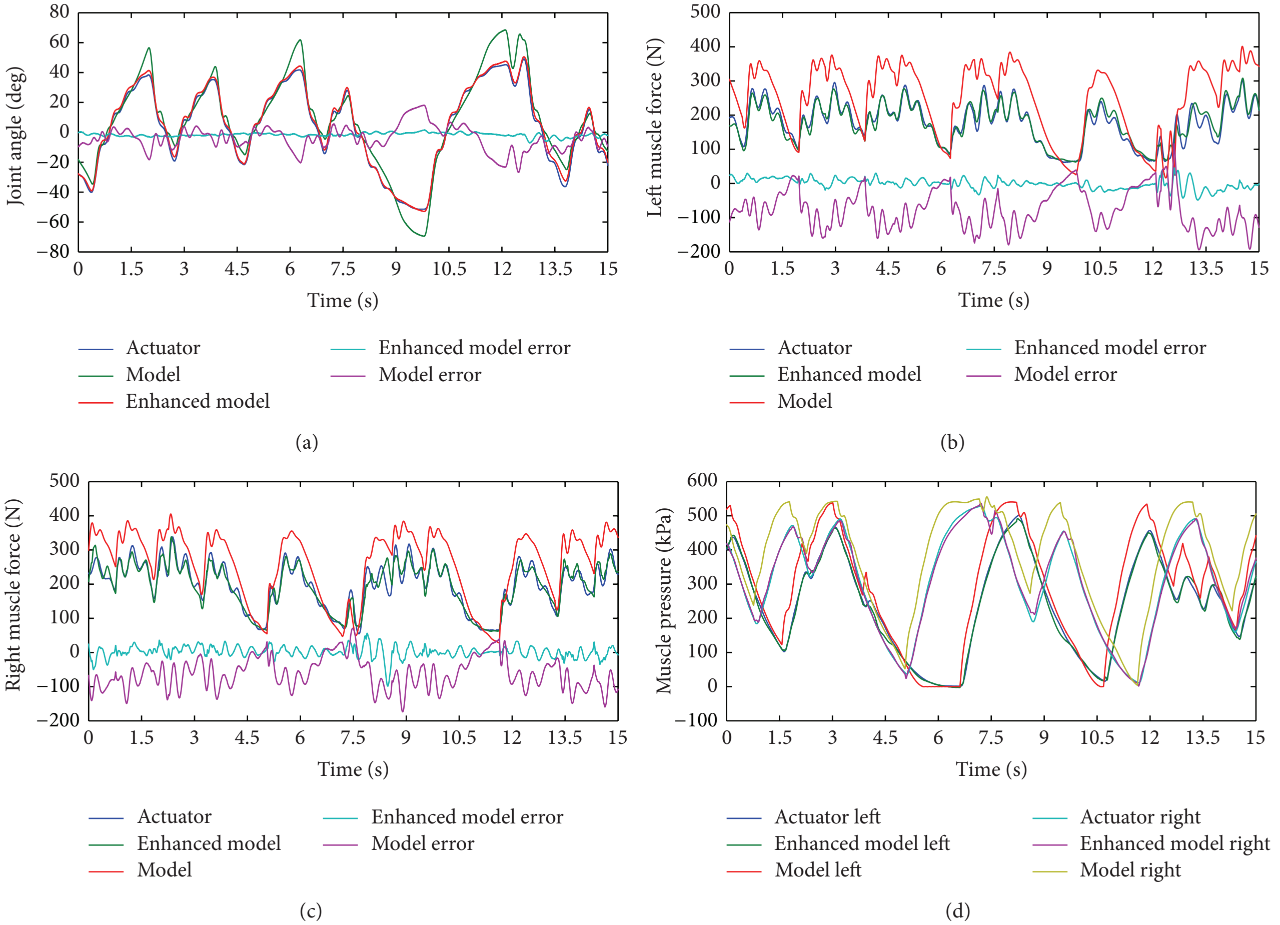
(a)

(b)

(c)

(d)

Figure 12: Comparison of model, enhanced model, and actuator generalization performance (test on unseen data with nominal of moment of inertia ((a) joint angle response, (b) left muscle force response, (c) right muscle force response, and (d) muscle pressure response in both muscles)).



(a)

(b)

(c)

(d)

Figure 13: Comparison of model, enhanced model, and actuator generalization performance (test on unseen data with 11-multiple of nominal moment of inertia ((a) joint angle response, (b) left muscle force response, (c) right muscle force response, and (d) muscle pressure response in both muscles)).

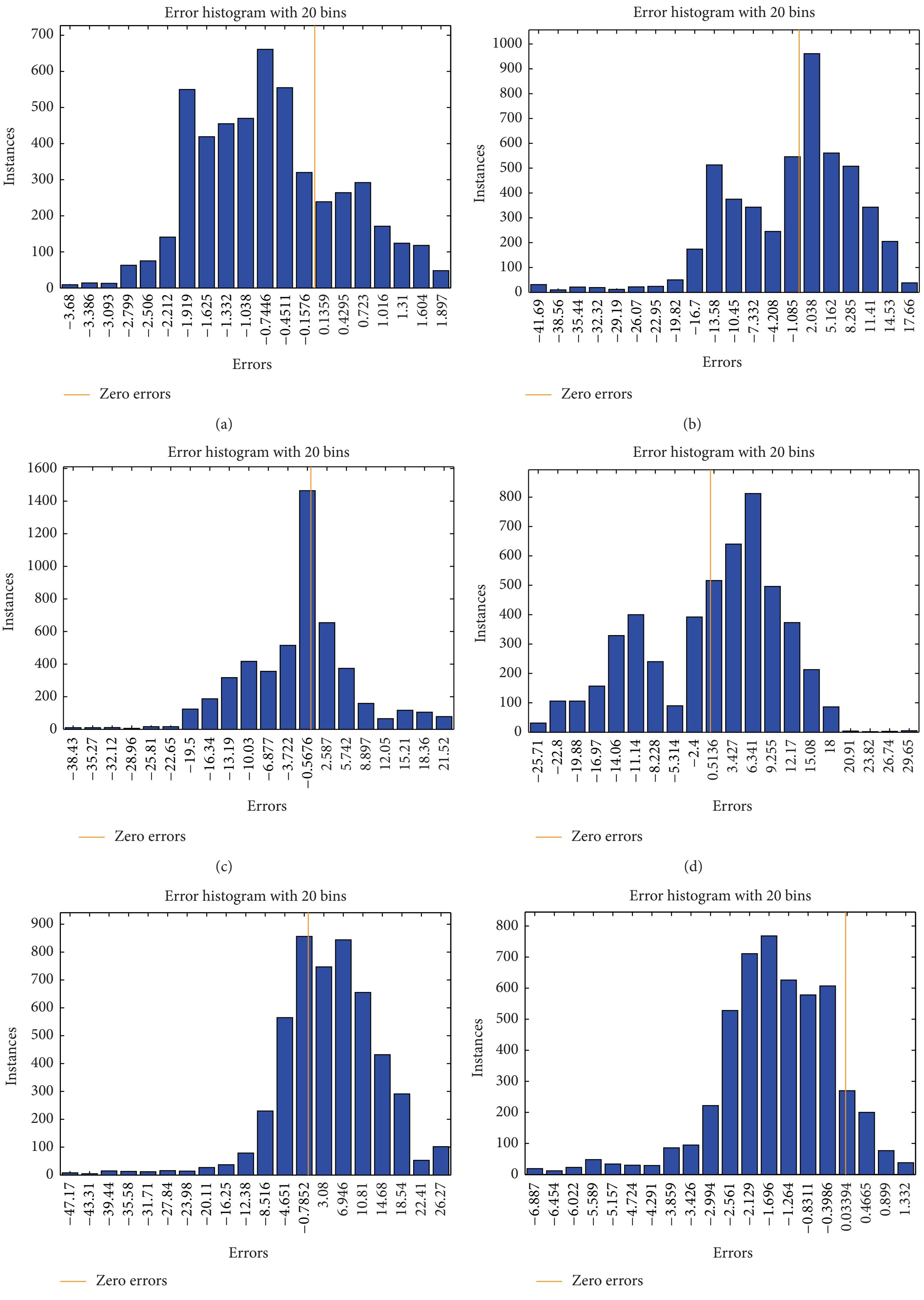
see that most error instances are distributed within a similar range (approximately 15° to both sides of the zero errors) with very low number of instances of larger errors occurring for right muscle force.

This corresponds with the need for incorporation of additional variable into $\mathbf{U_m}$ to achieve similar values of MSE for right muscle force neural network and corroborates the asymmetry of actuator's operation. The error in muscle pressure prediction was reduced to 2.59% (left muscle) and 1.76% (right muscle). This time, the oscillations are also present in muscle pressure responses and these are captured very well for both muscles. Most error instances occur in the range of −4.5 kPa to 10 kPa for left muscle and −10 kPa to 15 kPa for right muscle. The results shown in histograms support the hypothesis of complex dynamic phenomena, which take place within the system during its operation since the number of instances of enhanced model errors in larger ranges is higher compared to the nonenhanced model. Moreover, the asymmetry of error distribution for different variables as well as their different absolute ranges observed in histograms can be linked to the complexity of actuator behavior, which further justifies the need for powerful approximators like neural networks.

It is easily observed that the nonenhanced dynamic model can be used for the basic description of the actuator's operation, yet its accuracy is compromised due to a number of simplifications made during the model development. This was even more significant when the moment of inertia was changed and the effect of unmodeled dynamics was magnified. The results confirm that its combination with a recurrent neural network can be a viable solution to obtain an accurate dynamic model of the PAM-based actuator with retained physical interpretation. Since the main purpose of this model is simulation (in contrast to prediction, where an output from a real plant allows for the model output correction), recurrent neural networks seem to be a better option compared to network without internal memory.

## 4. Conclusion

In this work, an enhanced dynamic model of one-DOF pneumatic muscle actuator was developed. One of the critical points of selected approach was to retain the physical interpretation of the resulting model so that it could still offer an insight into the physics of the system. As a part of this work,



(a)

(b)

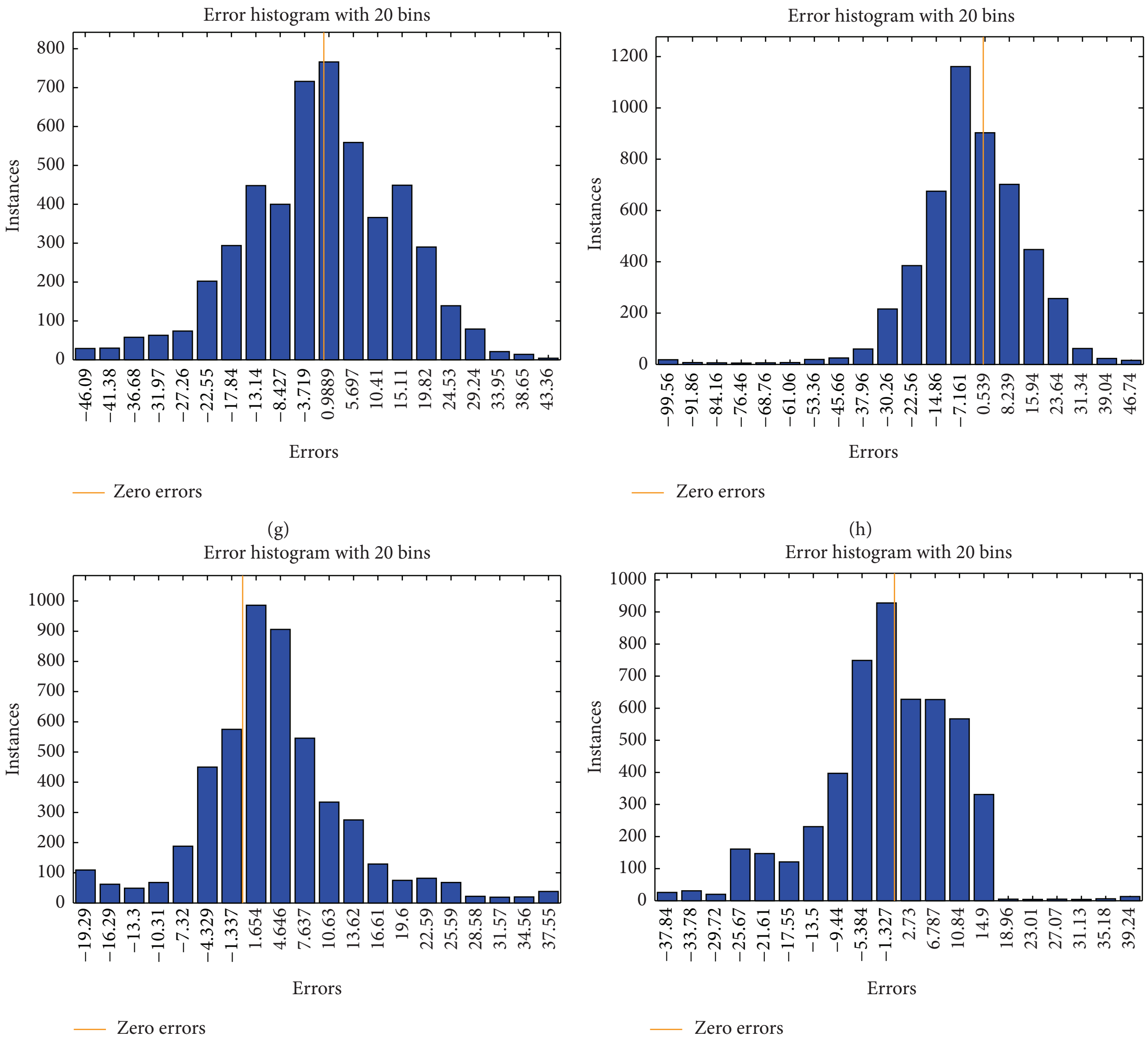
(c)

(d)

(e)

(f)

Figure 14: Continued.



(g)

(h)

(i)

(j)

Figure 14: Error histograms of enhanced model for test data set (5000 samples). In the figure, (a) is joint angle ($J = J_n$), (b) is left muscle force ($J = J_n$), (c) is right muscle force ($J = J_n$), (d) is left muscle pressure ($J = J_n$), (e) is right muscle pressure ($J = J_n$), (f) is joint angle ($J = 11J_n$), (g) is left muscle force ($J = 11J_n$), (h) is right muscle force ($J = 11J_n$), (i) is left muscle pressure ($J = 11J_n$), and (j) is right muscle pressure ($J = 11J_n$).

an analytical dynamic model of the actuator based on the nonlinear mass-spring-damper analogy was derived and verified using the responses of real actuator. Despite its capability to capture the salient points of the system's dynamics, the accuracy suffered from a number of simplifications that were made during the development of the analytical model. This was even more profound for changing the nominal moment of inertia, which was considered important for the development of a robust controller. We proposed to use a recurrent (Elman) neural network in the role of the identifier of disturbance term, which included all the effects causing the deviation of the model performance from the actuator performance. It was shown that the enhanced model offered very good improvement of the accuracy of original analytical model while keeping its physical interpretation.

Some limitations of the method can be linked to standard limitation of the conventional neural network training, like its only local optimality. It is therefore possible to concentrate on the use of global optimization methods (for instance, hybrid metaheuristic methods) in further research. In addition to this, it is also important to test the performance of this approach when used in closed-loop mode since this will directly affect the results of model-based controller design.

## Conflict of Interests

Authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

The research work is supported by the Project of the Structural Funds of the EU, Operational Programme Research and Development, Measure 2.2 Transfer of knowledge and technology from research and development into practice. Title of the project is Research and Development of Nonconventional Artificial Muscles-Based Actuators, ITMS code: 26220220103.

## References

[1] E. Kelasidi, G. Andrikopoulos, G. Nikolakopoulos, and S. Manesis, "A survey on pneumatic muscle actuators modeling," *Journal of Energy and Power Engineering*, vol. 6, no. 9, pp. 1442–1452, 2012.

[2] S. Davis, N. Tsagarakis, J. Canderle, and D. G. Caldwell, "Enhanced modelling and performance in braided pneumatic muscle actuators," *International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 213–227, 2003.

[3] S. Davis and D. G. Caldwell, "Braid effects on contractile range and friction modeling in pneumatic muscle actuators," *International Journal of Robotics Research*, vol. 25, no. 4, pp. 359–369, 2006.

[4] M. Doumit, A. Fahim, and M. Munro, "Analytical modeling and experimental validation of the braided pneumatic muscle," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1282–1291, 2009.

[5] J. L. Serres, *Dynamic Characterization of a Pneumatic Muscle Actuator and Its Application to a Resistive Training Device*, Wright State University, Dayton, Ohio, USA, 2009.

[6] T. Kerscher, J. Albiez, J. M. Zöllner, and R. Dillmann, "Evaluation of the dynamic model of fluidic muscles using quick-release," in *Proceedings of the International Conference on Biomedical Robotics and Biomechatronics*, pp. 637–642, Pisa, Italy, February 2006.

[7] K. C. Wickramatunge and T. Leephakpreeda, "Empirical modeling of pneumatic artificial muscle," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, pp. 1726–1730, Hong Kong, 2009.

[8] K. C. Wickramatunge and T. Leephakpreeda, "Study on mechanical behaviors of pneumatic artificial muscle," *International Journal of Engineering Science*, vol. 48, no. 2, pp. 188–198, 2010.

[9] A. Hildebrandt, O. Sawodny, R. Neumann, and A. Hartmann, "Cascaded control concept of a robot with two degrees of freedom driven by four artificial pneumatic muscle actuators," in *Proceedings of the American Control Conference*, vol. 1, pp. 680–685, IEEE, Portland, Ore, USA, June 2005.

[10] F. Schreiber, Y. Sklyarenko, G. Runge, and W. Schumacher, "Model-based controller design for antagonistic pairs of fluidic muscles in manipulator motion control," in *Proceedings of the 17th International Conference on Methods and Models in Automation and Robotics (MMAR '12)*, pp. 499–504, Miedzyzdrojie, Poland, August 2012.

[11] S. V. Krichel, O. Sawodny, and A. Hildebrandt, "Tracking control of a pneumatic muscle actuator using one servovalve," in *Proceedings of the American Control Conference (ACC '10)*, pp. 4385–4390, Baltimore, Md, USA, July 2010.

[12] J. Zhong, J. Fan, Y. Zhu, J. Zhao, and W. Zhai, "One nonlinear pid control to improve the control performance of a manipulator actuated by a pneumatic muscle actuator," *Advances in Mechanical Engineering*, vol. 2014, Article ID 172782, 19 pages, 2014.

[13] A. Hošovský and M. Havran, "Dynamic modeling of one degree of freedom pneumatic muscle-based actuator for industrial applications," *Tehnicki Vjesnik*, vol. 19, no. 3, pp. 673–681, 2012.

[14] M. Balara and M. Tóthová, "Static and dynamic properties of the pneumatic actuator with artificial muscles," in *Proceedings of the IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY '12)*, pp. 577–581, Subotica, Serbia, September 2012.

[15] M. Tóthová, J. Piteľ, and J. Boržíková, "Operating modes of pneumatic artificial muscle actuator," *Applied Mechanics and Materials*, vol. 308, pp. 39–44, 2013.

[16] A. Hošovský, J. N. Marcinčin, J. Piteľ, J. Boržíková, and K. Židek, "Model-based evolution of a fast hybrid fuzzy adaptive controller for a pneumatic muscle actuator," *International Journal of Advanced Robotic Systems*, vol. 9, no. 56, pp. 1–11, 2012.

[17] A. Macurova and S. Hrehova, "Some properties of the pneumatic artificial muscle expressed by the nonlinear differential equation," *Advanced Materials Research*, vol. 658, pp. 376–379, 2013.

[18] FESTO Fluidic Muscle DMSP/MAS datasheet, 2008, http://www.festo.com/rep/en_corp/assets/pdf/info_501_en.pdf.

[19] P. Beater, *Pneumatic Drives: System Design, Modelling and Control*, Springer, New York, NY, USA, 2007.

[20] S. H. Zak, *Systems and Control*, Oxford University Press, New York, NY, USA, 2003.

[21] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Motion and Control*, Marcel Dekker, New York, NY, USA, 2nd edition, 2004.

[22] Hyperphysics—calculation of moments of inertia, http://hyperphysics.phy-astr.gsu.edu/hbase/mi.html.

[23] R. N. Jazar, *Theory of Applied Robotics*, Springer, New York, NY, USA, 2nd edition, 2010.

[24] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering*, Auerbach Publications, Boca Raton, Fla, USA, 1st edition, 2006.

[25] S. Haykin, *Neural Networks—A Comprehensive Foundation*, Pearson Prentice Hall, Delhi, India, 2nd edition, 2005.

[26] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing—A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, NJ, USA, 1st edition, 1997.

[27] B. H. Demuth, M. Beale, and M. T. Hagan, *Neural Network Design*, PWS Publishing, Boston, Mass, USA, 1st edition, 1996.