AKADÉMIAI KIADÓ

International Review of Applied Sciences and Engineering

11 (2020) 3, 251–260

DOI:
10.1556/1848.2020.00090

ORIGINAL RESEARCH PAPER

Design of hybrid Neuro-Robust deadbeat controller for higher order linear systems based on optimized mixed reduction method

EKHLAS H. KARAM[1]*, NOOR S. ABDUL-JALEEL[2] and BASMA J. SALAH[1]

[1] Department of Computer Engineering, Mustansiriyah University, Baghdad, Iraq

[2] Department of Electrical Engineering, Mustansiriyah University, Baghdad, Iraq

Received: January 6, 2020 • Accepted: May 24, 2020
Published online: October 8, 2020

ABSTRACT

The control of higher order linear system is one of the main fields of research area that has been studied for decades because of the difficulty in designing a controller for such systems. One of the best approaches to solve this problem is by reducing the order of the system into a second orders, based on this reduction many approaches can be proposed for controlling the higher order system, therefore many reduction methods are suggested and developed for this purpose, one of these methods is the Mixed Reduction Method (MRM). The first contribution of this paper is to improve the efficiency of MRM by using a flower optimization algorithm.

The second contribution of this paper lies in proposing a hybrid Neuro-Robust deadbeat controller using Matlab facilities to control higher order linear systems based on the optimized MRM. Where the robust deadbeat control algorithm is combined with a modified adaptive radial basis neural network to improve the robustness and efficacy of the deadbeat controller, which is partially lost when designing this controller for the higher order based on model reduction. The suggested radial basis function neural network has a simple design. The proposed control scheme assures the stability of the overall closed loop-controlled system; therefore, it can be applied to control any linear higher order systems. Results of different simulation examples show the efficiency of the proposed hybrid controller (Neuro-robust deadbeat) in tracking different reference signals compared to the robust deadbeat controller.

KEYWORDS

deadbeat, radial basis neural network, higher order linear systems, model reduction, flower optimization algorithm

1. INTRODUCTION

One of the important topics in control engineering is how to reduce higher order systems [1, 2], where the objective is to design a controller that has a lower order, while preserving effectiveness when used with higher order systems [1].

Higher order systems can be found in multifarious industries such as oil, cement, chemical, pharmaceutical, aircraft system, atomic nuclear plant, flexible robot manipulator [3], quadrotor with a variable degree of freedom (DOF) [4], fuel injector and spark timing of automobiles, etc. [5]. Nonlinear systems are usually linearized at different operating points, and appear to be higher order systems [6, 7]. In such a case, higher order system can be obtained using the first principle method or the finite-element model [8, 9]. Due to poor-conditioning, time and memory limitation, most of these models may not be suitable for many applications such as improving the design of control systems or analyzing them, especially in large-scale systems [10, 11]. Due to the complexity and high cost, it is difficult to design a precise control of large-scale control systems. Therefore, many researches proposed suitable controllers based on model order reduction, the reduction methods in most of these

*Correspondng author.
E-mail: ek_karam@yahoo.com

researches are: Routh approximation, Pade approximation, minimizing and rounding the model by reducing the flow of errors, the principal component analysis method, and the model reduction using impulse/step error minimization, in all these approaches, the higher order system is approximated to a lower order one by using model reduction, through basic step-to-step method for element analysis [12] and balanced-truncation method [13]. Recently, several reduction methods have been suggested based the optimization algorithms such as in [14–17]. These algorithms consist of finding a solution to problems that may minimize or maximize costs. Depending on model reduction methods, many controllers can be simply designed to control the higher order system such as PID [18, 19], Internal Model Controller (IMC) [20], Robust Dead-beat Control (RDBC) [21, 22].

The RDBC is one of the common controllers that has been designed to control the lower and higher linear systems. This controller has been successfully implemented on first and second order systems. It was shown from the result of the tested system that the settling time increases when the order of the system is increased, in addition to the difficulty in obtaining the controller parameters [22]. So the main challenges in this paper are to simplify the calculation of the RDBC parameters as well as achieving the predefined specifications and hence reduce the settling time of the overall control system. Therefore in this paper, the model order reduction method, called Mixed Reduction Method (MRM) is first optimized by Flower Pollination Algorithm (FPA), then the hybrid neuro-deadbeat control scheme is proposed to control different higher order systems, where the RDBC is combined with a simple adaptive modified radial basis neural network, to enhance the robustness and the performance of the RDBC in tracking any set point. FPA is used here because this algorithm is adaptable, simple and efficient, it is established to balance between exploitation and exploration by a variable called switching probability.

The rest of this paper is organized as follows: Section 2, presents the model order MRM. Section 3 explains the suggested optimized MRM by using flower pollination algorithm. Section 4 illustrates the suggested RDBC design, while Section 5 shows the proposed improved RDBC by using simple modified radial neural network. Section 6 addresses the simulation results of the proposed controller for different test examples. Finally, summary and conclusions are presented in Section 7.

2. MODEL ORDER BY MIXED REDUCTION METHOD (MRM)

One of the techniques utilized for order reduction is the MRM, which is used in numerical simulations to reduce the complexity of the mathematical models [18]. Due to the complexity and high dimensionality, most modern mathematical models are challenging, with regard to numerical simulations. In this method, the approximation of the original model is calculated by subtracting the space dimensions of the state accompanying the degrees of freedom of the model. This allows for a reduced model to be evaluated with less precision, yet in much less time [18]. To get the kth reduced order models, two steps must be applied:

Step-1: Determine the denominator using the modified pole clustering technique [23]. The same procedure is used for the real and the imaginary parts of the complex conjugate poles, and the modified cluster center is implemented as $\emptyset_{ej} = A_{ej} \pm jB_{ej}$, where: $\emptyset_{ejj} = A_{ej} + jB_{ej}$ and $\emptyset_{ej} = A_{ej} - jB_{ej}$. The algorithm flow chart shown in Fig. 1 can be used to find a modified cluster center.

Step-2: The numerator of the reduced model is determined by using the technique of error minimization. This is achieved by equating the higher order transfer function to the form of reduced order transfer function as given below:

$$G_n(s) = \frac{N_{h(s)}}{D_h(s)} = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_{m-1}s^{m-1} + a_m s^m}{b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1}s^{n-1} + b_n s^n} = \frac{d_0 + d_1 s + d_2 s^2 + \cdots + d_{q-1}s^{q-1} + d_q s^q}{e_0 + e_1 s + e_2 s^2 + \cdots + e_{r-1}s^{r-1} + e_r s^r} \quad (1)$$

By cross multiplying Eq.(1) and comparing terms of similar powers of (s) for both sides:

$$\begin{aligned} a_0 e_0 &= b_0 d_0 \\ a_0 e_1 + a_1 e_0 &= b_0 d_1 + b_1 d_0 \\ a_0 e_2 + a_1 e_1 + a_2 e_0 &= b_0 d_2 + b_1 d_1 + b_2 d_0 \\ a_m e_r &= b_n d_q \end{aligned} \quad (2)$$

The unknown numerator coefficients d_0, d_1,. . .., d_q can be

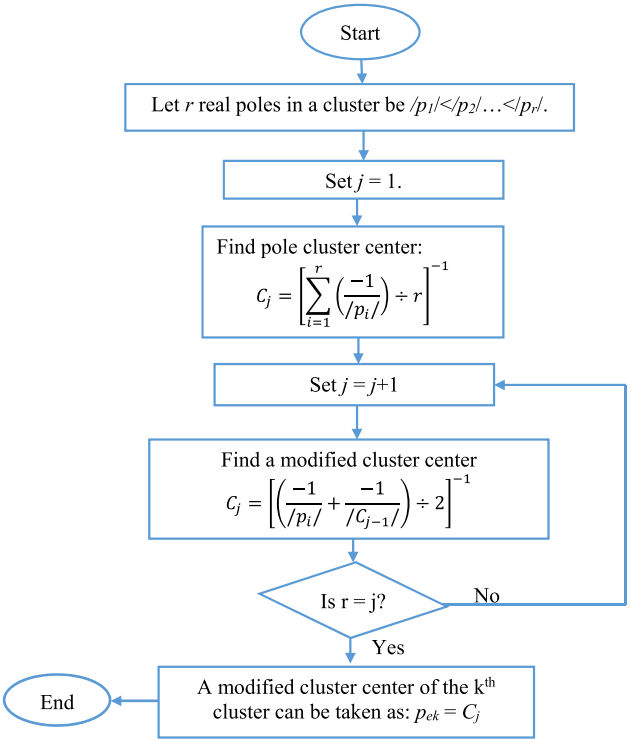

Fig. 1. The algorithm of modified cluster center

found by solving the above equations, and the reduced order numerator can be written as [18]:

$$N_r(s) = d_0 + d_1 s + d_2 s^2 + \cdots + d_{q-1} s^{q-1} + d_q s^q \quad (3)$$

In this paper, steps 1 and 2 are used to reduce the order of the higher order equations to second order equations that has the following form:

$$G_r(s) = \frac{N_r(s)}{D_r(s)} = \frac{d_0 + d_1 s}{e_0 + e_1 s + e_2 s^2} \quad (4)$$

or

$$G_r(s) = \frac{N_r(s)}{D_r(s)} = \frac{n_1 s + n_2}{s^2 + m_1 s + m_2} \quad (5)$$

where $n_1 = \left(\frac{d_1}{e_2}\right)$, $n_2 = \left(\frac{d_0}{e_2}\right)$, $m_1 = \left(\frac{e_1}{e_2}\right)$, and $m_2 = \left(\frac{e_0}{e_2}\right)$.

3. FLOWER POLLINATION ALGORITHM (FPA)

The FPA was proposed by Yang in 2012 [24]. It is inspired by the process of flowers pollination in zoology where flowers' pollens are spread to another plant for reproduction. Sometimes the flowers pollinate with the aid of biotical using several animals or insects. Others pollinate via water or wind. Pollination [24] can be accomplished by cross-pollination or self-pollination, depending on whether it involves flowers from different plants or flowers from the same plant. This method was developed to tackle optimization problems of global domain with multiple objectives and multiple varied principles. This technique requires the participation of several individuals called pollinators. Self (Biotic) pollination depends on local pollination operations while the global pollination operations are used with cross-pollination. The technique supposes that a solution is a single pollen. Pollination technique can be employed to motivate members' reproduction, which includes vector estimation and candidate solution. The size of the step in the next epoch is based on the pollination intensity. The Levy flights equation is used to simulate the pollinator movement for a global pollination [25]. To evaluate the pollination, the candidate vectors are selected randomly from a uniform distribution. If the selected new member is optimal, it is re-trained, and updated the worst members, however, one ignored. This iterative process is continued until the optimum member is obtained. Furthermore, this algorithm has more points to be investigated, such as non-homogeneity, multiple dimensionality and complexity of time [26]. The FPA depends on the following four rules [27]:

(i) Biotic and Levy flights can be used for carrying pollinators, for global pollination, and Cross-pollination.
(ii) Self -pollination and abiotic-pollination are used for local pollination.
(iii) Pollinators, like insects, can increase flower firmness. This role is equivalent to cloning probability, which refers to the similarity of two implicated flowers.
(iv) The switching between local and global pollination can be controlled by interaction probability $P_n \in [0, 1]$.

To formulate the algorithm update procedure, the assumptions above are required to be translated into computable steps. Such as, the flower pollen gametes are moved by birds or insects for the global pollination, the birds or insects can often move and fly for a long distance. Rule (i) and the flower firmness can be illustrated mathematically as in Eq. (6) below [28]:

$$X_i^{t+1} = X_i^t + \beta L(\beta)\left(X_i^t - B\right) \quad (6)$$

where X_i^t is the ith pollen (solution vector X_i) at the tth epoch. B is the current optimal solution found among all solutions at the current epoch. Here, β is a scaling factor which controls the step size. $L(\beta)$ is the parameter that identifies the length of the pollination.

Since insects may fly over a long range with different length of steps, the Levy flight can be used to simulate this case. Let $L > 0$ from Levy equation:

$$L \sim \frac{\beta\, \Gamma(\beta)\sin(\beta)}{\pi} \frac{1}{S^{1+\beta}} \quad S \gg S_0 > 0 \quad (7)$$

Here, $\Gamma\beta$ is the standard gamma function, and this equation is proper for large steps S > 0. Next, to simulate the local pollination, both rule (ii) and rule (iii) can be illustrated as:

$$X_i^{t+1} = X_i^t + U\left(X_J^t - X_K^t\right) \quad (8)$$

where X_J^t and X_K^t are the pollen from different flowers of the same plant species. This equation basically simulates the flower firmness in a defined neighborhood. Mathematically, if X_J^t and X_K^t are selected from the same population, this similarity becomes a local random walk if U is drawn from a uniform distribution in [0, 1]. Nevertheless, the natural actions of flower pollination may appear at all ranges; neighboring patches of flower are very similar to be pollinated by local flower pollen than those distant. To simulate this, the probability of switch corresponding to rule (iv) or the closest probability p^s of transition from local pollination to global pollination.

3.1. Optimized Mixed Reduction Method (OMRM) by using FPA

Each flower (or pollen) in FPA refers to a solution in control problem. Likewise, when employing FPA to improve the MRM for reducing the higher order system, every individual agrees to a candidate solution which is referred to as higher order factor. The first step of FPA is the initialization of the population. The initial population consists of N individuals. Each member presents a candidate solution in on N-dimensional space of solutions. The population parameters initialization and the fitness function are represented as:

$$\text{Pop} = [n_1,\ n_2,\ m_1,\ \text{and}\ m_2] \quad (9)$$

where n_1 is X_1^t, n_2 is X_2^t, m_1 is X_3^t, m_2 is X_4^t, and t is the iteration index. The Fitness function is chosen as:

$$\text{Fitness function} = \sum_{i=1}^{N} \left(y_n(i) - y_r(i)\right) \tag{10}$$

where: $y_h(i)$ and $y_r(i)$ are the original higher order system and the reduced model by MRM (which optimize by flower pollination parameters) respectively. The implementation steps of the optimized MRM by FPA are summarized in Fig. 2.

4. ROBUST DEADBEAT CONTROLLER DESIGN

Deadbeat control was originally introduced as a linear state-feedback control for discrete-time systems. It is the simplest of all performance methods, and it is the most common method applied today [29]. The overall control system provides good parametric robustness in a steady state; the control error is practically zero [30] which is the main goal of the control system to reach the desired value within specified settling time [31]. A deadbeat response should have zero steady state error ($e_{s.s}$), minimum rise time (t_r), controllable settling time (t_s), overshoot (M_p) from 0 to 2%, undershoot <2%. Also have robustness against external disturbance and parameter uncertainty [31]. However, uncertainties on the input gain remain problematic [32].

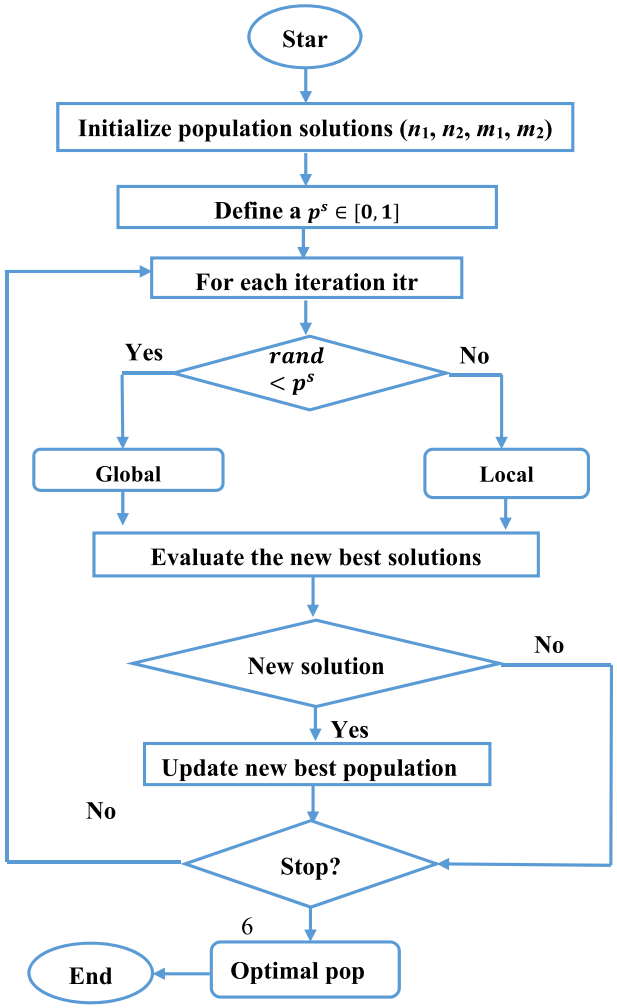

Fig. 2. The flow chart of FPA for optimizing the MRM

4.1. Deadbeat controller design for higher order system

Fig. 3 shows the basic structure of the deadbeat controller.

The main control law of this structure is:

$$u(s) = u_o(s) + u_c(s) \tag{11}$$

with

$$u_0(s) = H_2(s)y(s) = K_a y(s) \tag{12}$$

$$u_c(s) = KK_C \frac{(S^2 + Xs + Y)}{S} e_h(s) \tag{13}$$

where $u_o(s)$ is an output feedback control, while $u_c(s)$ is the portion of the control signal generated by cascade PID with series gain K, which is introduced in order to compensate the higher-order plants, to deal with uncertainties, and to be able to perform the desired specifications such as settling time, error steady state and maximum overshoot.

The error signal $e_h(s)$ is defined by;

$$e_h(s) = r(s) - H_1(s)y(s) \tag{14}$$

where $H_1(s)$ is an additional derivative controller, its structure depends on the order of $G_c(s)G_p(s)$ transfer function as shown in Table 1.

The deadbeat controller parameters can be determined by comparing the equivalent characteristic equation of closed-loop controlled system with the following desired deadbeat characteristic equation:

$$s^{n_p} + \alpha_1\omega_n s^{n_p-1} + \alpha_2\omega_n^2 s^{n_p-2} + \cdots + \omega_n^{n_p} \tag{15}$$

The α_i coefficients and the deadbeat settling time t_s of the step response corresponding to different $G_c(s)G_p(s)$ order are illustrated in Table 2.

Depending on desired settling time, the normalized frequency (ω_n) is described as follows:

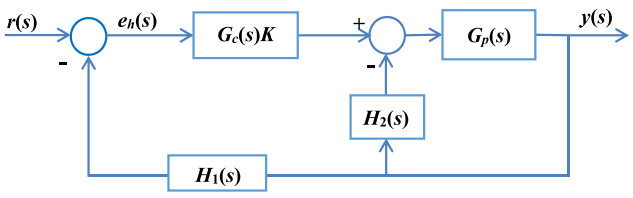

Fig. 3. Block diagram for the basic robust deadbeat control structure

Table 1. The structure of the $H_1(s)$ equation

$G_c(s)G_p(s)$ order	2	3 or 4	5
$H_1(s)$ equation	$H_1(s) = 1$	$H_1(s) = 1 + K_b s$	$H_1(s) = 1 + K_b s + K_c s^2$

Table 2. Coefficient of the desired deadbeat characteristic equation [21, 33].

$G_c(s)G_p(s)$ order	α_1	α_2	α_3	α_4	t_s
2nd	1.82	–	–	–	4.82
3rd	1.90	2.2			4.04
4th	2.20	3.50	2.8		4.81
5th	2.70	4.90	5.40	3.40	5.43

$$\omega_n = \frac{t_s}{\mu t_{ds}} \tag{16}$$

where μ is a selected positive constant between $0.8 \le \mu \le 0.95$.

4.2. Design the Robust deadbeat based on reduction model

The block diagram for the suggested simple, robust deadbeat controller based on reduction model is shown in Fig. 4 below.

where $G_r(s)$ is the reduction model (Eq. (5)) which parameters values (n_1, n_2, m_1, and m_2) are determine by the improved OMRM method. While, the G_c is only the PI controller multiplied by the series gain K:

$$G_c = \frac{K(K_c s + K_c x)}{s} = \frac{K[K_c(s+x)]}{s} \tag{17}$$

with $x = \frac{1}{T_i}$

T_i is a suitable selected integral time, while H_1 and H_2 as shown in Fig. 3, are given by:

$$H_1(s) = (1 + K_b s) \tag{18}$$

$$H_2(s) = K_a \tag{19}$$

Since K_b is a derivative gain, then it can be written as;

$$K_b = K_c T_d = K_c \frac{T_i}{4} = \frac{4K_c}{T_i} \tag{20}$$

And, since K_a is just a proportional gain, it can be determined from K_b as follows:

$$K_a = \frac{K_p}{T_d} = \frac{4K_p}{T_i} \tag{21}$$

So, the transfer function of closed loop controlled system will be:

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_r(s)}{1 + G_r(s)H_2(s) + G_c(s)G_r(s)H_1(s)} \tag{22}$$

or

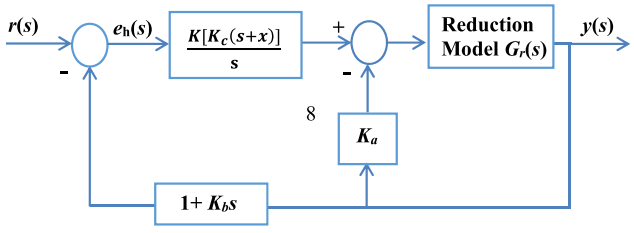

Fig. 4. Block diagram of the suggested robust deadbeat controller based on reduction model structure

$$\text{C.L.T.F} = \frac{2.394xKaKd + 2.394KaKds}{s^2 + (2.657 + 2.394KaKd)s + 2.394Ka + 2.394xKaKd} \tag{23}$$

The desired characteristic equation which will compare with the characteristic equation of the closed-loop controlled system is:

$$s^3 + \alpha_1\omega_n s^2 + \alpha_2\omega_n^2 s + \omega_n^3 \tag{24}$$

In this case, the procedure for design the robust deadbeat includes the following steps:

1. The α_1 and α_2 are taken from Table 2, while ω_n is selected according to specified t_s (Eq. (16)) then the desired characteristic equation is determined.
2. K_b is chosen to be less than one.
3. K_a is calculated according to Eq. (21)
4. After comparing the overall closed loop equation with desired characteristic equation, the K_c is determined as below:

$$K_c = \frac{\omega_n^3}{b_2 x - \omega_n^3 K_b b_1} \tag{25}$$

5. THE PROPOSED HYBRID MODIFIED RBF NEURO-DEADBEAT CONTROLLER DESIGN

Instead of explicitly choosing a value for the series gain K of the designed deadbeat controller, an adaptive self-tune function $K(t)$ is designed based on the modified feed-forward radial basis function (RBF) neural network.

In general, the basic structure of the RBF neural network consists of three layers (input layer, hidden layer and output layer). The input layer is just buffered units, which pass the signal without altering it. The hidden layers are nonlinear activation functions (Gauss basis function). The output layers have linear units which add the signals fed to them [34].

In this paper, the suggested modified radial basis function (MRBF) neural network is shown in Fig. 5, this network consists of input layer with single node (neuron) g(t), hidden layer with three node h_j (j = 1, 2, 3) and single output layer in addition to nonlinear function $f(t)$, which is used to improve the performance of RBF network.

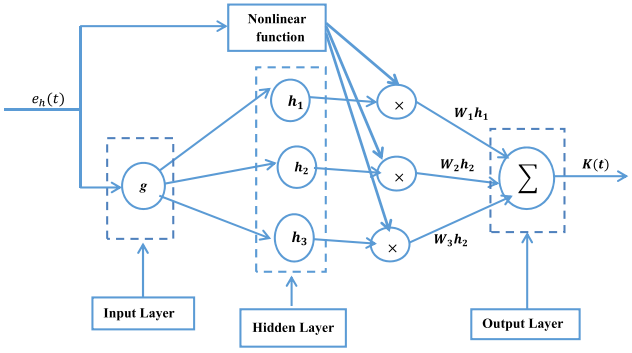

Fig. 5. The structure of the proposed modified RBF neural network

The function of the single node input layer ($g(t)$) is given by:

$$g(t) = \gamma \cdot e_h(t) \tag{26}$$

where γ is a selected constant value.

The nonlinear function of the hidden layer node is [35]:

$$h_j = \exp\left(\frac{-\left\|e_h(t) - c_{ij}\right\|^2}{2b_j^2}\right) \quad j = 1,\ 2,\ 3 \tag{27}$$

where the error signal $e_h(t)$ represent the input of the RBF network and $f(t)$ function, b_j is a measure of the width of the ith Gasussian function with width center c_{ij}, jth is the number of hidden layer nodes in the network. The function of the neural network weights $W = [W_1,\ W_2,\ W_3]T$ is designed as:

$$W_j(t) = W_j(t-1) + f(t) \quad j = 1,\ 2,\ 3 \tag{28}$$

The $f(t)$ is designed as [36]:

$$f(t) = \mu_f \cdot [\text{sat}(e_h(t))]^2 \tag{29}$$

where μ_f is a suitable selected value, the sat(.) function is used to limit the error signal. The function of the single output node of the RBF network is given by:

$$K(t) = \sum_{i=1}^{3} W_j(t) h_j(t) \tag{30}$$

As mentioned previously, the proposed MRBF network is used to generate the $K(t)$ for the deadbeat controller instead of using series gain K. Accordingly, combining the deadbeat controller and the MRBF, the control law of Eq. (11) will be modified to the following form:

$$u(s) = K(t)u_c(s) + u_o(s) \tag{31}$$

6. SIMULATION RESULTS

Firstly, the efficiency of the reduction methods (MRM method and the improved OMRM) was evaluated by calculating the Integral Square Error (ISE) between the output response of the original model and the reduced models by using Matlab software version R2015a. Table 3 shows three higher order tested examples whose reduction models by MRM, and OMRM are given in Table 4. The simulation results for these examples tested by unit step input signal are illustrated by Fig. 6. It can be seen from these results and the values of ISE in Table 4 that the OMRM result in more accurate reduction model, which is given a very close behavior to the performance of the original higher order model either in transient or steady state response than the MRM. This is due to the use of the FPA, which gives optimum values for the reduction model by MRM.

The above results also shows that if the order of the original system is very high (Ex.3), there will be an error in transient response of the obtained reduction model by MRM, but this error is less in the model obtained by the OMRM.

In other words, the suggested RDBC, RDBC-MRBF are designed to control the higher order system of Table 3 depending on the reduced model of OMRM, with $t_s = 4.82$ s, $\mu = 0.95$, $t_{ds} = 2.5368$ s and hence $\omega_n = 2$ rad/s for all examples, and the final parameters for both controllers are given in Table 5, expect K value is not used with RDBC-MRBF. While the parameters of the designed MRBF neural network are explained by Table 6.

The robustness, stability assurance, and the efficiency of the proposed RDBC, and RDBC-MRBF are illustrated by simulation results in Fig. 7 and Table 7 for step input signal, these figures show output responses, and control efforts of the original three tested systems. These results show that the performance of both controllers (RDBC, and RDBC-MRBF) are efficient, especially RDBC-MRBF because it makes the original higher order model flow the reference input signal with small settling time, small peak time, and small overshoot as compared with RDBC with same steady state control signal ($u_{s.s}$), as illustrated in Table 7. The efficiency of the RDBC-MRBF is due to making the

Table 3. Higher order systems tested examples

Example No.	Original higher order systems
Ex.1 [23]	$\dfrac{s^3 + 7s^2 + 24s + 24}{s^4 + 10s^3 + 35s^2 + 50s + 24}$
Ex.2 [37]	$G(s) = \dfrac{3s^5 + 10s^4 + 1221s^3 + 5792s^2 + 11860s + 8400}{s^6 + 41s^5 + 571s^4 + 3491s^3 + 10060s^2 + 13100s + 6000}$
Ex.3 [38]	$G(s) = \dfrac{35s^7 + 1086s^6 + 13285s^5 + 82402s^4 + 278376s^3 + 511812s^2 + 482964s + 194450}{s^8 + 33s^7 + 437s^6 + 3017s^5 + 11870s^4 + 27470s^3 + 37492s^2 + 28880s + 9600}$

Table 4. Comparison of the reduced model and ISE for MRM and OMRM

Example No.	Reduced model by MRM	Reduced model by OMRM	ISE of MRM	ISE of OMRM
Ex.1	$G_r = \dfrac{0.3263s + 3.9203}{s^2 + 4.5733s + 3.923}$	$G_r = \dfrac{0.7472s + 1.4309}{s^2 + 2.2866s + 1.4358}$	0.001629	0.0001018
Ex.2	$G_r = \dfrac{2.6881s + 8.6545}{s^2 + 6.6889s + 6.1818}$	$G_r = \dfrac{2.6881s + 8.6545}{s^2 + 6.622s + 6.188}$	0.000105	0.0000741
Ex.3	$G_r = \dfrac{43.1009s + 182.730}{s^2 + 8.718s + 9.02}$	$G_r = \dfrac{38.7908s + 182.7300}{s^2 + 7.0616s + 9.02}$	7.917	1.12

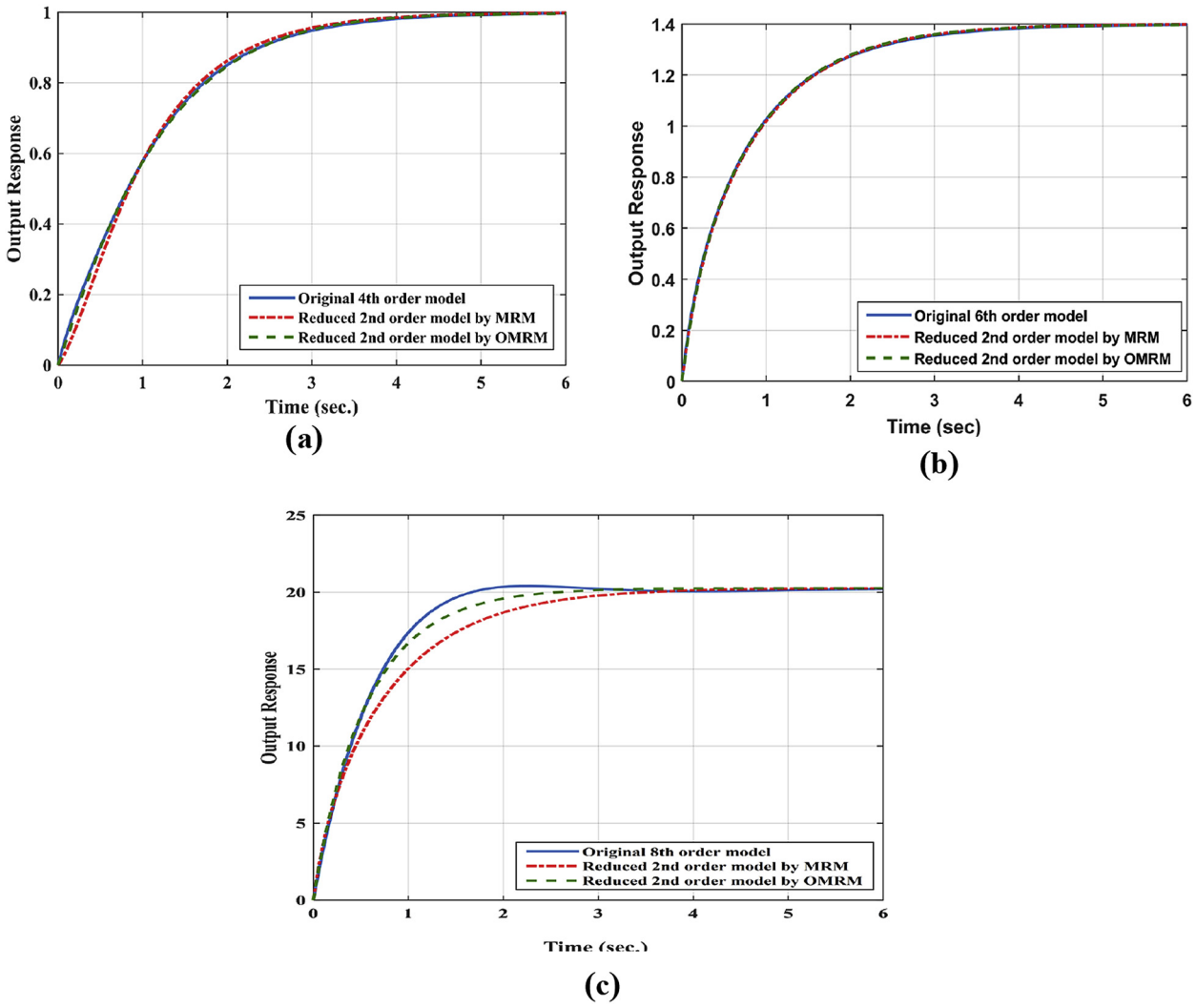

Fig. 6. Comparison of step responses for the original model and the reduced model of the three tested examples

constant gain K update by the suggested MRBF, which gives the appropriate value to improve the efficiency of the RDBC.

By using the same designed parameter values of Table 5 and unit step input, the performance efficiency of the RDBC, and RDBC-MRBF is also evaluated by testing the original higher order examples with 20% uncertainty in their parameters, the simulation results in this case are illustrated in Table 8, which shows the efficiency of both RDBC and RDBC-MRBF specially RDBC-MRBF because the higher order models flow the reference signal with small M_p, t_p , t_s, $e_{s.s}$, ISE with this controller.

Table 5. Deadbeat design parameters based on the reduced model

Example No.	T_i	x	K_b	K_a	K_c	K
Ex.1	0.1	10	0.05	2	1.1594	4
Ex.2	0.1	10	0.05	2	0.3416	5
Ex.3	0.1	10	0.01	0.4	0.0214	7

Table 6. Parameters of the designed MRBF network for the tested examples

Parameters	γ	μ_f	C_1	C_2	C_3
Values	0.5	1	1.5	1.5	0

To check the ability of the designed RDBC, and RDBC-MRBF to force the higher order tested examples to follow an arbitrary reference input signal at same designed parameter values given by Table 5 are used to test the performance of Ex.3 (eighth order model) for an arbitrary input signal. The simulation results of this case for output response, control signal, error signal, and the update $K(t)$ are shown in Fig. 8. Again, the simulation results show the efficiency of both controllers, especially RDBC-MRBF, which make the higher order model flow the arbitrary input signal very fast with nearly zero steady state error and no overshoot, in addition to smooth control signal with adaptive parameters for the $K(t)$.

7. SUMMARY AND CONCLUSIONS

The RBF neural network has a simple design with promising performance, it shows tolerance to different noises that come with inputs and data real-time capture capability. The

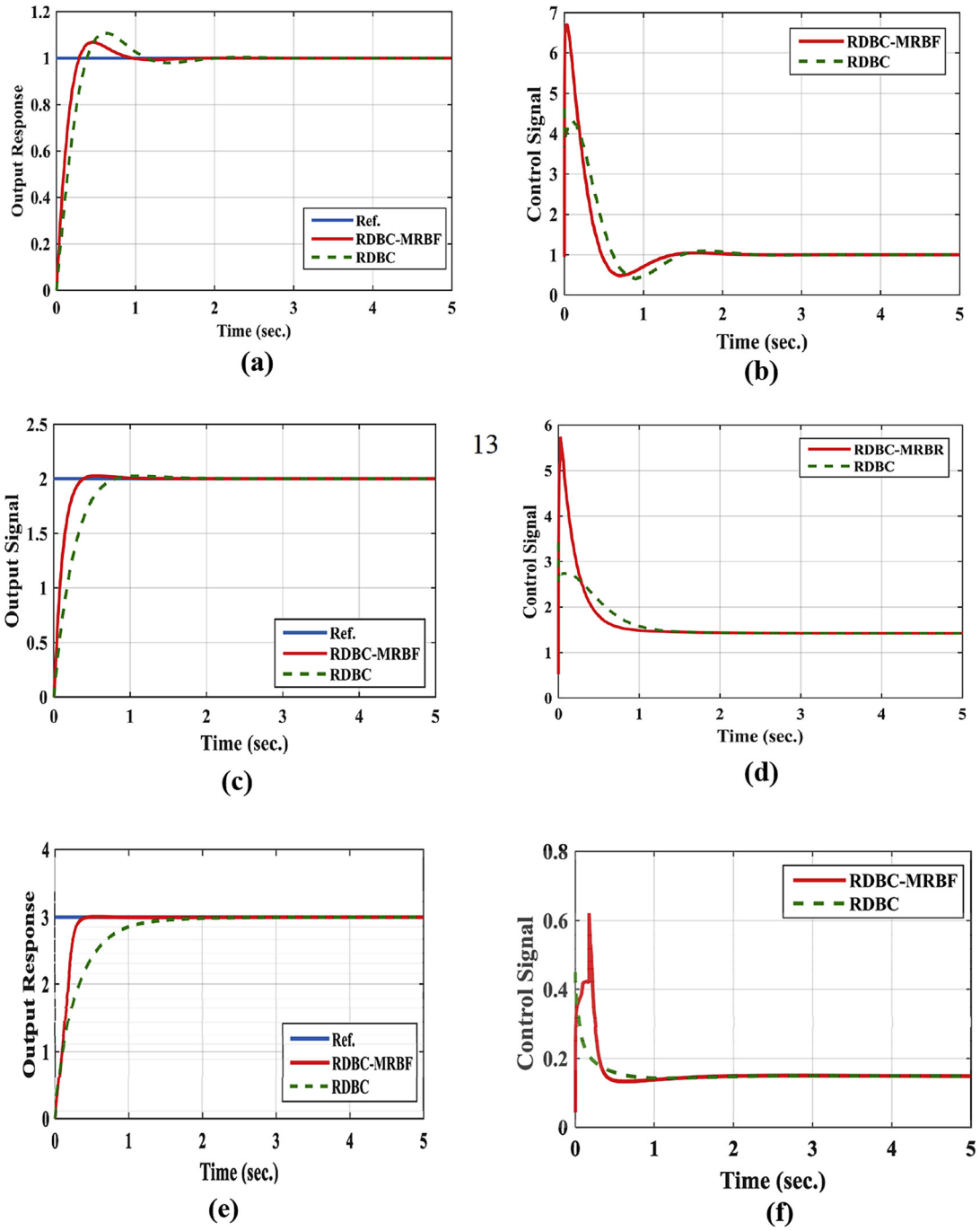

Fig. 7. Simulation results for original higher order examples controlled by (RDBC, and RDBC-MRBF) with step reference input, (a) Output response of Ex.1, (b) control signal of Ex.1, (c) Output response of Ex.2, (d) control signal of Ex.2; (e) Output response of Ex.3, (f) control signal of Ex.3

Table 7. The Performance comparison of the (RDBC, and RDBC-MRBF) for the tested examples

Example No.	Controller type	M_p	t_p	t_s	$e_{s.s}$	$u_{s.s}$	$f_{s.s}$	ISE
Ex.1	RDBC	0.071	0.6	1.4392	0	1	–	0.05776
Ex.1	RDBC-MRBF	0.0689	0.4293	0.7719	0	1	9.138	0.02341
Ex.2	RDBC	0.01245	1.0816	0.6891	0	1.4287	–	0.2985
Ex.2	RDBC-MRBF	0.01245	0.534	0.3122	0.0001	1.4285	20.5018	0.05885
Ex.3	RDBC	0	–	1.3415	0.0005	0.1856	–	1.259
Ex.3	RDBC-MRBF	0.00767	0.5	0.3327	0.0002	0.1858	16.6587	0.7363

Table 8. The Performance comparison of the (RDBC, and RDBC-MRBF) for the three tested examples with step input and 20% parameter uncertainty

Example No.	Controller type	M_p	t_p	t_s	$e_{s.s}$	$u_{s.s}$	$K_{s.s}$	ISE
Ex.1	RDBC	0.0862	0.7	1.7244	0	1.499	–	0.07263
Ex.1	RDBC-MRBF	0.132	0.479	0.8203	0	1.25	9.4707	0.02994
Ex.2	RDBC	0.0238	0.7	0.6898	0	0.8929	–	0.8719
Ex.2	RDBC-MRBF	0.0862	0.4790	0.8597	0	0.8929	9.5826	0.04583
Ex.3	RDBC	0	–	1.34	0.0002	0.0619	–	0.1399
Ex.3	RDBC-MRBF	0	–	0.98	0.0001	0.0619	8.9585	0.1159

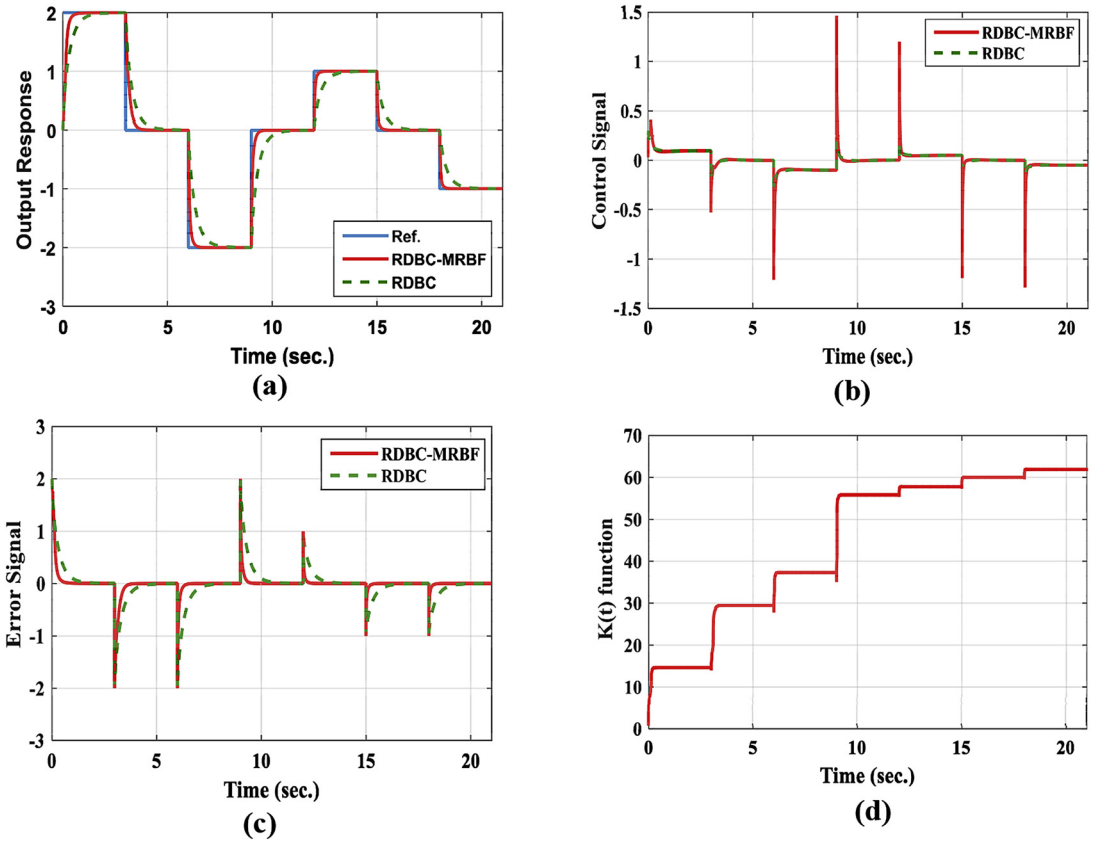

Fig. 8. Simulation results for original 8th order model (Ex. 3) controlled by (RDBC, and RDBC-MRBF) with arbitrary input signal, (a) Output response, (b) control signal, (c) error signal, (d) nonlinear function K(*t*) response

characteristics of this network make it very suitable for adaptive control framework. Therefore, in this paper, a simple modified RBF neural network is proposed for enhancing the performance of the suggested RDBC, which is used to control higher order linear systems based on a reduced second order model. The reduced model is produced by a suggested optimized mixed reduction method.

The simulation results for three higher order systems have shown that the proposed hybrid controller scheme effectively controlled these systems, even with arbitrary input or there is 20% uncertainty in system parameters. The hybrid controller scheme is simple; therefore, it can be used to control the single input-single output (SISO) nonlinear systems based on a linearized model such as inverted pendulum, mass-spring, dc motor, and single robotic arm and so on.

REFERENCES

[1] J. Yadav, N. Patidar, J. Singhai, S. Panda, and C. Ardil, "A combined conventional and differential evolution method for model order reduction," *Int. J. Comput. Intell.*, vol. 5, no. 2, pp. 111–8, 2009.

[2] S. Panda, J. Yadav, N. Patidar, and C. Ardil, "Evolutionary techniques for model order reduction of large scale linear systems," *Int. J. Appl. Sci. Eng. Technol.*, vol. 5, no. 1, pp. 22–8, 2009.

[3] V. Azimi, M. B. Menhaj, and A. Fakharian, "Tool position tracking control of a nonlinear uncertain flexible robot manipulator by using robust H_2/H_∞ controller via T–S fuzzy model", *Sadhana*, vol. 40, pp. 307–33, 2005.

[4] V. Emre, U. büyükşahin, R. Artar, A. Kirli, and T. Nurullah, "An experimental stationary quadrotor with variable DOF," *Sadhana*, vol. 38, pp. 247–64, 2013.

[5] S. Deepa, and G. Sugumaran, "Design of PID controller for higher order continuous systems using MPSO based model formulation technique," *Int. J. Electron. Eng.*, vol. 5, pp. 289–95, 2011.
[6] H. Khalil, *Nonlinear Systems*, Upper Saddle River, Prentice-Hall, 2002.
[7] S. Saha, S. Das, R. Ghosh, et al. "Design of a fractional order phase shaper for iso-damped control of a PHWR under step-back condition," *IEEE Trans.*, vol. 57, pp. 1602–12, 2010.
[8] MATHWORKS, *Simplifying Higher-Order Plant Models*, 2008 http://in.mathworks.com/help/robust/examples/simplifying-higher-order-plant-models.html.
[9] S. Das, S. Saha, S. Das, and A. Gupta, "On the selection of tuning methodology of FOPID controllers for the control of higher order processes," *ISA Trans.*, vol. 50, pp. 376–88, 2011.
[10] D. Jukic and T. Saric, "Dynamic model reduction: an overview of available techniques with application to power systems," *Serbian J. Electr. Eng.*, vol. 9, pp. 131–69, 2012.
[11] P. Shah and S. Agashe, "Design of controller for a higher order system without using model reduction methods," *Prog. Fract. Differ. Appl. Int. J.*, vol. 3, no. 4, pp. 289–304, 2017.
[12] B. Moore, "Principal component analysis in linear systems: controllability, observability, and model reduction," *IEEE Trans. Automatic Control*, vol. 26, pp. 17–32, 1981.
[13] M. Safonov and R. Chiang, "A schur method for balanced-truncation model reduction," *Automatic Contr. IEEE Trans.*, vol. 34, pp. 729–33, 1989.
[14] A. Askarzadeh and E. Rashedi, "Harmony search algorithm: basic concepts and engineering applications," in *IGI Global, Book: Recent Developments in Intelligent Nature – Inspired Computing*, 2017.
[15] N. Anand and A. Krishna, "Model order reduction using hybrid optimization algorithm", *Int. J. Res. Eng. Appl. Manage. India*, vol. 3, no. 1, pp. 274–7, 2018.
[16] S. Xiao, W. Wang, H. Wang, et al. "An improved artificial bee colony algorithm based on elite strategy and dimension learning," *Mathematics, China*, vol. 7, pp. 1–17, March 2019.
[17] H. Abdullah, "A hybrid bacteria foraging and modified particle swarm optimization for model order reduction," *Int. J. Comput. Eng. Iraq*, vol. 9, no. 2, pp. 42–9, April 2019.
[18] M. Kumar, S. Harish, and A. Apparao, "Design of PID controller via novel model order reduction technique," *Int. J. Adv. Res. Computer Commun. Eng.*, vol. 4, no. 5, pp. 89–97, May 2015.
[19] J. Rana, R. Prasad, and R. Agarwal, "Designing of a controller by using model order reduction techniques," *Int. J. Eng. Innov. Res.*, vol. 5, no. 3, pp. 134–42, 2016.
[20] L. Priyadarshini and J. Lather, "Design of IMC-PID controller for a higher order system and ITS comparison with conventional PID controller," *Int. J Innov. Res. Electr. Electron. Instrum. Control Eng.*, vol. 1, no. 3, pp. 281–7, 2013.
[21] B. Kada and Y. Ghazzawi, "Robust PID controller design for an UAV flight control system," in *Proceedings of the World Congress on Engineering and Computer Science, USA*, 2, pp. 19–21, 2011.
[22] H. Mansor and S. B. Mohd Noor, "Design of QFT-based self-tuning deadbeat controller, world academy of science," *Eng. Technol. Int. J. Electr. Computer Eng.*, vol. 7, no. 7, pp. 13–22, 2013.
[23] J. Pal, A. Sinha, and N. Sinha, "Reduced-order modelling using Pole clustering and time-moments matching," *J. Inst. Eng. India*, vol. 76, pp. 1–6, 1995.
[24] X. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, Berlin, Springer, 2012, pp. 240–9.
[25] X. Yang, M. Karamanoglu, and H. Xingshi, "Flower pollination algorithm: a novel approach for multi-objective optimization," *Eng. Optim.*, vol. 46, pp. 1222–37, 2014.
[26] A. Kumar Kar, "Bio inspired computing – a review of algorithms and scope of applications," *Expert Syst. Appl.*, vol. 59, pp. 20–32, 2016, Science Direct, Elsevier.
[27] P. Saxena and A. Kothari, "Linear antenna array optimization using flower pollination algorithm," *Springer Plus*, vol. 5, no. 306, pp. 1–15, 2016.
[28] Y. Zhou, R. Wang, and Q. Luo, "Elite opposition-based flower pollination algorithm," *Neurocomputing*, vol. 188, pp. 294–310, 2015, Science Direct, Elsevier.
[29] J. Franc, V. Hagenmeyer, and S. Kuehl, "Deadbeat control for electrical drives: a robust and performant design based on differential flatness," *IEEE Trans. Power Electron.*, vol. 30, no. 8, pp. 4585–96, August 2015.
[30] K. Kim and M. Youn, "A simple and robust digital current control technique of A PM synchronous motor using time delay control approach," *IEEE Trans. Power Electron.*, vol. 16, no. 1, pp. 72–82, 2001.
[31] A. Bedekar and S. Shinde, "Robust deadbeat control of twin rotor multi input multi output system," *Int. J. Eng. Res. Technol.*, vol. 4, no. 05, pp. 62–77, May 2015.
[32] S. Yang and C. Lee, "A deadbeat current controller for field oriented induction motor drives," *IEEE Trans. Power Electron.*, vol. 17, no. 5, pp. 772–8, Sep. 2002.
[33] A. Bedekar and S. Shinde, "Robust deadbeat control of twin rotor multi input multi output system," *Int. J. Eng. Res. Technol.*, vol. 4, no. 05, pp. 623–31, 2015.
[34] Y. Tao, J. Zheng, and Y. Lin, "A sliding mode control based on a RBF neural network for deburring industry robotic systems," *Int. J. Adv. Rob. Syst.*, vol. 13, no. 1, pp. 1–8, 2016.
[35] C. Pham and Y. Wang, "Robust adaptive trajectory tracking sliding mode control based on neural networks for cleaning and detecting robot manipulators," *J. Intell. Rob. Syst.*, vol. 79, pp. 1–14, 2014.
[36] M. Fallahi and S. Azadi, "Adaptive control of a DC motor using neural network sliding mode control," in *IAENG Proceedings of the International Multi Conference of Engineers and Computer Scientists, Hong Kong*, 2, Mar. 18–20, 2009.
[37] B. Philip and J. Pal, "An evolutionary computation based approach for reduced order modelling of linear systems," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), India*, 2010.
[38] S. Palaniswami and S. Sivanandam, "Design of PID controller using lower order model," in *National Symposium on Intelligent Measurement and Control, India*, 2000, pp. 290–6.