http://dx.doi.org/10.5755/j02.eie.28780

Model Predictive Control of Nonlinear MIMO Systems Based on Adaptive Orthogonal Polynomial Networks

Marko T. Milojkovic*, Andjela D. Djordjevic, Stanisa Lj. Peric, Miroslav B. Milovanovic,
Zoran H. Peric, Nikola B. Dankovic
Faculty of Electronic Engineering, University of Nis,
Aleksandra Medvedeva 14, 18000 Nis, Republic of Serbia
marko.milojkovic@elfak.ni.ac.rs

***Abstract*—This paper considers a new design of model predictive control based on specific models in the form of adaptive orthogonal polynomial networks, built around a specially tailored basis of generalized orthogonal functions. Polynomial model has a single layer structure and a smaller number of model parameters than classical neural networks, usually used for model predictive control design, leading to lower complexity and shorter calculation time. Desired property of adaptability of the model is achieved by using additional variable factors inside the orthogonal basis. The designed controller was applied in control of twin-rotor aero-dynamic system as a representative of nonlinear multiple input-multiple output systems and compared to the other state-of-the-art control algorithms.**

***Index Terms*—Adaptive polynomial neural network; Generalized orthogonal functions; Model predictive control; Twin-rotor aero-dynamic system.**

I. INTRODUCTION

Model predictive control (MPC) is a feedback control algorithm that uses the model of the plant to predict future plant outputs over a specified time horizon. These predictions are then utilized for selecting the optimal control by solving a certain optimization problem while satisfying a set of predefined constraints [1], [2]. First applications of MPC were in the process industries in chemical plants and oil refineries already in the 1980s, but the real boom happened only in the last years with the development of very strong processors with large memory at an affordable price. The reason that slowed the applications of this control algorithm for several decades and the main drawback of MPC is its need for many calculations, because the algorithm must solve a complex online optimization problem with constraints at each time step [3]. Necessary calculational time was often much greater than the time available in real time control, so in the beginning, MPC was typically applied only in slow processes. Today, MPC is the most widely used advanced control technology in various technical fields like aerospace, robotics, energy production, food processing, industrial manufacturing, mining, and metallurgy [4]–[6].

MPC proved to handle multiple-input multiple-output (MIMO) systems exceptionally well because of its multivariable control that controls the outputs simultaneously by considering all interactions between system variables and constraints on the input, output, as well as state [7]. On the other hand, traditional control algorithms like, e.g., PID control, 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 [8], [9]. Another problem would be tuning too many controller gains. In addition, unlike PID control, MPC has the ability to anticipate future events and can take control actions accordingly, ensuring good tracking performance, closed-loop stability, and robustness.

In this work, MPC of a laboratory twin-rotor aero-dynamic system (TRAS) is considered. Laboratory TRAS [10] imitates a simplified helicopter with two degrees of freedom represented by two rotors (main and tail) driven by DC motors. Control goal usually is to stabilize the beam carrying the rotors in an arbitrary position (azimuth and pitch angles) or to make it track some desired trajectory. This system has MIMO nature with present high-order nonlinear dynamics and cross-couplings, and is prone to parameter variations, external disturbances, and unwanted induced vibrations. These features are making TRAS very challenging for designing an appropriate control algorithm. Over the years, many different strategies for TRAS control have been developed, ranging from classical [11] to advanced [12], and intelligent [13], [14].

In recent years, the best results in the control of TRAS have been achieved by using various modifications of MPC. Authors of the papers in [15], [16] deal with MPC design for achieving positioning or trajectory tracking for TRAS in coupled form by using linearized models. These solutions provide a simplification of the control problem to a series of direct matrix algebra calculations that are fast and robust. On the other hand, the main shortcoming of using such linearized models in the vicinity of some operating point for highly nonlinear plants is their inadequate accuracy for

Manuscript received 4 February, 2021; accepted 26 March, 2021.

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia and by the Science Fund of the Republic of Serbia under Grant No. 6527104, AI-Com-in-AI.

changed operating conditions. This problem can be partially mitigated either by recalculating and changing linearized models on the fly (adaptive MPC) or by having a set of predefined models and switching between already designed different linear MPC controllers (gain-scheduled MPC). One interesting approach is also offered in [17] wherein an online linearization of the nonlinear TRAS mathematical model is used both for transforming the optimization problem into a convex one and for optimal estimation of the states.

The other set of solutions deals with the insufficient accuracy of MPC with linearized models by using nonlinear models directly in the control application at a price of significantly increased computational burden. The most widely used generic nonlinear models are those based on neural networks (NN) [18], as it is well known that NNs can approximate any nonlinear function to arbitrary high accuracy. NNs belong to the black box modelling methods, where we use only input-output signals for training, while the knowledge of the physical principles of a modelled plant and the solution of possibly complicated set of mathematical equations are not required. Radial basis function NN was successfully applied in MPC of dissolved oxygen concentration in a wastewater treatment process in [19]. Similar approach can be found in [20], where a nonlinear auto-regressive moving average with exogenous inputs (NARMAX) model, implemented as multilayer NN, was applied in the nonlinear MPC of piezoelectric actuators.

Finding the optimal architecture and setting the initial values of NN parameters can be a very tedious process where we need to find the right measure of the trade-off between accuracy and network complexity [21]. NN complexity directly affects both the training time and real time calculations of the network’s outputs essential for successful MPC. The solution for these drawbacks, proposed in this paper, is to use a special type of networks (known as polynomial neural networks [22]) for modelling of the plant. These are single-layer neural networks based on orthogonal functions, where the desired output accuracy determines the required number of processing elements. Variations of Legendre orthogonal polynomial basis demonstrated excellent performance in the approximation of arbitrary functions in the sense of convergence time and approximation error due to their natural in-build optimality relative to the basis made of other types of functions [23], [24]. These orthogonal polynomials were already successfully applied in the design of very efficient tools for modelling [25] and control [26] of dynamic systems.

In this paper, TRAS was modelled by adaptive orthogonal polynomial network (AOPN) based on generalized Legendre quasi-orthogonal polynomials. These polynomials are specifically tailored for the application in the modelling of complex dynamical systems with time-varying behaviour. Variable factors incorporated inside the orthogonal functions enable the adaptivity of the designed models to the ever-changing operating environment. In addition, thanks to the single-layer structure and less model parameters, AOPN models demand a lot less calculation during determining outputs compared to their classical NN competitors. The main idea tested here is that the incorporation of these newly designed models into MPC structure could provide 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 the other MPC approaches.

The reminder of the paper is structured as follows. Section II explains the basic principles of classical MPC, as well as the proposed modifications. Section III describes the generalized Legendre quasi-orthogonal functions and designing of the general AOPN model. The process of modelling twin-rotor aero-dynamic system by using the developed polynomial network is explained in detail in Section IV. Performances of the designed MPC are then compared to the other state-of-the-art control algorithms in Section V. Finally, Section VI concludes the paper.

II. Model Predictive Control

The characteristics of the general MPC-based strategy are given in Fig. 1. The predictive future outputs of the plant (controlled system), calculated using the model within the prediction horizon N_p are denoted as y_{k+j} for j = 1, ..., N_p. These future outputs are the result of the past outputs or equivalently the value of the state at time k and the control signals u_{k+j} for j = 0, ..., N_c, where N_c represents the control horizon. Optimal future control signal u_{k+j}, which should be applied to the plant, is obtained by minimization of some objective function J (usually in quadratic form), with the goal to keep the plant output as close as possible to the given reference trajectory x_{k+j}. When the optimal sequence of control signals is calculated, only the first sample is applied to the plant. The same optimization process is repeated at every time instant. This concept, known as the receding horizon method, is used in all controllers based on MPC with numerous variations depending on the nature of the controlled system [1], [2].

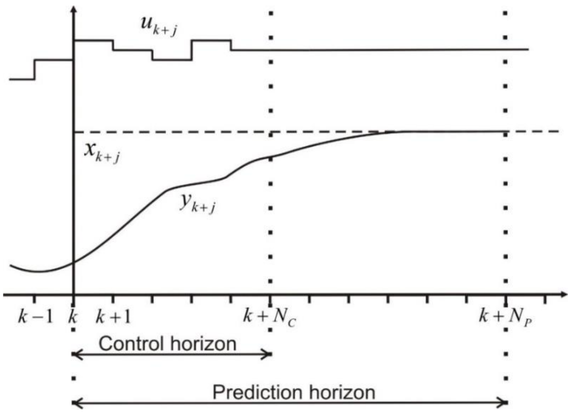

Fig. 1. The general MPC-based strategy.

Adapted MPC structure applied in this paper is shown in Fig. 2. Model predictive controller consists of an orthogonal polynomial model and the optimization block. The first step is to determine the plant model in the form of a polynomial neural network, and this process will be explained in detail in Section III. Although the model is trained offline by using previously recorded sets of input-output plant data for training, it also incorporates an adaptive factor δ dependant on the current plant conditions, which enables online adaptiveness of the model to the ever-changing operating environment. This mechanism will be also elaborated in the next section.

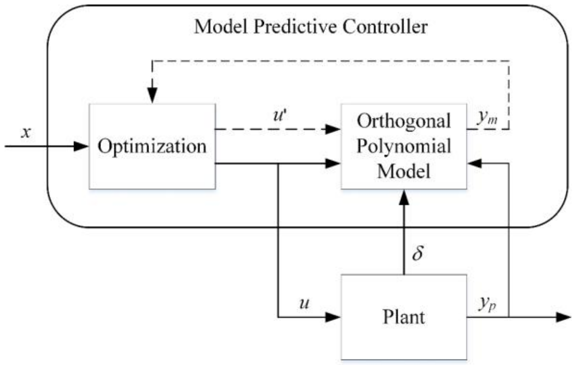

Fig. 2. Model predictive controller.

Orthogonal polynomial model, trained in this way, is then embedded into the MPC controller, where the model uses the currently applied tentative control input *u*′, the previous values of the actually used control inputs *u*, as well as the previous plant outputs y_p to calculate the possible future values of the plant output y_m. Then these values are used by the optimization program, together with the desired response (trajectory) *x*, to determine the next optimal control value by minimizing the objective function. In the case of TRAS control in this paper, the objective function was calculated as the weighted squared sum of the predicted errors and control signal increments, emphasizing not only the small tracking error, but also the smooth control. When the optimal value of the control signal is determined, it is then applied to the plant and the process is repeated for each time sample.

III. ADAPTIVE MODEL BASED ON ORTHOGONAL POLYNOMIAL NETWORK

The property of orthogonal polynomials to provide the optimal approximation of arbitrary functions in the sense of the number of addends needed in the approximation sum to reach the desired approximation accuracy, compared to other types of basis functions, is already well known [27]. Authors of this paper have previously designed some new forms (generalizations) of orthogonal polynomials and developed a comprehensive mathematical framework around them [23], [24]. Newly developed orthogonal functions were successfully applied in modelling [25] and control of dynamic systems [13], [26]. One form of Legendre polynomials, particularly interesting for control systems applications, will be considered here. These polynomials, labelled as quasi-orthogonal, represent a generalization of the classical Legendre polynomials and can be defined by a polynomial sequence

$$P_n^{(k,\delta)}\left(x\right)=\sum_{i=0}^{n}A_{n,i}^{k,\delta}x^i, \tag{1}$$

where

$$A_{n,i}^{k,\delta}=\left(-1\right)^{n+i+k}\frac{\prod_{j=1}^{n-k}\left(i+j\delta\right)}{i!\left(n-i\right)!}. \tag{2}$$

Parameter *k* marks the order of quasi-orthogonality, and previous research [23]–[25] proved that the best results in the modelling of dynamical systems are achieved for *k* = 1, 2. The main advantage of generalized Legendre polynomials defined in such a way lies in the adaptive factor *δ*. That factor has a value very close to one (*δ* ≈ 1) and it enables a small perturbation of different polynomials inside the sequence, making their integral of the inner product not zero anymore (definition of classical orthogonality), but rather some constant very close to zero. In such a way, we can use the adaptive factor *δ* to model the operating of systems in real-world conditions, time-varying behaviour, and uncertainties due to wearing over time or environmental changes. Further information relating generalized orthogonal polynomials, including definitions and derived mathematical relations, can be found in [23], [24].

For example, here are the first few second-order generalized quasi-orthogonal Legendre polynomials (*k* = 2) in the sequence defined by (2) and (3)

$$\begin{cases} P_1^{(2,\delta)}\left(x\right)=\left(1-\delta^2\right)x, \\ P_2^{(2,\delta)}\left(x\right)=\left(\delta+2\right)x^2-\left(\delta+1\right)x, \\ P_3^{(2,\delta)}\left(x\right)=\frac{\left(\delta+3\right)}{6}x^3-\frac{\left(\delta+2\right)}{2}x^2+\frac{\left(\delta+1\right)}{2}x-\frac{\delta}{6}, \\ P_4^{(2,\delta)}\left(x\right)=\frac{\left(\delta+2\right)\left(\delta+4\right)}{12}x^4-\frac{\left(\delta+3\right)\left(2\delta+3\right)}{6}x^3+ \\ +\frac{\left(\delta+1\right)\left(\delta+2\right)}{2}x^2-\frac{\left(\delta+1\right)\left(2\delta+1\right)}{6}x+\frac{\delta^2}{12}\ . \end{cases} \tag{3}$$

These quasi-orthogonal functions will be the basis for designing models of dynamical systems in the form of adaptive orthogonal polynomial neural networks shown in Fig. 3. Such a network generates function, a sum of weighted Legendre (or some other kind) polynomials, capable of approximating an arbitrary function. Thereby the orthogonal polynomial expansion guaranties the natural optimality in the sense of approximation accuracy and shorter convergence time compared to the basis made of other functions. The expansion also incorporates additional adaptivity thanks to the built-in adaptive factor (*δ*). Block marked as Legendre expansion generates quasi-orthogonal Legendre polynomials (P_i) based on the current and time-delayed valued of the inputs and outputs of the plant (*a* previous input values and *b* previous output values). Past instances are provided by the blocks Tapped Delay Line (TDL), whose role is to buffer previous values of signals.

The AOPN model depicted in Fig. 3 provides model of the plant in the form of approximation function

$$y_m\left(X,y_p,\delta\right)=f\left(\sum_{i=1}^{n}w_i p_i\left(X,y_p,\delta\right)+b\right), \tag{4}$$

where *X* represents the vector of inputs, w_i are weights of the network, and *f* is the activation function.

Model of the considered plant is obtained after the polynomial network training, i.e., after optimizing the weights w_i ($i = 1, 2, \ldots, n$) of the network. This optimization is performed by minimizing the modelling error, which is calculated as the difference between the plant (y_p) and the model (y_m) outputs. Training algorithm applied in the modelling of TRAS was Levenberg-Marquardt or dumped least-squares method [28], [29], which is the most efficient algorithm for the training of polynomial networks [22], [30]. The algorithm is based on a gradient vector and the Jacobian matrix and uses the sum of squared errors as the performance (cost) function

$$Q = \sum_{k=1}^{m} \left(y_p(k) - y_m(k) \right)^2 = \sum_{k=1}^{m} e_k^2. \tag{5}$$

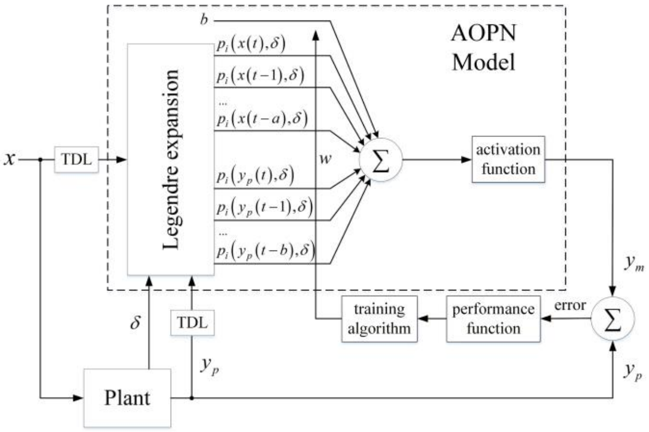

Fig. 3. Orthogonal neural network applied in the modelling of an arbitrary dynamic system.

IV. Modelling of Twin-Rotor Aero-Dynamic System

Laboratory twin-rotor aero-dynamic system (TRAS) will be considered as a case study for MPC of nonlinear MIMO systems based on AOPN model. The system [10] is controlled from a PC with software operating under MATLAB/Simulink environment. Main components of TRAS can be seen in Fig. 4. Position of the system with two degrees of freedom (azimuth and pitch angles labelled as α_h and α_v, respectively) is controlled by two DC motors representing the drives for the main and tail rotors. These two propellers, which are located at the ends of the beamwith a counterbalance pivoted on its base, enable rotation of the beam in horizontal and vertical planes. Embedded sensors (encoders for angular positions and tachogenerators for angular velocities) are responsible for measuring state variables in real time.

Tracking of the desired trajectory is achieved by calculating and applying adequate control inputs, i.e., supply voltages for DC motors labelled as U_h and U_v. Variations in these control voltages result in different rotational speed of the corresponding propeller and change in the position of beams. However, due to significant cross-couplings, each rotor actually affects both position angles. Cross-coupling, together with some other TRAS features like high-order nonlinear dynamics, susceptibility to external disturbances, parameter variations, and unwanted induced vibrations, make TRAS an extremely difficult system to model (or control) by conventional white box first-principle methods, but also to black-box methods based on identification by using input/output data sets.

AOPN modelling (Fig. 3) can be used for laboratory TRAS after adapting the general structure for two-input (U_h, U_v), two-output (α_h, α_v) system. Legendre expansion generates generalized quasi-orthogonal polynomials according to (1) and (2) based on current, but also buffered values (blocks TDL) of previous input/output samples. In this concrete case, one previous time instance was used for inputs and two for output signals. Presence of both inputs and outputs in polynomial development guaranties adequate modelling of existing cross-couplings in the system. Sigmoidal function $f(x) = 1/(1 + e^{-x})$ was applied as the activation function. As already stated, Levenberg-Marquardt algorithm was used for training of the network, i.e., for determining the optimal values of network weights (w_i) and bias (b), based on modelling error - difference between the measured outputs of the laboratory TRAS and those obtained by the AOPN model. Initial modelling was performed with the nominal value of parameter δ set to one, but in the following experiments, this parameter was perturbated to simulate models' adaptivity to environmental changes, measurement uncertainties, and sensed disturbances.

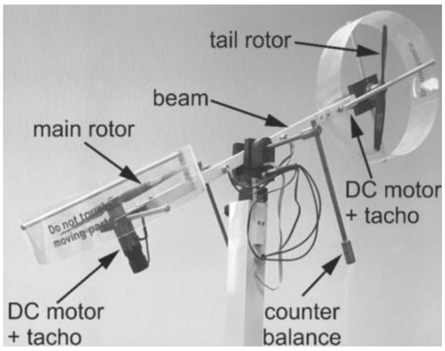

Fig. 4. Laboratory TRAS system.

The AOPN model is trained on experimental input/output data sets. The same input signals were applied both to the plant and AOPN models: U_h - square wave with 0.15 amplitude and 1/50 [Hz] frequency; U_v - sine wave with the amplitude of 0.25 and frequency of 1/50 [Hz]. These signals were first normalized to a range of [-1, +1] corresponding to a voltage range of [-24 V, +24 V]. Signals in the input/output data sets were sampled with a period of 0.01 seconds with an overall duration of the excitation of 90 seconds. AOPN model was trained with six terms in polynomial expansion (4) because that number makes the best trade-off between model accuracy and training time (network complexity), as can be seen in Table I, where root-mean-square error (RMSE) was calculated as

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(\left(y_{hp}(k) - y_{hm}(k) \right)^2 + \left(y_{vp}(k) - y_{vm}(k) \right)^2 \right)}. \tag{6}$$

TABLE I. PERFORMANCES OF AOPN MODELS WITH DIFFERENT NUMBER OF TERMS IN THE ORTHOGONAL EXPANSION.

Number of terms in the orthogonal expansion	3	4	5	6	7
Training set RMSE	9.9739	4.8690	2.1492	0.8231	0.5636
Training time (s)	4.22	6.52	11.02	24.05	78.84

V. MODEL PREDICTIVE CONTROL OF TWIN-ROTOR AERO-DYNAMIC SYSTEM

Validation of the proposed control algorithm, i.e., Model Predictive Control based on Adaptive Orthogonal Polynomial Networks (AOPNMPC), given in Fig. 2, was performed by comparing the performances with the other two control strategies already proven to be suitable for TRAS. One complex control structure, labelled as Orthogonal Endocrine Intelligent Controller (OEIC), is presented in [13] and has a form of an intelligent hybrid controller with two main components: orthogonal endocrine neural network and adaptive neuro-fuzzy inference system (ANFIS). The second controller is a classical MPC controller designed with linearized model of TRAS given in [15].

Optimization of the MPC was done by default functions provided by Matlab for backtracking search best suited to use with the quasi-Newton optimization algorithms. Polynomial model was described in Matlab similarly to the neural state-space model presented in [18]. Objective function had the following form

$$J = \sum_{j=1}^{N_p} \left[x(k+j) - y_m(k+j) \right]^2 +$$

$$+\rho \sum_{j=1}^{N_c} \left[u'(k+j-1) - u'(k+j-2) \right]^2 , \qquad (7)$$

and it emphasized not only minimizing the tracking error, but also smoothing the control of TRAS and avoiding sharp turns. The importance (contribution) of each of these two factors is controlled by an adjustable parameter ρ. All experiments were executed with $\rho = 0.25$.

Reference tracking positions were: azimuth - square wave with amplitude of 0.4 rad and frequency of 1/50 Hz and pitch - sine wave with amplitude of 0.25 rad and frequency of 1/60 Hz. Duration of the experiments was 90 seconds with a sample time of 0.01 s. Obtained results for azimuth and pitch angles for all three applied controllers (AOPNMPC, OEIC, MPC) are given in Figs. 5 and 6.

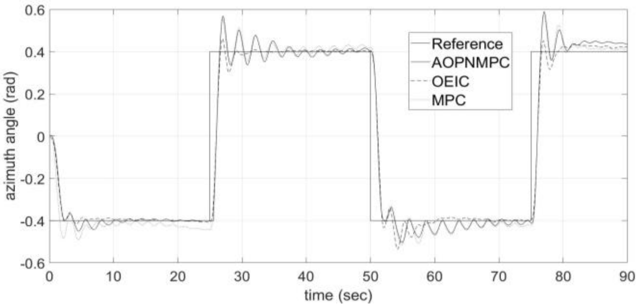

Fig. 5. Azimuth angles for different controllers.

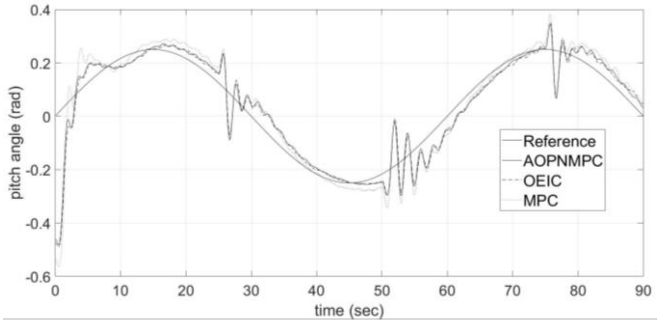

Fig. 6. Pitch angles for different controllers.

We can see from the figures that all three controllers fulfil their purpose in tracking the desired trajectory relatively well, but the real difference can be detected if we calculate root-mean-square-error

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(\left(y_{hp}(k) - x_{hr}(k) \right)^2 + \left(y_{vp}(k) - x_{vr}(k) \right)^2 \right)}, \qquad (7)$$

where x_{hr} and x_{vr} are the inputs - referent azimuth and pitch trajectories, y_{hr} and y_{vr} are outputs - obtained trajectories for a given controller, and N = 9000. Results for all three controllers, together with the training times for AOPNMPC and OEIC, are given in Table II.

TABLE II. PERFORMANCES FOR DIFFERENT CONTROLLERS.

Controller	**RMSE**	**Training time**
AOPNMPC	0.176	8.2
OEIC	0.151	25.4
MPC	0.52	-

Plain MPC has the worst tracking accuracy because of the shorter prediction horizon due to a lot of online recalculation. AOPNMPC has much better tracking accuracy, but still worse than OEIC, although with shorter network training time. On the other hand, OEIC achieves its best accuracy with enormous controller complexity and cost [13]. Real strength of AOPNMPC can be noticed if we artificially introduce disturbances into the original nominal system to imitate environmental changes, measurement errors or occurred uncertainties. This was achieved by programming the artificial measurement error (noise) into feedback signals coming from position sensors (encoders) responsible for reporting azimuth and pitch angles. Two more sets of experiments were performed with the introduced change in nominal system response with a noise to signal ratio (SNR) of 1 % and 3 %. We can see from Table III that OEIC and MPC cannot adjust well to the changes without either new network training or finding new state-space matrixes because their RMSE significantly increases. On the other hand, AOPNMPC has incorporated the measure of variations δ directly inside its AOPN model, so this controller does not need to be trained again. The only modification, which has to be implemented, is to set δ to

1.01 and 1.03 for changes of 1 % and 3 %, respectively, and the controller will keep its tracking accuracy.

TABLE III. TRACKING ACCURACY IN CASE OF DISTURBANCES.

Controller	**RMSE (SNR = 1 %)**	**RMSE (SNR = 3 %)**
AOPNMPC	0.185	0.144
OEIC	0.331	0.46
MPC	0.935	1.772

VI. CONCLUSIONS

This paper considers a new approach to model predictive control of nonlinear multiple-input multiple-output systems. The approach implies an innovative design of the model of the plant based on adaptive orthogonal polynomial networks, which uses the basis of specially tailored generalized quasi-orthogonal functions. Such models take advantage of the natural superiority of orthogonal polynomials, compared to other functions, to approximate arbitrary functions with better accuracy for a lower number of terms in expansion. Additionally, thanks to the incorporated adaptive factor, quasi-orthogonality enables these models to easily accommodate to time-varying behaviour and perturbations occurring during the work of real systems.

AOPN models are similar to the classical neural networks, but much simpler to design, with a single-layer structure and weights actually representing coefficients in polynomial expansion. Optimal coefficients can be determined by a training process (in this case, Levenberg-Marquard), during which we try to minimize the modelling error (difference between the plant and the model outputs for the same applied inputs). AOPN model can be embedded into the MPC controller, where it is used for calculating possible plant outputs based on the currently applied tentative control input, previous values of actually used control inputs, and previous plant outputs. Matlab optimization routine was applied in choosing the optimal control based on the desired and possible outputs, whereby the objective function was designed in such a way to take into account not only the small tracking error, but also to smooth the control as much as possible.

As a case study for validation of the described control approach, twin-rotor aero-dynamic system was chosen as a suitable representative of complex nonlinear MIMO systems with cross-couplings very susceptible to unwanted vibrations and external disturbances. Comparative analysis of the proposed AOPNMPC controller was performed with two other state-of-the-art controllers: a classical MPC with a linearized model recalculated online and a complex hybrid intelligent controller (OEIC) built as a combination of orthogonal endocrine neural network and ANFIS. The new controller demonstrated satisfactory performances in conducted experiments in pairs with a much more complex and expensive OEIC and much better accuracy than the classical MPC. The superiority of the novel controller becomes even more obvious when we introduced disturbances into the original nominal system to imitate environmental changes and measurement errors, thanks to built-in adaptability of AOPN model.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

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

[2] Y. Xi and D. Li, *Predictive Control: Fundamentals and Developments*. Wiley Online Library, USA, 2019. DOI: 10.1002/9781119119593.

[3] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson College Div., 2001.

[4] T. Geyer, Model Predictive Control of High Power Converters and Industrial Drives. Wiley, London, 2016. DOI: 10.1002/9781119010883.

[5] V. Slapak, K. Kyslan, and F. Durovsky, “Position controller for PMSM based on finite control set model predictive control”, *Elektronika ir Elektrotechnika*, vol. 22, no. 6, pp. 17–21, 2016. DOI: 10.5755/j01.eie.22.6.17217.

[6] C. Bordons, F. Garcia-Torres, and M. Ridao, *Model Predictive Control of Microgrids*. Springer International Publishing, Switzerland, 2020. DOI: 10.1007/978-3-030-24570-2.

[7] E. F. Camacho and C. B. Alba, *Model Predictive Control*. Springer-Verlag London, 2007. DOI: 10.1007/978-0-85729-398-5_6.

[8] M. Pastor and J. Dudrik, “Comparison of MPC and PI controller for grid-connected cascade inverter”, *Elektronika ir Elektrotechnika*, vol. 20, no. 6, pp. 46–50, 2014. DOI: 10.5755/j01.eee.20.6.7266.

[9] H. Efheij, Model Predictive Control and PID Controller in Real Time System: Design, Implementation and Comparison of MPC and PID Controller in Real Time Process Control System. LAP LAMBERT academic publishing, 2020. DOI: 10.1109/STA.2019.8717271.

[10] *Two rotor aero-dynamical system (TRAS), User’s manual for MATLAB*, Inteco, Poland, 2009. [Online]. Available: http://www.inteco.com

[11] P. S. Kannan and P. Sheenu, “PID control of twin rotor MIMO system”, *International Journal of Advanced Research in Basic Engineering Sciences and Technology*, vol. 3, no. 23, pp. 1–7, 2017.

[12] Ch.-W. Tao, J.-Sh. Taur, Y.-H. Chang, and Ch.-W. Chang, “A novel fuzzy-sliding and fuzzy-integral-sliding controller for the twin rotor multi-input-multi-output system”, *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 893–905, 2010. DOI: 10.1109/TFUZZ.2010.2051447.

[13] M. B. Milovanović, D. S. Antić, M. T. Milojković, and M. D. Spasić, “Adaptive control of nonlinear MIMO system with orthogonal endocrine intelligent controller”, *IEEE Transactions on Cybernetics*, to be published. DOI: 10.1109/TCYB.2020.2998505.

[14] N. M. H. Norsahperi and K. A. Danapalasingam, “Particle swarm-based and neuro-based FOPID controllers for a twin rotor system with improved tracking performance and energy reduction”, *ISA Transactions*, vol. 102, pp. 230–244, 2020. DOI: 10.1016/j.isatra.2020.03.001.

[15] R. G. Tiwalkar, Sh. S. Vanamane, S. S. Karvekar, and Sh. B. Velhal, “Model predictive controller for position control of twin rotor MIMO system”, in *Proc. of the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India, 2017, pp. 952–957. DOI: 10.1109/ICPCSI.2017.8391852.

[16] U. M. Nath, C. Dey, and R. K. Mudi, “Controlling of twin rotor MIMO system (TRMS) based on multivariable model predictive control”, in *Circuits and Communication Systems (NCCS). Lecture Notes in Electrical Engineering*, vol. 692. Springer, Singapore, 2021. DOI: 10.1007/978-981-15-7486-3_44.

[17] D.-A. Duțescu, M.-B. Radac, and R.-E. Precup, “Model predictive control of a nonlinear laboratory twin rotor aero-dynamical system”, in *Proc. of the IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2017)*, Herl’any, Slovakia, 2017, pp. 000037–000042. DOI: 10.1109/SAMI.2017.7880339.

[18] A. Czajkowski and K. Patan, “Model predictive control of the two rotor aero-dynamical system using state space neural networks with delays”, in *Intelligent Systems in Technical and Medical Diagnostics. Advances in Intelligent Systems and Computing*, vol. 230. Springer, Berlin, Heidelberg, 2014. DOI: 10.1007/978-3-642-39881-0_9.

[19] H.-G. Han, L. Zhang, Y. Hou, and J.-F. Qiao, “Nonlinear model predictive control based on a self-organizing recurrent neural network”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 402–415, 2016. DOI:

10.1109/TNNLS.2015.2465174.

[20] L. Cheng, W. Liu, Z.-G. Hou, J. Yu, and M. Tan, "Neural-network-based nonlinear model predictive control for piezoelectric actuators", *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7717–7727, 2015. DOI: 10.1109/TIE.2015.2455026.

[21] 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 Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 228–240, 2012. DOI: 10.1109/TII.2012.2187914.

[22] S.-K. Oh, W. Pedrycz, and B.-J. Park, "Polynomial neural networks architecture: Analysis and design", *Computers & Electrical Engineering*, vol. 29, no. 6, pp. 703–725, 2003. DOI: 10.1016/S0045-7906(02)00045-9.

[23] M. Milojković, S. Nikolić, B. Danković, D. Antić, and Z. Jovanović, "Modelling of dynamical systems based on almost orthogonal polynomials", *Mathematical and Computer Modelling of Dynamical Systems*, vol. 16, no. 2, pp. 133–144, 2010. DOI: 10.1080/13873951003740082.

[24] D. Antić, B. Danković, S. Nikolić, M. Milojković, and Z. Jovanović, "Approximation based on orthogonal and almost orthogonal functions", *Journal of the Franklin Institute*, vol. 349, no. 1, pp. 323–336, 2012. DOI: 10.1016/j.jfranklin.2011.11.006.

[25] 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", *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, pp. 091013-1–091013-6, 2015. DOI: 10.1115/1.4030758.

[26] S. Lj. Perić, D. S. Antić, M. B. Milovanović, D. B. Mitić, M. T. Milojković, and S. S. Nikolić, "Quasi-sliding mode control with orthogonal endocrine neural network-based estimator applied in anti-lock braking system", *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 754–764, 2016. DOI: 10.1109/TMECH.2015.2492682.

[27] G. Szegö, *Orthogonal Polynomials*. American Mathematical Society, Colloquium Publications, 1975.

[28] B. M. Wilamowski and H. Yu, "Improved computation for Levenberg-Marquardt training", *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 930–937, 2010. DOI: 10.1109/TNN.2010.2045657.

[29] J. J. de Rubio, "Stability analysis of the modified Levenberg-Marquardt algorithm for the artificial neural network training", *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020. DOI: 10.1109/TNNLS.2020.3015200.

[30] 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 Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1631–1642, 2016. DOI: 10.1109/TNNLS.2014.2360879.