Designing optimal models of nonlinear MIMO systems based on orthogonal polynomial neural networks

Marko Milojković, Miroslav Milovanović, Saša S. Nikolić, Miodrag Spasić and Andjela Antić

Electronic Engineering, Department of Control Systems, University of Niš, Niš, Republic of Serbia

**ABSTRACT**

This paper presents a new method for modelling of dynamic systems by using specially designed orthogonal polynomial neural networks. These networks utilize the feature that the basis made of orthogonal functions can be used for approximation of arbitrary function, while their property of orthogonality enables optimal performances in the sense of both convergence time and approximation error. In this regard, generalized quasi-orthogonal polynomials, specifically tailored for the application in the modelling of complex dynamic systems with time-varying behaviour, are considered. Adaptivity of the designed model is achieved by using variable factors inside the orthogonal basis. The designed orthogonal neural network is applied in modelling of laboratory twin-rotor aero-dynamic system as a representative of nonlinear multiple input-multiple output systems. Detailed comparative analysis is performed for a different number of polynomials in expansion with the purpose of finding the optimal model in the sense of trade-off between model accuracy and complexity.

**1. Introduction**

Efficient modelling of complex, MIMO (multiple input–multiple output) nonlinear systems with time-varying behaviour, prone to changes in a plant environment and deviations in the plant structure, always represented a great challenge in control system theory and practice. In that sense, neural networks proved to be an excellent tool for modelling of an arbitrary dynamic system [1,2] because of their theoretical capability of approximating any measurable function to any desired degree of accuracy, having the right structure [3]. Also, modelling by neural networks belongs to black box modelling methods, where we only need input-output signals, while the knowledge of physical principle of modelled plant and solution of a possibly complicated set of mathematical equations is not required. On the other hand, finding the optimal architecture (number of hidden layers and processing elements), determining training method for the networks’ weights and setting their initial values can still be a very tedious process if we need to find the right measure of the trade-off between accuracy and network complexity [4,5].

**ARTICLE HISTORY**

Received 17 May 2020
Accepted 18 March 2021

**KEYWORDS**

Polynomial neural network; generalized orthogonal functions; modelling of dynamic systems; twin-rotor aero-dynamic system
The problem of finding optimal structure and initial parameters of general-purpose neural networks can be avoided by using special type known as polynomial neural networks [6,7]. These are single-layer neural networks based on orthogonal functions where desired output accuracy determines the required number of processing elements. Orthogonal polynomial neural networks demonstrated excellent performance in approximation of arbitrary functions in the sense of convergence time and approximation error due to their natural in-build optimality relative to other functions. Paper [8] presents one approach to training polynomial neural networks based on genetic programming. Paper [9] is also interesting because it considers training methods for Chebyshev neural network utilized for fault classification in series compensated transmission line. Neural networks with Legendre basis in hidden neurons were also used in various applications like solving systems of ordinary differential equations [10] or prediction of chaotic time series [11]. However, it should be emphasized that classical, plain Legendre basis used in previous papers is different than quasi-orthogonal variant with adaptive capabilities, for the first time considered in our paper and applied in neural networks.

Orthogonal functions applicable in practical engineering and their properties were the subject of extensive research for the authors of this paper. Several new types of orthogonal polynomials and their generalization in the form of almost and quasi-orthogonal functions were developed which are suitable for analysis and synthesis of dynamical systems, especially for their modelling [12,13]. Complete accompanying mathematical apparatus was developed in terms of the definition of the inner product, calculating the appropriate norms, as well as the deriving significant relations. Based on these functions, numerous filters were practically realized and applied [14].

The main advantage of designed orthogonal functions was the possibility to incorporate into them a certain adaptive measure of variability into them, suitable for modelling time-varying behaviour and uncertainties of real-time operating dynamical systems. That adaptive measure, when implemented into a neural network, is called endocrine mechanism, because it emulates the biological process of secreting hormones. That process enables homoeostasis and provides several advantages to the network like improved sensitivity and adaptability to environmental factors, sensed disturbances, measurement errors, and occurred uncertainties within dynamic systems. Endocrine neural networks [15,16] proved to be a valuable tool in modelling and control of dynamical systems [17].

In the recent years, there were many efforts in designing hybrid structures where neural networks were combined with the concept of fuzzy logic to suppress individual drawbacks and emphasize their advantages [18,19]. For instance, fuzzy Cohen-Grossberg neural networks [20,21], one type of cellular neural networks, have been extensively studied due to their wide applications in classification, pattern recognition, parallel computing, and image processing. These hybrid structures usually integrate fuzzy logic into the structure of neural network by replacing single steps of the inference process by different layers of the neural network. Among many efforts, there were also several attempts to improve hybrid fuzzy-neural structures by using orthogonal polynomials [22]. Authors of this paper proved that orthogonal functions can be successfully used to upgrade Adaptive Neuro Fuzzy Inference Systems (ANFIS) and design very efficient structures for various applications of modelling [23] and control [24] of dynamic systems.
The remainder of the paper is structured as follows. Section 2 describes deriving specially tailored generalized Legendre quasi-orthogonal basis to be used for designing the neural network. These polynomials have adaptive factors incorporated inside themselves. Section 3 explains the proposed orthogonal neural network model in detail, as well as the training procedure. Section 4 describes the laboratory twin-rotor aerodynamic system, which will be used as a case study in Section 5 for designing an orthogonal neural network model based on a black box approach, i.e., input-output experimental data sets. The comparative analysis is also conducted for the different number of orthogonal polynomials in expansion with the purpose to find an optimal model in the sense of trade-off between model accuracy and complexity. Similarly, an optimally designed orthogonal model was compared with some other known state-of-the-art models of twin-rotor aero-dynamic system, where it demonstrated best overall performances regarding accuracy (modelling error), training time (model complexity) and robustness to internal perturbations and external disturbances. These features make designed model a perfect candidate for application in modern control algorithms, like Model Predictive Control, based on reliable models which can be simulated quickly during complex online optimization with constraints. Finally, Section 6 concludes the paper.

2. Generalized quasi-orthogonal Legendre polynomials

Legendre polynomials were discovered in 1782 during an effort to find the force of attraction between the celestial bodies during their revolution [25]. After that, orthogonal polynomials and their properties were a subject of extensive study of many mathematicians and scientists from other fields where orthogonal functions found numerous applications, control systems engineers among them.

First, let us consider the classical definition of orthogonality and orthogonal polynomials [26]. If we have the set of polynomials \( P_n(x) \):

\[
P_n(x) = \sum_{k=0}^{n} a_k x^k,
\]

then, we say that these polynomials are orthogonal over the interval \((a, b)\) if and only if:

\[
\int_{a}^{b} P_n(x)P_m(x)w(x)dx = \begin{cases} 
0, & n \neq m \\
N_n \neq 0, & n = m
\end{cases}
\]

(2)

where \(a\) and \(b\) are the boundaries of orthogonality interval and \(w(x)\) is weight function.

For our analysis of dynamical systems operating in real-world conditions, which are prone to a deviation of nominal values of their components because of wearing over time or under environmental changes (temperature, humidity, pressure), it is convenient not to consider orthogonal polynomials in their pure form (2), but rather polynomials are known in literature as almost orthogonal [12]. The integral is no longer equal to the zero for \( n \neq m \), but, because of the mentioned conditions of real-time operating, is some constant very close to zero. That adaptive variable will enable us to partially overcome the drawback of the black-box modelling approach that it is generally valid only for a narrow set of signals, it was calculated from.
One important class of orthogonal polynomials, convenient for technical applications because of their properties [26], are Legendre polynomials, defined over orthogonality interval \([-1, 1]\) and with weight function \(w(x) = 1\). For applications in control systems theory, we should rather consider shifted Legendre polynomials in their explicit form where the interval of orthogonality is shifted to \([0, 1]\) by using affine transformation, i.e., shifting function \(x \rightarrow 2x - 1\):

\[
P_n(x) = \frac{1}{n!} \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} \frac{(n+j)!}{j!} x^j. \tag{3}
\]

Shifting enables later application of Legendre polynomials inside technical systems that operate in real-time, so, with additional substitution \(x = e^{-t}\), we can obtain exponential polynomials orthogonal over the real-time interval \((0, \infty)\). By applying the Laplace transform to the exponential orthogonal polynomials, we get orthogonal rational functions, capable of emulating transfer functions of arbitrary dynamic systems [12].

If we now apply the definition of almost orthogonality on shifted Legendre polynomials (3), we obtain the sequence of almost orthogonal Legendre polynomials:

\[
P_n^{(\delta)}(x) = \sum_{i=0}^{n} A_{n,i}^{\delta} x^i \tag{4}
\]

where:

\[
A_{n,i}^{\delta} = (-1)^{n+i} \frac{\Gamma(n\delta + i + 1)}{\Gamma(n\delta + 1)! (n-i)!} \tag{5}
\]

\(\Gamma\) is the symbol for the gamma function and \(\delta \approx 1\) is a constant very close to one.

The final form of generalized Legendre polynomials can be derived by using the following definition of quasi-orthogonality [27]:

\[
\int_{a}^{b} P_n^k(x) P_m^k(x) w(x) dx = \begin{cases} 
0, & 0 \leq m \leq n - k - 1 \\
N_{n,m}^k \neq 0, & n \geq k + 1
\end{cases} \tag{6}
\]

where \(k\) represents the order of quasi-orthogonality, and \(a\) and \(b\) are the boundaries of quasi-orthogonality interval. Filters based on these polynomials enable simpler modelling of the systems with the difference in the degree of polynomials in the numerator and denominator in transfer function greater than one (equal to \(k + 1\)). If we apply definition (6) to the sequence of almost orthogonal Legendre polynomials given by (4) and (5), we obtain generalized quasi-orthogonal Legendre polynomials over the interval \((0, 1)\):

\[
P_n^{(k,\delta)}(x) = \sum_{i=0}^{n} A_{n,i}^{k,\delta} x^i \tag{7}
\]

where:

\[
A_{n,i}^{k,\delta} = (-1)^{n+i+k} \frac{\prod_{j=1}^{n-k} (i+j\delta)}{i!(n-i)!} \tag{8}
\]
For example, here are the first few second-order \( (k = 2) \) generalized quasi-orthogonal Legendre polynomials of this sequence, which will be used for designing the orthogonal neural network with the task of modelling twin-rotor system:

\[
\begin{align*}
    p_1^{(2,\delta)}(x) &= (1 - \delta^2)x, \\
    p_2^{(2,\delta)}(x) &= (\delta + 2)x^2 - (\delta + 1)x, \\
    p_3^{(2,\delta)}(x) &= \frac{(\delta + 3)}{6}x^3 - \frac{(\delta + 2)}{2}x^2 + \frac{(\delta + 1)}{2}x - \frac{\delta}{6}, \\
    p_4^{(2,\delta)}(x) &= \frac{(\delta + 2)(\delta + 4)}{12}x^4 - \frac{(\delta + 3)(2\delta + 3)}{6}x^3 + \\
    &\quad + \frac{(\delta + 1)(\delta + 2)}{2}x^2 - \frac{(\delta + 1)(2\delta + 1)}{6}x + \frac{\delta^2}{12}, \\
    p_5^{(2,\delta)}(x) &= \frac{(\delta + 5)(2\delta + 5)(3\delta + 5)}{120}x^5 - \frac{(\delta + 2)(\delta + 4)(3\delta + 4)}{12}x^4 + \\
    &\quad + \frac{(\delta + 1)(\delta + 3)(2\delta + 3)}{4}x^3 - \frac{(\delta + 1)(\delta + 2)(3\delta + 2)}{6}x^2 + \\
    &\quad + \frac{(\delta + 1)(2\delta + 2)(3\delta + 1)}{24}x - \frac{\delta^3}{20}.
\end{align*}
\]

It should be mentioned that another orthogonal polynomial basis could be also used for designing orthogonal neural network model described in the next section, for instance Chebyshev or Laguerre. Legendre polynomials are chosen because authors have already developed whole mathematical framework for generalized quasi-orthogonal variant of these polynomials with a lot of useful mathematical relations, including relations analogue to Christoffel and Rodrigues formulas and Bonnet recurrence relation valid for classical orthogonal polynomials \([13,14,28]\). Programming modules were also previously developed with all the functions necessary for easy programming of designed models.

### 3. Orthogonal neural network models

Developed quasi-orthogonal functions given by \((7)\) and \((8)\) can be used for designing very efficient adaptive polynomial neural networks. These networks can be treated as a special subtype of higher-order neural networks \([29]\). The structure of such a network is presented in Figure 1. It produces a function that represents a combination of Legendre (or other) quasi-orthogonal polynomials and can be used for approximation of an arbitrary function. Thereby, orthogonal polynomial expansion is superior to approximations by other functions because of natural in-build optimality of orthogonal basis, leading to the better accuracy of the approximation and shorter convergence time for a similar number of terms in the expansion.

The polynomial neural network shown in Figure 1 is applied in modelling of an arbitrary plant (object of modelling). Theoretical background for such an application can be found in the fact that an arbitrary function \(y(X)\), where \(X\) represents the vector of inputs \(X = [x_1, x_2, \ldots, x_m]\), can be approximated by orthogonal expansion of quasi-orthogonal polynomials \(p_i(X)\) of a neural network:
\[ y_{\text{model}}(X, y_{\text{plant}}, \delta) = f \left( \sum_{i=1}^{n} w_i p_i(X, y_{\text{plant}}, \delta) + b \right) + R(X, y_{\text{plant}}, \delta, n) \quad (10) \]

where \( w_i \) is weight of the network, \( f \) is an activation function and \( R(X, y_{\text{plant}}, \delta, n) \) is the approximation error that decreases with an increase of the number of expansion terms \( n \) [30]. Label \( x \) in Figure 1 and equation (10), marking input signals to the plant, should not be confused with \( x \) in equations (7) and (8), used for generating quasi-orthogonal Legendre polynomials, where \( x \) is just a label of a general variable.

As already explained in Section 2, \( \delta \) is an adaptive variable with a value close to one that enables the feature of almost orthogonality i.e., shifted orthogonality which can describe small deviations in plant structure or operating mode because of wearing over time or environmental changes. In that way, designed models will gain some level of flexibility and validity over a wider range of operating conditions.

Past instances of the plant inputs/outputs are also included in orthogonal neural network design, providing its recurrence. These past samples of input/output signals are provided by blocks labelled TDL (Tapped Delay Line) which memorize signal values in previous instances. Block named Legendre expansion then generates quasi-orthogonal Legendre polynomials \( (P_i) \) based on current and time-delayed values of inputs and outputs and the value of \( \delta \). In Figure 1 there are \( a \) previous input values and \( b \) previous output values. We experimented with up to three delayed input/output samples and, in experiments presented in this paper, we used \( a = 1 \) and \( b = 2 \).

An adequate model of the plant of interest is obtained by finding the optimal values for network weights \( w_i \) \( (i = 1, 2, \ldots, n) \) through the training of the polynomial network. For that purpose, the chosen training algorithm uses some performance function, usually

**Figure 1.** Orthogonal neural network applied in modelling of an arbitrary dynamic system.
based on modelling error – difference between the plant \(y_{plant}\) and model \(y_{model}\) outputs, with the goal to minimize it. Several papers deal with optimal training methods for polynomial neural networks [6,9,31]. The performed comparative analysis demonstrated that the best training method for polynomial neural networks in the sense of accuracy and convergence time is the Levenberg-Marquardt algorithm or damped least-squares method [32,33]. This algorithm is based on a gradient vector and the Jacobian matrix and is designed specifically for performance (cost) function given in the form of a sum of squared errors:

\[
Q = \sum_{k=1}^{m} (y_{plant}(k) - y_{model}(k))^2 = \sum_{k=1}^{m} e_k^2
\]

(11)

where \(m\) is the number of instances in the input-output data set.

The Levenberg-Marquardt algorithm is practically modified Newton’s method for function optimization with respect to a specific variable. During the execution of this training algorithm, the change in weight vector is calculated as:

\[
\Delta w = \left[ J^T(w)J(w) + \lambda I \right]^{-1} J^T(w)e(w)
\]

(12)

where \(w\) is a weight vector with \(n\) adjustable parameters and \(J\) represents Jacobian matrix with size \(m \times n\) containing the derivatives of the errors with respect to the weights:

\[
J_{i,j} = \frac{\partial e_i}{\partial w_j}
\]

(13)

for \(i = 1, \ldots, m\) and \(j = 1, \ldots, n\). Parameter \(\lambda\) is called the damping factor, and, it is usually initialized to be large, so at first, updates are small steps in the gradient descent direction. In the next step, if the cost function increases, then \(\lambda\) should be increased even more. Otherwise, if the cost function decreases, \(\lambda\) should be decreased, and, in that case, the Levenberg-Marquardt algorithm approaches the Gauss-Newton method. The Levenberg-Marquardt algorithm is considered to be very fast and easy to implement by simple modifications in the traditional backpropagation method.

### 4. Case study: twin-rotor aero-dynamic system

As a case study for modelling of dynamical systems by using designed orthogonal neural network models, explained in Section 3, laboratory twin-rotor aero-dynamic system (TRAS), shown in Figure 2, was used. This system is convenient because of its multiple input–multiple output (MIMO) nature with present high order nonlinear dynamics and cross-couplings, very difficult to model with conventional methods based on Lagrange equations. It is also very prone to parameter variations, external disturbances, and unwanted induced vibrations. The system is controlled in real-time from a PC in the MATLAB/Simulink environment [34].

As its name suggests, the main components of TRAS (see Figure 2) are two rotors – the main rotor and tail rotor, driven by DC motors. These rotors (propellers) are attached at the ends of the beam with a counterbalance pivoted on its base and enable the beam to rotate in the horizontal and vertical planes, so the system has two degrees of freedom defined by pitch and azimuth angles. Besides these two variables measured by position
sensors (encoders) at the pivot point, the state of the beam is also described by corresponding angular velocities which are software reconstructed. There are two more state variables – angular velocities of the rotors, measured by associated tacho-generators.

It should be mentioned that this laboratory system acts like a helicopter only in some aspects and it is not suitable for serious considerations regarding aeromechanics and flight control. It cannot fly through the air and imitate all aerodynamical effects like gusts of wind, but it will be good enough for our purposes, because we only need system with multiple inputs, cross-couplings, easy appliance of perturbations or disturbances, and developed computer functions for comfort experimenting, in order to validate designed modelling method presented in this paper.

Controlling the TRAS system usually means stabilizing the beam in an arbitrary desired position or making it track the desired trajectory. Controlled variables (output signals) are horizontal position (azimuth angle) and vertical position (pitch angle) of the TRAS beam \( \alpha_h \) and \( \alpha_v \), respectively. Control inputs for TRAS are supply voltages of the DC motors \( U_h \) and \( U_v \) in Figure 3. Change in input voltage results in the different rotation speed of the corresponding rotor (propeller) and appropriate change in the beams’ position. However, there are also significant cross-couplings between the actions of the rotors, resulting in each rotor influencing both position angles.

Modern methods of design of real-time controllers require high-quality models of the system, and, as already stated, because of non-linear dynamics, cross-couplings, parameter variations, susceptibility to external disturbances and other factors, modelling of twin-rotor aero-dynamic system is not an easy task, if we approach it from the point of the white box – from the point of complete mathematical-physical analysis. This

![Figure 2. Laboratory TRAS system and main components.](image)
approach, although it provides a general model valid for the whole range of plant inputs and states, is only suitable for simple plants with a small number of parameters. Comprehensive models of TRAS with necessary mathematical and physics background can be found in [35–37].

In this paper, for the purpose of comparing designed adaptive orthogonal neural network model with some state-of-the-art models, two other white-box models will be analysed: linearized model described in [38] and default nonlinear model provided with laboratory set-up given in Figure 3 without further explaining of labels, which can be found in [34].

5. Modelling of twin-rotor aero-dynamic system by using adaptive orthogonal polynomial neural network

As a case study for designing optimal models of nonlinear MIMO systems by using orthogonal polynomial neural network, laboratory twin-rotor aero-dynamic system – TRAS (Figure 2) was considered. General neural network structure (Figure 1) was adapted for two input-two output system, where inputs are voltages of DC motors: \( U_h \) – the horizontal DC-motor PWM control signal (tail rotor) and \( U_v \) – the vertical DC-motor PWM control signal (main rotor), and outputs are: \( \alpha_h \) – horizontal position (azimuth angle) and \( \alpha_v \) – vertical position (pitch angle) of the TRAS beam. The adjusted model in parallel to the laboratory system is presented in Figure 4. Block named Legendre expansion produces generalized orthogonal polynomials according to (7) and (8) by using input/output current and time delayed samples (one previous sample for inputs and two for outputs have been used in presented experiments). Having both inputs in polynomial development, we can easily emulate input-output
cross-coupling inside the plant. In all experiments, second-order \((k = 2)\) Legendre polynomials \((9)\) were used because of their property to model well internal transfer functions characteristic for the twin-rotor system with the difference of three in orders of numerator and denominator. Starting experiments were performed for strictly quasi-orthogonal polynomials with value of parameter \(\delta\) equal to one. Later, \(\delta\) was perturbated in order to simulate adaptability to changed environmental factors, sensed disturbances, measurement errors, and occurred uncertainties. As activation function, classical sigmoidal function \(f(x) = \frac{1}{1+e^{-x}}\) was applied, guarantying a smooth, differentiable model. Bias \(b\) was not needed in our experiments, i.e., it was set to zero. Weights in the model were adjusted by Levenberg-Marquardt training algorithm based on error, i.e., the difference between real – measured outputs and model outputs, as explained in Section 3.

Modelling process described in this paper is based on a black-box approach (input-output experimental data sets). The same input signals were applied both to the orthogonal model and a plant (TRAS). Thereby, the input signals \(U_h\) and \(U_v\) are normalized in the range \([-1, +1]\) corresponding to the input voltage range of \([-24 \, \text{V}, +24 \, \text{V}\]). In nonlinear system identification, input signals are usually chosen to have different levels of amplitudes to be able to excite most of the dynamic modes of interest. So, the following normalized input voltages (Figure 5) driving the motors were applied: the azimuth input – square wave with 0.15 amplitude and 1/50 [Hz] frequency, and the pitch signal – sine wave with the amplitude 0.25 and frequency 1/50 [Hz]. Signals in input/output data sets were sampled with period of 0.01 seconds with overall duration of the excitation of 100 seconds.

To find an optimal model in the sense of trade-off between model accuracy and complexity, experiments were repeated for a different number of terms in the polynomial expansion. Two criteria were considered: model accuracy and training time. Model accuracy was evaluated by using the performance index in the form of root-mean-square error (RMSE):

**Figure 4.** Set up for modelling of laboratory twin-rotor aero-dynamic system by using orthogonal polynomial neural network.
After training the network by applying training input data set and calculating RMSE, model accuracy was also considered for two more testing sets with different shapes than original. Testing set 1 had for the azimuth input – sine wave with 0.75 normalized amplitude and 1/25 [Hz] frequency, and for the pitch input – sine wave with the amplitude 0.25 and frequency 1/25 [Hz]. Testing set 2 had for the azimuth input – sine wave with 0.4 normalized amplitude and 1/50 [Hz] frequency, and for the pitch input – sawtooth signal with the amplitude 0.6 and frequency 1/50 [Hz]. The results are presented in Table 1.

We can see from results that model accuracy expectedly increases with the increase of the number of terms (polynomials) in orthogonal expansion. On the other hand, training time also increases rapidly, so, if the network complexity and speed of training are most important, higher-order models soon become unusable. Small gains in accuracy do not always justify unnecessary complex networks. Similar trends regarding model accuracy are also present for two testing sets (two responses to different inputs), with obvious discrepancy for the model built with seven terms in the polynomial expansion. The reason for this decrease in accuracy can be found in the effect of overfitting, i.e., reduction in generalization ability for higher-order polynomials. On the other hand, lower-order

\[
RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left( (y_{hp}(k) - y_{hm}(k))^2 + (y_{vp}(k) - y_{vm}(k))^2 \right) }
\]

(14)

**Figure 5.** Shapes of applied input signals \(Uh\) and \(Uv\).

| Number of terms in the orthogonal expansion | 3     | 4     | 5     | 6     | 7     |
|-------------------------------------------|-------|-------|-------|-------|-------|
| Training set RMSE                         | 11.0822 | 5.4101 | 2.3880 | 0.9145 | 0.6263 |
| Testing set 1 RMSE                        | 18.3990 | 7.7742 | 4.0718 | 2.7408 | 3.7751 |
| Testing set 2 RMSE                        | 16.8043 | 7.0016 | 3.7222 | 2.9035 | 4.2407 |
| Training time (sec)                        | 4.52 | 7.07 | 12.12 | 31.05 | 80.63 |
networks are inadequate to achieve targeted modelling precision. We can see that there exists some optimal value for the number of expansion terms which achieves the best accuracy, not only for training, but also for testing data, and all that with bearable training time. In this particular case, that optimal expansion is with six terms. Parallel responses of the laboratory TRAS and optimal orthogonal model with six terms are given in Figures 6 and 7.

Let’s now compare designed orthogonal polynomial models with some other known models of twin-rotor aero-dynamic system:

**Figure 6.** Azimuth angles for real system and optimal orthogonal neural network model.

**Figure 7.** Pitch angles for real system and optimal orthogonal neural network model.
- The linearized model described in detail in [38], obtained by analytic linearization implying assumption for small deviations of the variables around their nominal operating values.
- The nonlinear model provided with laboratory set-up as a part of the Simulink environment shown in Figure 3.
- Nonlinear ARX neural network [39] with two hidden layers with 10 neurons each.

The first and the second model are white-box first principle models with included nominal values of the laboratory set-up parameters given in [34], and third is black-box model – neural network trained just like our orthogonal neural network. Modelling results are presented in Table 2.

It should be noted that for linearized and nonlinear models we do not have a training phase, so given results are actually for training set used just as another testing set. We can see that the linearized model cannot catch all the complexities inside the system, resulting in the worst accuracy results. The nonlinear system is better, but not as good as NARX neural network, confirming the assumption that black-box models, although not interpretable as the first principle model, provide much better accuracy for complex MIMO systems, being able to mimic all dynamic modes in these systems. NARX network achieves satisfying results and, it should be mentioned, with much shorter training time, but that result is still with almost double RMSE than for optimal orthogonal neural network.

The true power of orthogonal networks comes to the surface if we introduce changes to original nominal systems imitating environmental changes, measurement errors or occurred uncertainties in the form of perturbations around nominal values of 1% and 3% and repeat the calculation of RMSE for developed models of TRAS. Perturbations are incorporated into experiments by programming artificial measurement error (noise) into feedback information obtained by position sensors (encoders) responsible for reporting current TRAS position, i.e., azimuth and pitch angles. Percentage of change in relation to normal (nominal) system response is calculated as a ratio of overall signal power, i.e. noise to signal ratio. We can see from the results in Table 3 that other models are rigid and cannot follow the changes well. On the other hand, designed orthogonal model possess incorporated measure of variations – δ that can be now set to 1.01 and 1.03 for changes of 1% and 3% respectively, making models adaptive to changes by shifting a little

### Table 2. Accuracy of different models of TRAS for nominal conditions.

| Models                  | Linearized | Nonlinear | NARX    |
|-------------------------|------------|-----------|---------|
| Training set RMSE       | 13.5390    | 7.6219    | 3.7287  |
| Testing set 1 RMSE      | 17.9333    | 6.4844    | 5.1992  |
| Testing set 2 RMSE      | 14.3101    | 6.0037    | 5.0017  |

### Table 3. Accuracy of different models of TRAS for changed working conditions.

| Models                           | Linearized | Nonlinear | NARX    | Optimal orthogonal NN |
|----------------------------------|------------|-----------|---------|-----------------------|
| Testing set 1 RMSE, δ = 1.01     | 19.7431    | 9.8333    | 6.5518  | 2.3401                |
| Testing set 2 RMSE, δ = 1.01     | 17.4400    | 9.0252    | 7.6309  | 2.9185                |
| Testing set 1 RMSE, δ = 1.03     | 22.2851    | 11.8824   | 8.2194  | 2.5410                |
| Testing set 2 RMSE, δ = 1.03     | 20.7373    | 11.0909   | 8.1766  | 2.8813                |
their orthogonal inner products, without the need to train the model again, and obtaining similar accuracy, that way.

6. Conclusion

This paper considers a method for modelling complex nonlinear MIMO systems based on designing one new type of orthogonal polynomial neural networks. These networks use the basis of specially tailored quasi-orthogonal functions suitable for modelling dynamic systems that utilize the feature of orthogonal polynomial expansions to approximate arbitrary functions with any given accuracy (approximation error), of course, by using the sufficient number of terms in the expansion. Thereby, orthogonal polynomials are naturally superior to other functions in the sense of optimal development (same accuracy for fewer terms in expansion or better accuracy for the same number of terms). Applied quasi-orthogonal polynomials (Legendre in this case) are very convenient for catching time-varying behaviour and perturbations in real systems thanks to the incorporated adaptive factor inside the polynomial terms.

Coefficients in the polynomial expansion are weights in a neural network that can be optimized – trained by using classical training methods. In this research, the modified Levenberg-Marquardt algorithm was applied. Thanks to the orthogonal basis and single-layer structure, orthogonal neural network avoids the main problem of traditional feedforward neural networks – finding the optimal structure in terms of the numbers of layers and processing elements, as well as the initial values of weights. The required number of processing elements, which are composed of the expansion terms of quasi Legendre polynomials, is determined according to the desired output accuracy. Due to uniqueness of the weights, rapid convergence during training of weights is ensured.

Modelling potential of designed adaptive orthogonal polynomial neural networks was verified on the example of twin-rotor aero-dynamic system as a suitable representative of complex nonlinear multiple input–multiple output systems with present cross-couplings, and prone to external disturbances and unwanted induced vibrations. Modelling of such a system can be very challenging for traditional first-principles methods, but also to other black-box methods based on identification by using input-output data sets. First, a set of models were designed with a different number of terms in the polynomial expansion. It was proven that there exists a sweet spot, i.e., optimal number of terms that guarantee the best trade-off between model accuracy and complexity, expressed by computation time. The effect of overfitting was also noticed for larger polynomial expansions.

The next step was to compare the optimal orthogonal polynomial model with three other known models of twin-rotor aero-dynamic system: first – obtained by analytic linearization, second – nonlinear Simulink model provided by laboratory twin-rotor aero-dynamic system manufacturer and third – Nonlinear ARX neural network. Optimal orthogonal neural network demonstrated best overall performances among all models, especially better than the first two – white-box models. The superiority of the orthogonal model is even more evident when we introduce disturbances and measurement noise into the system. Thanks to the adaptive factor, which is part of the quasi-orthogonal expansion, the orthogonal model can recognize perturbations and provide the same level of accuracy under new conditions.
Designed orthogonal models can be especially useful in modern control algorithms based on reliable models, like Model Predictive Control (MPC) [40]. MPC is a feedback control algorithm that uses the model to predict future plant outputs and solves optimization problem to select optimal control, while satisfying a set of constraints. This type of control can handle well MIMO systems, like TRAS, because it is multivariable control that controls the outputs simultaneously by considering all the interactions between system variables [41]. On the other hand, for instance, PID controller design would be challenging in this situation because the separate control loops for different system variables would operate independent of each other, as if there are no interactions between them. Another problem would be tuning too many controller gains.

The biggest problem of MPC, that slowed the applications of that control algorithm for several decades, is the need for a lot of calculation, because the algorithm must solve a complex online optimization problem with constraints at each time step. That can be very challenging in the case of complex models of the plants and can require very strong processors with a large memory. Thanks to the single-layer structure and less model parameters, orthogonal polynomial models presented in this paper demand lot less calculations during determining outputs and should provide much better performances for MPC in the sense of shorter calculation time for the same time prediction horizon or the larger horizon for the same calculation time, comparing to other models. First experiments in that direction have already proved to be encouraging.

Acknowledgments

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Marko Milojković https://orcid.org/0000-0001-7623-1495
Saša S. Nikolić https://orcid.org/0000-0003-2745-3862
Miodrag Spasić https://orcid.org/0000-0002-3061-6872

References

[1] M.H. Hassoun, Fundamentals of Artificial Neural Networks, MIT Press, Cambridge, Massachusetts, London, 1995.
[2] K.-L. Du and M.N.S. Swamy, Neural Networks and Statistical Learning, Springer-Verlag, London, 2019.
[3] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks 4 (2) (1991), pp. 251–257. doi:10.1016/0893-6080(91)90009-T.
[4] D. Hunter, H. Yu, M.S. Pukish, J. Kolbusz, and B.M. Wilamowski, Selection of proper neural network sizes and architectures—A comparative study, IEEE T. Ind. Inform 8 (2) (2012), pp. 228–240. doi:10.1109/TII.2012.2187914.

[5] S. Ding, H. Li, C. Su, J. Yu, and F. Jin, Evolutionary artificial neural networks: A review, Artif. Intell. Rev. 39 (2013), pp. 251–260. doi:10.1007/s10462-011-9270-6.

[6] S.-K. Oh, W. Pedrycz, and B.-J. Park, Polynomial neural networks architecture: Analysis and design, Comput. Electr. Eng 29 (6) (2003), pp. 703–725. doi:10.1016/S0045-7906(02)00045-9.

[7] S.-S. Yang and C.-S. Tseng, An orthogonal neural network for function approximation, IEEE T. Syst. Man C.Y.-S 26 (5) (1996), pp. 779–785. doi:10.1109/3477.537319.

[8] N.Y. Nikolaev and H. Iba, Learning polynomial feedforward neural networks by genetic programming and backpropagation, IEEE T. Neural Networ 14 (2) (2003), pp. 337–350. doi:10.1109/TNN.2003.809405.

[9] B.Y. Vyas, B. Das, and R.P. Maheshwari, Improved fault classification in series compensated transmission line: Comparative evaluation of Chebyshev neural network training algorithms, IEEE T. Neur. NeT. Lear 27 (8) (2016), pp. 1631–1642. doi:10.1109/TNNLS.2014.2360879.

[10] Y. Yang, M. Hou, and J. Luo, A novel improved extreme learning machine algorithm in solving ordinary differential equations by Legendre neural network methods, Adv Differ Equ 469 (1) (2018), pp. 469. doi:10.1186/s13662-018-1927-x.

[11] H. Wang and H. Gu, Prediction of chaotic time series based on neural network with Legendre polynomials, 6th International Symposium on Neural Networks, Wuhan, China, 2009.

[12] M. Milojković, S. Nikolić, B. Danković, D. Antić, and Z. Jovanović, Modelling of dynamical systems based on almost orthogonal polynomials, Math. Comp. Model. Dyn 16 (2) (2010), pp. 133–144. doi:10.1080/13873951003740082.

[13] D. Antić, B. Danković, S. Nikolić, M. Milojković, and Z. Jovanović, Approximation based on orthogonal and almost orthogonal functions, J. Franklin. I 349 (1) (2012), pp. 323–336. doi:10.1016/j.jfranklin.2011.11.006.

[14] M. Milojković, D. Antić, S. Nikolić, Z. Jovanović, and S. Perić, On a new class of quasi-orthogonal filters, Int. J. Electron 100 (10) (2013), pp. 1361–1372. doi:10.1080/00207172.2012.743087.

[15] J. Timmis, M. Neal, and J. Thorniley, An adaptive neuro-endocrine system for robotic systems, IEEE workshop on robotic intelligence in informationally structured space, Nashville, TN, USA, 2009.

[16] C. Sauze and M. Neal, Artificial endocrine controller for power management in robotic system, IEEE T. Neur. NeT. Lear 24 (12) (2013), pp. 1973–1985. doi:10.1109/TNNLS.2013.2271094.

[17] M. Milovanović, D. Antić, M. Milojković, S. Nikolić, S. Perić, and M. Spasić, Adaptive PID control based on orthogonal endocrine neural networks, Neural Networks 84 (2016), pp. 80–90. doi:10.1016/j.neunet.2016.08.012.

[18] Y.-J. Zheng, H.-F. Ling, S.-Y. Chen, and J.-Y. Xue, A hybrid neuro-fuzzy network based on differential biogeography-based optimization for online population classification in earthquakes, IEEE T. Fuzzy Syst 23 (4) (2015), pp. 1070–1083. doi:10.1109/TFUZZ.2014.2337938.

[19] J.-S. Jang, ANFIS adaptive-network-based fuzzy inference system, IEEE T. Syst. Man C.Y.-S 23 (4) (1993), pp. 655–685. doi:10.1109/21.256541.

[20] M. Abdelaziz and C. Farouk, Exponential lag synchronization and global dissipativity for delayed fuzzy Cohen–Grossberg neural networks with discontinuous activations, Neural Process. Lett 51 (2) (2020), pp. 1653–1676. doi:10.1007/s11063-019-10169-8.

[21] M. Abdelaziz and C. Farouk, Piecewise asymptotic almost periodic solutions for impulsive fuzzy Cohen–Grossberg neural networks, Chaos Soliton. Fract 132 (2020), pp. 109575.

[22] B.-J. Park, W. Pedrycz, and S.-K. Oh, Fuzzy polynomial neural networks: Hybrid architectures of fuzzy modeling, IEEE T. Fuzzy Syst 10 (5) (2002), pp. 607–621. doi:10.1109/TFUZZ.2002.803495.
[23] M. Milojković, D. Antić, M. Milovanović, S. Nikolić, S. Perić, and M. Almawlawe, *Modeling of dynamic systems using orthogonal endocrine adaptive neuro-fuzzy inference systems*, J. Dyn. Syst.-T. ASME. 137 (9) (2015), pp. 091013-1–091013-6. DS-15-1098. 10.1115/1.4030758.

[24] S. Perić, D. Antić, M. Milovanović, D. Mitić, M. Milojković, and S. Nikolić, *Quasi-sliding mode control with orthogonal endocrine neural network-based estimator applied in anti-lock braking system*, IEEE-ASME T. Mech 21 (2) (2016), pp. 754–764. doi:10.1109/TMECH.2015.2492682.

[25] M. Kline, *Mathematical Thought from Ancient to Modern Times*, Oxford University Press, New York, 1972.

[26] G. Szegő, *Orthogonal Polynomials*, American Mathematical Society, Colloquium Publications, 1975.

[27] D. Antić, Z. Jovanović, V. Nikolić, M. Milojković, S. Nikolić, and N. Danković, *Modeling of cascade-connected systems using quasi-orthogonal functions*, Elektron. Elektrotech 18 (10) (2012), pp. 3–8. doi:10.5755/j01.eee.18.10.3051.

[28] S.S. Nikolić, D.S. Antić, M.T. Milojković, M.B. Milovanović, S.L. Perić, and D.B. Mitić, *Application of neural networks with orthogonal activation functions in control of dynamical systems*, Int. J. Electron 103 (4) (2016), pp. 667–685. doi:10.1080/00207217.2015.1036811.

[29] M.M. Gupta, I. Bukovsky, N. Homma, A.M.G. Solo, and Z.G. Hou, *Fundamentals of higher order neural networks for modeling and simulation*, in *Artificial Higher Order Neural Networks for Modeling and Simulation Chapter 6*, M. Zhang ed., IGI Global, 2013. pp.103-133.

[30] C.F. Sher, C.-S. Tseng, and C.-S. Chen, *Properties and performance of orthogonal neural network in function approximation*, Int. J. Intell. Syst 16 (12) (2001), pp. 1377–1392. doi:10.1002/int.1065.

[31] A. Gautier, Q. Nguyen, and M. Hein, *Globally Optimal Training of Generalized Polynomial Neural Networks with Nonlinear Spectral Methods*, Neural Information Processing Systems Barcelona, Spain, 2016.

[32] M. Bogdan, M. Wilamowski, and H. Yu, *Improved computation for Levenberg–Marquardt training*, IEEE T. Neural Networ 21 (6) (2010), pp. 930–937. doi:10.1109/TNN.2010.2045657.

[33] I. Marić, *Optimization of self-organizing polynomial neural networks*, Expert Syst. Appl 40 (11) (2013), pp. 4528–4538. doi:10.1016/j.eswa.2013.01.060.

[34] *Two Rotor Aero-dynamical System (TRAS) - User's Manual for MATLAB*, Inteco, Poland, 2009, available at http://www.inteco.com.

[35] P. Chalupa, J. Přikryl, and J. Novák, *Modelling of twin rotor MIMO system*, International Symposium on Intelligent Manufacturing and Automation, Procedia Engineering 100 (2015), pp. 249–258, Ipoh, Malaysia.

[36] T.D. Huu and I.B. Ismail, *Modelling of twin rotor MIMO system*, IEEE International Symposium on Robotics and Manufacturing Automation, Ipoh, Malaysia, 2016.

[37] P.S. Kannan and P. Sheenu, *PID control of twin rotor MIMO system*, International Journal of Advanced Research in Basic Engineering Sciences and Technology 3 (2017), pp. 1–7.

[38] P. Petkov, N. Christov, and M.M. Konstantinov, *Robust real-time control of a two-rotor aerodynamic system*, World Congress: The International Federation of Automatic Control, Seoul, Korea, 2008.

[39] H.T. Siegelmann, B.G. Horne, and C.L. Giles, *Computational capabilities of recurrent NARX neural networks*, IEEE T. Syst. Man C.Y.-S 27 (2) (1997), pp. 208–215. doi:10.1109/3477.558801.

[40] J.B. Rawlings and D.Q. Mayne, *Model Predictive Control: Theory and Design*, Nob Hill Publishing, Madison, Wisconsin, USA, 2009.

[41] R. Raghavan and S. Thomas, *MIMO model predictive controller design for a twin rotor aerodynamic system*, IEEE International Conference on Industrial Technology, Taipei, Taiwan, 2016.