ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2011.0037

AN ENERGY FUNCTION APPROACH FOR FINDING ROOTS OF CHARACTERISTIC EQUATION

Deepak Mishra[1] and Prem K. Kalra[2]

[1]*Department of Avionics, Indian Institute of Space Science and Technology, Kerala, India*
E-mail: dkmishra@gmail.com
[2]*Department of Electrical Engineering, Indian Institute of Technology Rajasthan, India*
E-mail: kalra@iitk.ac.in

Abstract
In this paper, an energy function approach for finding roots of a characteristic equation has been proposed. Finding the roots of a characteristics equation is considered as an optimization problem. We demonstrated that this problem can be solved with the application of feedback type neural network. The proposed approach is fast and robust against variation of parameter.

Keywords:
Hopfield Neural Network, Energy Function, Eigenvalues, Neural Networks

1. INTRODUCTION

Hopfield proposed a feedback neural network popularly known as Hopfield neural network [9], [10]. In his work, Hopfield designed a recurrent neural network as dynamical associative memory and showed that a single-layer of a fully connected network is capable of restoring a previously learned static pattern called a memory vector, ensuring its convergence from any initial condition [9]. The Hopfield network may be operated in a continuous mode or discrete mode, depending on the model adopted for describing the neurons. The continuous model of operation is based on an additive model [10]. On the other hand, the discrete mode of operation is based on McCulloch-Pitts model [9].

Hopfield network has been used for a variety of applications, such as associative memory [9], complex-valued associative memory [6], classification [31], and object recognition [27]. It is also used to solve optimization problems like Traveling salesman problem [11], finding solutions of linear and nonlinear equations [4], [23], Analog-to-Digital (ADC) converter [5], [29], Job Scheduling [30], Assignment problem [3] etc. However, the application of conventional Hopfield neural network in optimization was limited to quadratic cost functions only. To overcome this limitation, Samad and Harper [28] extended this network for higher order cost functions and proposed a higher order Hopfield neural network. Later on Miguel *et al.* applied it to several well known problems of optimization [2], [21].

The techniques of finding roots of a polynomial, found applications in diverse fields such as control system [25], digital signal processing [32], and image processing [1]. These problems can easily be solved by many traditional methods such as Laguerre method, Muller method, Hurwitz method, Newton-Raphson method [32], and Routh criterion [7]. Since last two decades, Neural network technique has been used successfully in many fields, e.g., in factorization of polynomials with two or more variables [12], [26], inversion of the non-singular matrices [19], solving linear/non–linear algebraic equations [4], [22], [24], etc. Similar to above mentioned applications application of neural networks in finding the roots of polynomial is a novel and important research topic [20]. Huang *et al.* contributed significant applications of feed-forward neural network for determining the roots of arbitrary order polynomials. The approach using backpropagation network to zero polynomial is to factorize the polynomial into subfactors on the hidden layer of the network, and use a suitable adaptive learning algorithm to train the connection weights, i.e. roots, from the input layer to the hidden layer until the defined output error between actual output and the desired output (the polynomial) converges to a predefined accuracy [16]. All these learning schemes are supervised learning schemes. In these schemes initialization of neural network parameters is one of the key issues. In their work, Huang et.al. depicts that neural network based root finder schemes compete equally with conventional root finder schemes. However, the application of Hopfield neural network for extracting the roots of the characteristic equation is not explored even today. In this paper, an energy function approach using slightly modified Hopfield neural network is attempted to compute the roots of a given characteristic equation. Our study concludes that a fixed weight neural network such as Hopfield neural network with minor modifications can be used for root finding.

The paper is organized as follows. Section 2 presents the formulation of modified Hopfield neural network for finding the roots of a given characteristic polynomial. Stability analysis of the proposed approach is carried out in section 3. Section 4, demonstrates few numerical experiments with proposed approach. We concluded our work with a brief discussion in section 5.

2. MODIFIED HOPFIELD NEURAL NETWORK FOR DETERMINING NON-REPEATED ROOTS OF A CHARACTERISTIC EQUATION

Consider the characteristic equation,

$$q(s) = a_0 s^n + a_1 s^{(n-1)} + ... + a_{(n-1)} s + a_n \tag{1}$$

$$a_0 > 0$$

Eq.(1), can be written in the following form,

$$q(s) = (s - s_1)(s - s_2)(s - s_3)\ldots(s - s_n) \tag{2}$$

where, s_1, s_2 . . ., s_n are the roots of Eq.(1). These roots can be real or in complex conjugate pair. Here, we assumed that all the roots can be written in the form of ($x + iy$). Variable s in Eq.(1) can be replaced by ($x + iy$). That results in following,

$$q(s) = a_0 (x+iy)^n + a_1 (x+iy)^{n-1} + \ldots + a_{n-1}(x+iy) + a_n ; a_0 > 0 \tag{3}$$

Expanding the Eq.(3) gives us a complex algebraic equation of form ($\sigma + i\omega$). That can be represented as follows,

$$q(s) = \Big\{ \Re\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) + \Im\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) \Big\}$$

or

$$q(s) = \sigma + i\omega \tag{4}$$

Here, $\Re(.)$gives the real part of the characteristic equation q(s) after expansion, while $\Im(.)$ gives the imaginary part. The individual terms of Eq.(4), i.e. σ and $\omega$are used for energy function formulation for the proposed model. This energy function will be used further to determine all possible non-repeated roots of the given characteristic equation. In our work, we consider this problem as an optimization problem and, in order to use a neural network model to solve an optimization problem, the problem is cast into the form of an energy function that the model minimizes. Hence, the energy function using Eq.(4) can be shown as follows,

$$\begin{aligned} E(x,y) = &\Big\{ \Re\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) + \Im\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) \Big\} \\ &\times \Big\{ \Re\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) - \Im\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) \Big\} ; a_0 > 0 \end{aligned}$$

or

$$E(x,y) = (\sigma + i\omega)(\sigma - i\omega) \tag{5}$$

which can be further simplify as,

$$E(x,y) = \Big[\Re\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) \Big]^2 + \Big[\Im\Big(a_0 (x+iy)^n + \ldots + a_{n-1}(x+iy) + a_n \Big) \Big]^2$$

or

$$E(x,y) = \left(\sigma^2 + \omega^2 \right) \tag{6}$$

The energy function given in Eq.(6) is used to design a modified Hopfield neural network for finding the non-repeated roots of the characteristic equation Eq.(1). Calculation of partial derivatives of Eq.(6) with respect to variables x and y, gives the dynamics of network, which can be written as,

$$\frac{du_x}{dt} = -\frac{\partial}{\partial x} E(x,y)$$

$$\frac{du_y}{dt} = -\frac{\partial}{\partial y} E(x,y)$$

$$x = \varphi(u_x)$$

$$y = \varphi(u_y) \tag{7}$$

Equations (6) and (7) are used to determine weights and biases of the proposed neural network.

As evident from [9], that the conventional Hopfield neural network utilizes linear aggregation of weights, biases, and inputs at the particular node. The aggregated inputs are then passed through a nonlinearity to achieve a nonlinear input-output relation. The output of every neuron in the network is fed back to all the neurons of the network. Hence, it forms a fully connected recurrent neural network. However, in the proposed modified Hopfield neural network model, nonlinear combinations of inputs are aggregated at the summing node. The nonlinear combination includes linear as well as nonlinear terms. This is loosely inspired from the phenomenon of nonlinear dendritic interaction. This nonlinear combination of inputs is aggregated via unique weights. Other features are identical to conventional Hopfield model.

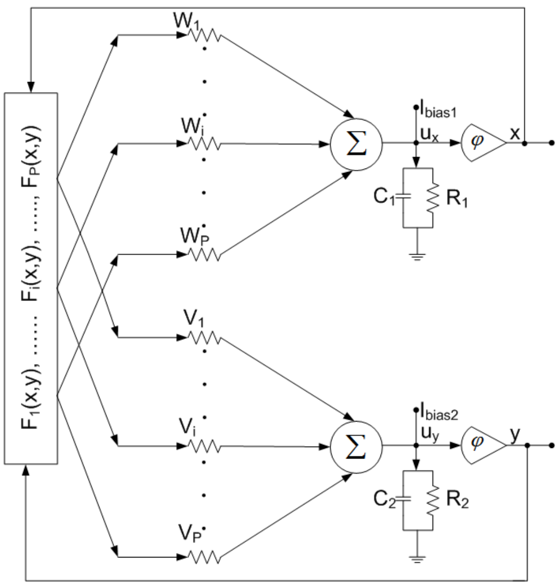

Fig.1. Generalized architecture of proposed model

The proposed architecture for the modified Hopfield neural network is shown in Fig.1. It is observed from the figure that highest order of the polynomial decides the number of weights and combination of states. As a general rule if n is the highest order present in the characteristic equation then the number of weights and combinations of x and y are given by expression $p(p+3)/2$, $p > 2$, here $p = n + 1$. For example, if we have to find the roots of a quadratic equation, then $n = 2$ and $p = 3$ which results in 9 different combinations of x and y which are aggregated through 9 synaptic weights.

Fig.1 helps us to write dynamical equations for the proposed network. These dynamical equations are expressed as,

$$\frac{du_x}{dt} = w_1 F_1(x,y) + \ldots + w_i F_i(x,y) + \ldots + w_P F_P(x,y) + I_{bias1}$$

$$\frac{du_y}{dt} = v_1 F_1(x,y) + \ldots + v_i F_i(x,y) + \ldots + v_P F_P(x,y) + I_{bias2}$$

$$x = \varphi(u_x)$$

$$y = \varphi(u_y)$$

$$P = \frac{p(p+3)}{2};\, p > 2 \tag{8}$$

where w_1 ,. . ., w_i ,. . ., w_P and v_1, . . ., v_i, . . ., v_P are the associated weights of the network. I_{bias1} and I_{bias2} are the constant bias values to the neurons x and y, respectively. $F_1(x, y)$,. . . , $F_i(x, y)$, . . ., $F_P(x, y)$ are various combinations of inputs x and y. The function $\varphi(.)$ is the input-output transfer function for the neurons. This function should be continually differentiable and monotonically increasing. Comparing Eq.(7) with Eq.(8) yields the weights w_1,. . ., w_i ,. . ., w_P, v_1 , . . ., v_i , . . ., v_P and biases I_{bias1} and I_{bias2}.

3. STABILITY ANALYSIS OF THE MODIFIED HOPFIELD NEURAL NETWORK

In this section, the stability analysis for the proposed modified Hopfield neural network is carried out. We use the basic notions of Lyapuonv stability theory [25], [34] in order to prove that the proposed energy function (Eq. 6) is a valid energy function.

Consider an autonomous system described by,

$$\frac{d}{dt}x(t) = f(x(t)) \tag{9}$$

and let, $x(x(t_0),t))$ be a solution. Let $E(x)$ is the total energy associated with the system. If the derivative $dE(x)/dt$ is negative for all $x(x(t_0),t)$ except the equilibrium point, then it follows that energy of the system decreases as t increases and finally the system reaches the equilibrium point. This holds because energy is non negative function of system state which reaches a minimum only if the system motion stops.

In section 2, we formulated the energy function for the proposed model with the help of Eq.(1) through Eq.(4). This energy $E(x,y)$ is the function of variables x, y and polynomial constant coefficients a_0, a_1, . . ., a_n. Eq.(5) can be rewritten as,

$$E(x,y) = \left(\sigma^2(x,y) + \omega^2(x,y)\right) \tag{10}$$

For the system to being stable, at any instant, the total energy $E(x,y)$ in the system is positive unless the system is at rest at the equilibrium state, where the energy is zero. This can be written as,

$$E(x,y) > 0; \text{when}\, x, y \neq 0$$

$$E(0,0) = 0; \text{when}\, x, y = 0 \tag{11}$$

The rate of change of energy is given by,

$$\frac{d}{dt}E(x,y) = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} \tag{12}$$

The dynamics for the modified Hopfield neural network is given by,

$$\frac{du_x}{dt} = -\frac{\partial}{\partial x}E(x,y) \tag{13}$$

$$\frac{du_y}{dt} = -\frac{\partial}{\partial y}E(x,y)$$

Using equations (12) and (13) we get,

$$\frac{d}{dt}E(x,y) = -\frac{du_x}{dt}\frac{dx}{dt} - \frac{du_y}{dt}\frac{dy}{dt} \tag{14}$$

States u $_x$ and u $_y$ are obtained by,

$$u_x = \varphi^{-1}(u_x)$$
$$u_y = \varphi^{-1}(u_y) \tag{15}$$

The activation function $\varphi(.)$ is chosen in such a way that, $\varphi(.)$ is continuously differentiable and strictly monotonically increasing, that is, $\varphi(p) > \varphi(p_0)$ if $p > p0$. Considering the relation as shown in Eq.(14) and Eq.(15) results in,

$$\frac{d}{dt}E(x,y) = -\left(\left(\frac{du_x}{dt}\right)^2 + \left(\frac{du_y}{dt}\right)^2\right)\frac{\partial\varphi^{-1}(.)}{\partial(x,y)} \tag{16}$$

Thus, $\frac{d}{dt}E(x,y)$ is negative at all points except the equilibrium state. This proves the stability of the modified Hopfield like network in Lyapunov sense. The total energy of the system will decrease with time and the proposed network is stable.

The gradient descent approach for minimizing the energy function $E(x, y)$ necessitates a positive slope for activation function $\varphi(.)$with respect to states x and y. In our experimentation, the input-output transfer function $\varphi(.)$ was linear. The input dynamics of network follows a gradient descent approach for minimizing the energy function $E(x, y)$, the network inputs are guaranteed to converge at a local minimum of $E(x, y)$, if $E(x, y)$ has one. Energy at a local minimum is zero, i.e. every minimum of $E(x, y)$ is a global minimum and hence corresponds to a solution [4].

4. ILLUSTRATIVE EXAMPLES

Methodology for calculating the non-repeated roots of a characteristic equation by using the modified Hopfield neural network approach is discussed in previous section. In this section we demonstrate usefulness of proposed method by several numerical experiments. Section 4.1 will describe the proposed approach in details. However, for subsequent examples only results have been presented. Initial states of the network are randomly initialized in the range of [-1, 1] and network is simulated on a machine configured with Intel Pentium 4 CPU 3.20GHz 512 MB RAM.

4.1 EXAMPLE 1

Consider a characteristic equation of quadratic form, i.e. $As^2+Bs+C = 0$. The parameters A, B, and C of this equation are any arbitrary real valued constants.

4.1.1 Energy Function Formulation:

The given characteristic equation has two roots which can be either complex conjugate pair or real depending on the values of parameters *A, B,* and *C*. The parameters $A, B, C \in R$ and variable $s \in C$; here *R* is the set of real number and *C* represents the set of complex number. Replacing *s* with $x + iy$ in the given characteristic equation results in,

$$A(x+iy)^2 + B(x+iy) + C = 0 \tag{17}$$

which is further simplified as,

$$\left(A\left(x^2 - y^2\right) + Bx + C\right) + i\left(2Axy + By\right) = 0 \tag{18}$$

The energy function for this problem is given by,

$$E(x, y) = (A(x^2 - y^2) + Bx + C)^2 + (2Axy + By)^2 \tag{19}$$

Eq.(19) is used further to calculate the weights and biases for the proposed network.

4.1.2 Design Details of Proposed Network:

With the calculated weights and bias values, network attempts to minimize the energy function $E(x, y)$ for obtaining the solutions. We assumed that the proposed network does not have any inherent loss terms, and its dynamics is governed by Eq.(7). Using equations (7) and (19), we get,

$$\frac{du_x}{dt} = w_1x^3 + w_2y^3 + w_3x^2y + w_4xy^2 + w_5x^2 + w_6y^2 + w_7xy + w_8x + w_9y + I_{bias1}$$

$$\frac{du_y}{dt} = v_1x^3 + v_2y^3 + v_3x^2y + v_4xy^2 + v_5x^2 + v_6y^2 + v_7xy + v_8x + v_9y + I_{bias2}$$

$$\begin{aligned} x &= \varphi\left(u_x\right) \\ y &= \varphi(u_y) \end{aligned} \tag{20}$$

Values of weights and biases are determined by comparing the RHS of Eq.(19) and Eq.(20). The architecture of the proposed network for quadratic characteristic equation is shown in Fig.2. Results for Example 1 are obtained by numerical simulation of Eq.(20) with A = 1, B = 2, and C = 3. In our experimentation, we noticed that in few cases the numerical method fails to converge if the integration step size is not appropriate and hence the selection of optimal integration step size is essential. Energy profile for the proposed network is drawn in Fig.3. It is evident from this figure that the energy of system minimizes with iterations and network requires small number of iterations to reach its minimum.

It is also noted here that all the non-repeated roots of the network are calculated sequentially. One complete simulation of the network results in one non-repeated root for the given polynomial. To evaluate all non-repeated roots the same network is simulated again and again with different initial states. In other words every time for a unique root for a characteristic equation the network is trained with unique random initial states. Each starting point in the basin of attraction must have different minima. If the starting point is same every time, we will get one root of the given characteristic equation. This is one of the limitations of proposed method because there is no mechanism for the selection of starting point. Because of this limitation the computational time is high as compared to conventional methods.

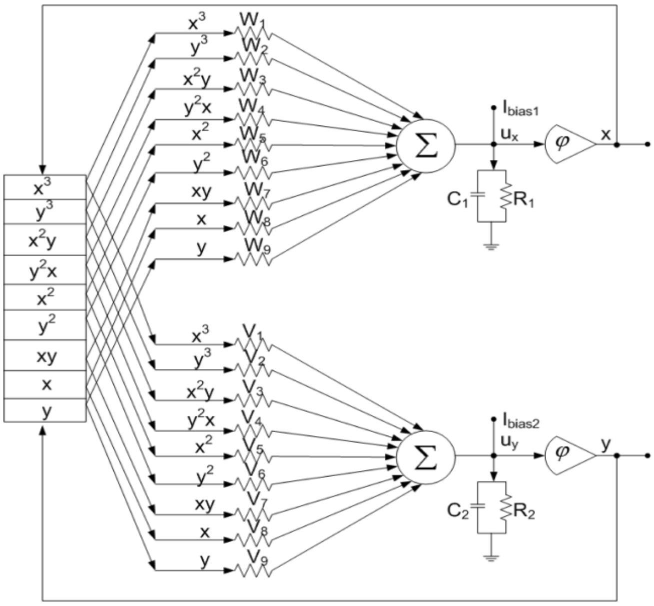

Fig.2. Architecture of the modified Hopfield neural network for Example 1

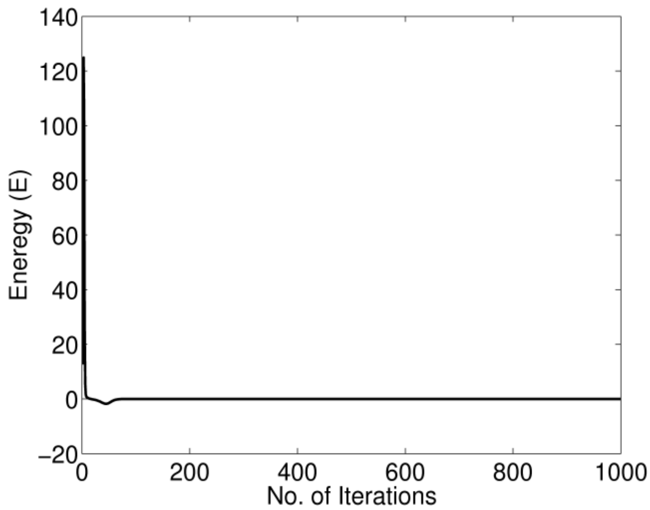

Fig.3. Energy profile for the proposed network while finding the roots of quadratic characteristic equation

Table.1 depicts the comparison between proposed energy minimization based neural network technique (PM) and conventional methods used for root determination, i.e. Lagurre[32] (LM) and Eigenvalues of balanced companion matrix method [8] (EBCM). It is evident from this table that the roots obtained with all the methods are almost identical but the computation time required by the proposed technique is high. Thus, the incorporation of nonlinear dendritic interaction in the Hopfield neural network can be used for determination of non-repeated complex roots of polynomial. However, this method is not computationally efficient method.

Table.1. The comparison of performance of the proposed method and Conventional methods for quadratic characteristic Equation, Eq.(17).

Indices	PM	LM	EBCM
Estimated roots	-1-1.414i -1+1.414i	-1-1.414i -1+1.414i	-1-1.414i -1+1.414i
CPU time(s)	0.39	0.094	0.015

4.2 EXAMPLE 2

The proposed methodology is used to calculate the non repeated roots of a cubic characteristic equation of form, i.e. $As^3 + Bs^2 + Cs + D = 0$. The parameters *A*, *B*, *C* and *D* can have any arbitrary real values. Table.2 presents the simulation results and comparison with conventional numerical methods.

Table.2. The comparison of performance of the proposed method and conventional methods for the cubic characteristic equation at A = 1, B = 2, C = 3, and D = 5

Indices	PM	LM	EBCM
Estimated roots	-1.8437 -0.07813-1.6449i -0.07813+1.6449i	-0.0781-1.644i -0.0781+1.644i -1.8437	-1.843 -0.0781+1.644i -0.0781-1.644i
CPU time(s)	0.50	0.078	0.047

4.3 EXAMPLE 3

In this section, we extended the proposed approach for finding all the non-repeated roots for a fifth order characteristic equation of form, i.e. $As^5 + Bs^4 + Cs^3 + Ds^2 + Es + F = 0$. The parameters *A*, *B*, *C*, *D*, *E*, and *F* can have any arbitrary real values. Table.3 presents the simulation results and comparison with conventional numerical methods.

Table.3. The comparison of performance of the proposed method and conventional methods for the fifth order characteristic equation at A = 1, B = 2, C = 3, D = 6, E = 3, and F = 5

Indices	PM	LM	EBCM
Estimated roots	-0.2703-1.166i -0.2703+1.166i -1.966 0.2534-1.307i 0.2534+1.307i	-0.2703-1.166i -0.2703+1.166i -1.966 0.2534-1.307i 0.2534+1.307i	-0.2703-1.166i -0.2703+1.166i -1.966 0.2534-1.307i 0.2534+1.307i
CPU time(s)	1.34	0.031	0.062

4.4 EXAMPLE 4

Here we exhibit simulation results for the seventh order characteristic equation of form $As^7 + Bs^6 + Cs^5 + Ds^4 + Es^3 + Fs^2 + Gs + H = 0$. The parameter *A*, *B*, *C*, *D*, *E*, *F*, *G*, and *H* can be any arbitrary real values. Table.4 presents the simulation results and comparison with conventional numerical methods.

Table.4. The comparison of performance of the proposed method and conventional methods for the seventh order characteristic equation at A = 1, B = 2, C = 3, D = 6, E = 3, F=5, G = 6 and H = 6.

Indices	PM	LM	EBCM
Estimated roots	-0.703-0.897i -0.03607-1.611i -0.03607+1.611i 0.7031+0.8971i -1.9081 -0.7129-0.6506i -0.7129+0.6506i	-0.7129-0.6506i -0.7129+0.6506i 0.7031+0.8971i 0.7031-0.8971i -0.03607-1.6113i -0.03607+1.6113i -1.9081	-1.908 -0.03607+1.6113i -0.03607-1.6113i 0.7031+0.8971i 0.7031-0.8971i -0.7129+0.6506i -0.7129-0.6506i
CPU time(s)	5.891	0.078	0.062

5. CONCLUSION AND DISCUSSION

In this paper, modified Hopfield neural network whose formulation is inspired from nonlinear dendritic aggregation is proposed. The modification lies in the application of linear as well as nonlinear terms in calculating the aggregation at each neuron unit.

We proposed a modified Hopfield neural network approach for determining the roots of a characteristic equation. We formulated an energy function that is used to determine the weights and biases for the proposed network. A generalized architecture of the proposed network is also given, which can be used to calculate non-repeated roots of any order of characteristic equation. Lyapunov stability analysis for the proposed network is carried out and it is found that proposed energy function for the network is stable in Lyapunov sense. Results for quadratic, cubic, fifth order, and seventh order characteristic equations are shown. These results are compared with the conventional Lagurre and Eigenvalues of balanced companion matrix methods. Computation time with each method is also compared. In our simulation the proposed method is not as fast as the conventional methods.

However, we believe that it is faster in comparison to feed forward type neural network. For determining the convergence and stability of the proposed network numerically, energy profile for the network is plotted. It has been noticed that the energy of the system minimizes with iterations. Thus, dynamics of the proposed network is stable.

Determination of proper weights and bias values, selection of suitable starting point, and higher computational time as compared to conventional numerical methods are some limitations of the proposed network. It is also observed that the proposed network should be trained with unique random initial states for approximately 5 times to the maximum number of roots in a given polynomial to evaluate non-repeated roots of given characteristic equation. It is also noted that each starting point in the basin of attraction must have different minima. If the starting point is same every time, we will get only one root. In this application, selection of starting point plays an important role. This can be rectified if some mechanism for the selection of starting point is developed.

This work can be taken as future research work. The proposed approach can also be extended for handling repeated roots of a characteristic polynomial. It is concluded that incorporating nonlinear dendritic aggregation in conventional Hopfield neural network and a proper formulation of energy function we can determine real as well as complex non-repeated roots of characteristic equations and set of nonlinear algebraic equations.

REFERENCES

[1] R. Anthony and R. Philip, "*A First Course in Numerical Analysis*", McGraw-Hill International Book Company, 1978.

[2] M. Atencia, G. Joya and F. Sandoval, "Dynamical Analysis of Continuous Higher-Order Hopfield Networks for Combinatorial Optimization", *Neural Computation*, Vol. 17, No. 8, pp. 1802–1819, 2005.

[3] C. Bousofio and M.R.W. Manning, "The Hopfield Neural Network Applied to the Quadratic Assignment Problem", *Neural Computing and Applications*, Vol. 3, No. 2, pp. 64–72, 1995.

[4] K. Chakraborty, K. Mehrotra, C.K. Mohan and S. Ranka, "An Optimization Network for Solving a Set of Simultaneous Linear Equations", *IEEE Proceedings of International Joint Conference on Neural Networks*, Vol. 2, pp. 516–521, 1992.

[5] V. Chande and P.G. Pooncha, "On Neural Networks for Analog to Digital Conversion", *IEEE Transaction on Neural Networks*, Vol. 6, No. 5, pp. 1269–1274, 1995.

[6] Z. Changhe and L. Liu, "The complex Hopfield model", *Optics Communications*, Vol. 103, No. 1–2, pp. 29–32, 1993.

[7] F.H. Chao, "Routh's method for finding complex roots", *Acta Mathematica Sinica*, *Chinese series*, Vol. 9, No. 2, pp.101–113, 1959.

[8] A. Edelman and H. Murakami, "Polynomial Roots from Companion Matrix Eigenvalues", *Mathematics of Computation*, Vol. 64, No. 210, pp. 763–776, 1995.

[9] J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of National Academy of Sciences of USA*, Vol. 79, No. 8, pp. 2554–2558, 1982.

[10] J.J. Hopfield, "Neurons with Graded Response have Collective Computational Properties like those of Two-State Neurons", *Proceedings of National Academy of Sciences of USA*, Vol. 81, No. 10, pp. 3088–3092, 1984.

[11] J.J. Hopfield and D.W. Tank, "Neural Computation of decision in Optimization problems", *Biological Cybernetics*, Vol. 52, pp. 141–152, 1985.

[12] R. Hormis, G. Antonion and S. Mentzelopoulou, "Separation of two-dimensional polynomials via a sigma-pie neural net", *Proceedings of the International Conference on Modelling and Simulation*, pp. 304–306, 1995.

[13] D.S. Huang, H.S.I. Horace, Z. Chi and H.S. Wong, "Dilation method for finding close roots of polynomials based on constrained learning neural networks", *Physics Letters A*, Vol. 309, No. 5–6, pp. 443– 451, 2003.

[14] D.S. Huang and Z. Chi, "Finding complex roots of polynomials by feedforward neural networks", *Proceedings of International Joint Conference on Neural Networks*, pp.A13–A18, 2001.

[15] D.S. Huang and Z. Chi, "Neural networks with problem decomposition for finding real roots of polynomials", *Proceedings of International Joint Conference on Neural Networks*, pp. A25–A30, 2001.

[16] D.S. Huang, H.S.I. Horace, K.C.K. Law and Z. Chi, "Zeroing polynomials using modified constrained neural network approach", *IEEE Transactions on Neural Networks*, Vol.16, No.3, pp. 721–732, 2005.

[17] D.S. Huang, H.S.I. Horace and Z. Chi, "A neural root finder of polynomials based on root moments", *Neural Computation*, Vol. 16, No. 8, pp. 1721–1762, 2004.

[18] D.S. Huang, "A constructive approach for finding arbitrary roots of polynomials by neural networks", *IEEE Transactions on Neural Networks*, Vol. 15, No. 2, pp. 477–491, 2004.

[19] D.S. Huang, "One-layer linear perceptron for the inversion of nonsingular matrix", *International Conference on ROVPIA'96*, Malaysia, pp. 639–643, 1996.

[20] D.S. Huang, H.S.I. Horace, K.C.K. Law, Z. Chi and H.S. Wong, "A new partitioning neural network model for recursively finding arbitrary roots of higher order arbitrary polynomials", *Applied Mathematics and Computation*, Vol. 162, No. 3, pp. 1183–1200, 2005.

[21] G. Joya, M.A. Atencia and F. Sandoval, "Associating Arbitrary-order Energy Functions to an Artificial Neural Network Implications Concerning the Resolution of Optimization Problems", *Neurocomputing*, Vol. 14, No. 2, pp. 139–156, 1997.

[22] L. Kinderman, A. Lewandowski, P. Protzel, "A framework for solving functional equations with neural networks", *Proceedings of the International Conference on Neural Information Processing*, Vol. 2, pp. 1075–1078, 2001.

[23] K. Mehrotra, C.K. Mohan and S. Ranka, "*Elements of Artificial Neural Networks*", The MIT Press, 1996.

[24] D. Mishra and P.K. Kalra, "Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations", *Engineering Letters on Soft Computing in Artificial Intelligence, Data and Web Mining, Machine Learning*. Vol. 14, No. 1, pp. 135-142, 2007.

[25] I.J. Nagrath and M. Gopal, "*Control Systems Engineering*", 2^{nd} edition, Halsted Press New York, NY, USA, 2003.

[26] S.J. Perantonis, N. Ampazis, S. Varoufakis and G. Antoniou, "Constrained learning in neural networks: application to stable factorization of 2-D Polynomials", *Neural Processing Letters*, Vol. 7, No. 1, pp. 5–14, 1998.

[27] N.M. Nasrabadi and W. Li, "Object Recognition by a Hopfield Neural Network", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 6, pp. 1523–1535, 1991.

[28] T. Samad and P. Harper, "High-order Hopfield and Tank Optimization Networks", *Parallel Computing*, Vol. 16, No. 2-3, pp. 287–292, 1990.

[29] D.W. Tank and J.J. Hopfield, "Simple Neural Optimization: An A/D Converter, a Single Decision Circuit and Linear Programming Circuit", *IEEE Transaction on Circuit and Systems*, Vol. 33, pp. 533–541, 1986.

[30] W. Wan-Liang, X. Xin-Li and W. Qi-Di, "Hopfield Neural Networks Approach for Job Shop Scheduling Problems", *Proceedings of the 2003 IEEE International Symposium on Intelligent Control Houston*, pp. 935–940, 2003.

[31] S. Wang, "Classification with Incomplete Survey Data: A Hopfield Neural Network Approach", *Computers and Operations Research*, Vol. 32, No. 10, pp. 2583–2594, 2005.

[32] H.P. William, A.T. Saul, T.V. William and P.F. Brian, "*Numerical Recipes in C: the Art of Scientific Computing*", Cambridge University Press, 1992.

[33] J.M. Zurada, "*Introduction to Artificial Neural Systems*", Jaico Publishing House, Mumbai, 1997.

[34] S.H. Zak, "*Systems and Control*", Oxford University Press, 2003.