A NEW MODIFIED LEVENBERG–MARQUARDT ALGORITHM IS USED IN BACK PROPAGATION NEURAL NETWORK (BPNN) TO A NONLINEAR DYNAMIC SYSTEM: A CASE STUDY

Jyh-Woei Lin

Department of Electrical Engineering, Southern Taiwan University of Science and Technology, Yungkang Dist., Tainan City, TAIWAN.

pgjwl1966@gmail.com

ABSTRACT

A new Modified Levenberg–Marquardt Algorithm (M-LMA), which has been a corrected version of traditional Levenberg–Marquardt Algorithm (LMA) was proposed to compared with the simulated results of LMA for minimizing back propagation errors using back propagation neural network (BPNN) in order to simulate a nonlinear dynamic system with same learning rates of 0.15. Resulted shown for this simulated nonlinear dynamic system, M-LMA was better than LMA.

Keywords: Modified Levenberg–Marquardt Algorithm (M-LMA); Levenberg–Marquardt Algorithm (LMA); Back Propagation Errors, Back Propagation Neural Network (BPNN), Nonlinear Dynamic System, Learning Rates

INTRODUCTION

The comments of Artificial Neural Network (ANN) related to Biological Neuron Network were given in a study (Lin, 2017). Updating the weight and bias using back-propagation is called back-propagation neural network (BPNN) (McCulloch and Pitts, 1943; Rochester, et al., 1956; Fukushima, 1980; Pickies, 1994; Bird, 2003; Mohri, et al., 2012). The classical Levenberg–Marquardt Algorithm (LMA) (Marquardt, 1963) is known as the damped least-squares (DLS) (Levenberg, 1944; Ortega and Rheinboldt, 2000), which is only a function of an independent variable. This learning rate is calculated arbitrarily in the classical LMA, causing it to converge prematurely when used for solving real world engineering problems. It has been used to decide desired minimum error (Hagan et al, 1997; Ngia and Sjoberg, 2000; Ham and Kostanic, 2001; Manolis and Lourakis 2005; Haykin, 2008; Finsterle and Kowalsky, 2010; Ipsen et al, 2011; Alippi, 2014; Pourbagher and Derakhshandeh; 2016). Therefore, the computed processing of The classical LMA to update weight and bias is not a parallel distribute processing **(PDP)** as a human neuron (Ferrier, 1876; Levenberg, 1944; Lin, 2017). Naveen et al (2010) used a type of classical LMA for Inverse Problems, which was a changed made from the classical LMA to enhance its performance by adaptive learning rate, wherein the learning rate was varied depending on the convergence of the objective function. Chen (2016) also used another type of classical LMA with line search for nonlinear equations. He corrected the computed style of classical LMA that is at every iteration, not only a LMA step, but also two additional approximating LMA steps in order to save the Jacobian calculation and employ line search for the step size, are computed. The results were very efficient and could save many calculations of the Jacobian. They have modified classical LMA. However their performances were not the PDP.

In this paper, BPNN is used to simulate a nonlinear dynamic system as shown in Figure.1 with a new modified Levenberg–Marquardt Algorithm (M-LMA) to update weight to perform a **PDP.**

LEVENBERG–MARQUARDT ALGORITHM (LMA)

A nonlinear real-valued function of f be composed of x. The parameter Y_t is defined as a target output (Rumelhart and McClelland, 1986) corresponding to the parameters x_t and y_t. That is, $Y_k = f(x_t)$. After approaching a nonlinear real-valued function, such as when applied to some simulated problems, the function f will generate the co-domain Y_k i.e., $Y_k = f(x_k), \mathrm{k} = 1,2....$. The codomain is defined as the initial codomain when k is 1. Therefore, limits of $\delta x \to 0$ can be considered. Let x^o best satisfies the real-valued functional relation f that must be determined to minimize $\varepsilon^T \varepsilon$ with the error $\varepsilon = Y_k - Y_t$, where T denotes the transposition of a vector. For small $\|\delta x\|$, where $\|.\|$ denotes the Euclidean norm, a Taylor-series expansion leads to the approximation

$$f(x+\delta x) \approx f(x) + J\delta x\,; \tag{1}$$

where J is the Jacobian matrix, $\dfrac{\partial f(x)}{\partial x}$; using this nonlinear iterative method then produces a series of (x_1), (x_2), (x_3)..., which finally converge toward a local minimizer $Y^o = f(x^o)$; that is, optimized output corresponding to the optimized x^o and. Therefore $\|Y_k - f(x+\delta x)\| \approx \|Y_k - f(x) - J\delta x\| = \|\varepsilon - J\delta x\|$ is estimated. The element of δx is used for a nonlinear least squares problem whose minimum is attained when $\varepsilon - J\delta x$ is orthogonal to the column space of J and $J^T(\varepsilon - J\delta x) = \delta x(uI)$, where I is the identity matrix.

Then, $$\delta x(uI) + J^T J\delta x = J^T \varepsilon \tag{2}$$

Thus formula (2) becomes

$$(I_\mu + H_\mu)(\delta x) = J^T \varepsilon\,, \tag{3}$$

where $H_a = J^T J$ is the approximated Hessian matrix (Ngia and Sjoberg, 2000; Pradeep et al, 2011; Fu et al, 2015) and $I_\mu = uI$, u is considered to be the learning rate.

MODIFIED LEVENBERG–MARQUARDT ALGORITHM (M-LMA)

Now let a nonlinear real-valued function of f be composed of x and y, which belong to independent variables. The parameter Y_t is defined as a target output corresponding to the parameters x_t and y_t. That is, $Y_k = f(x_t, y_t)$. After approaching a true curve, such as when applied to some simulated problems, the function f will generate the co-domain Y_k i.e., $Y_k = f(x_k, y_k), \mathrm{k} = 1,2....$. The codomain is defined as the initial codomain when k is 1. If $\delta x \to 0, \delta y \to 0$, then it is implied that $\delta x \delta y \to 0$. On the contrary, if $\delta x \delta y \to 0$, it is uncertain if $\delta x \to 0, \delta y \to 0$. Therefore, limits of $\delta x \to 0, \delta y \to 0$ can be considered. Let x^o and y^o best satisfy the real-valued curve functional relation f that must be determined to minimize $\varepsilon^T \varepsilon$ with the error $\varepsilon = Y_k - Y_t$, where T denotes the transposition of a vector. For small $\|\delta x, \delta y\|$, where $\|.\|$ denotes the Euclidean norm, a Taylor-series expansion leads to the approximation

$$f(x+\delta x, y+\delta y) \approx f(x, y) + J\delta x \delta y; \tag{4}$$

where J is the Jacobian matrix, $\dfrac{\partial f(x,y)}{\partial x \delta y}$; using this nonlinear iterative method then produces a series of (x_1, y_1), (x_2, y_2), (x_3, y_3)…, which finally converge toward a local minimizer $Y^o = f(x^o, y^o)$; that is, optimized output corresponding to the optimized x^o and y^o . Therefore $\|Y_k - f(x+\delta x, y+\delta y)\| \approx \|Y_k - f(x, y) - J\delta x \delta y\| = \|\varepsilon - J\delta x \delta y\|$ is estimated. The elements of δx and δy are used for a nonlinear least squares problem whose minimum is attained when $\varepsilon - J\delta x \delta y$ is orthogonal to the column space of J and $J^T(\varepsilon - J\delta x \delta y) = \delta x \delta y(uI)$, where I is the identity matrix. Then,

$$\delta x \delta y(uI) + J^T J\delta x \delta y = J^T \varepsilon \tag{5}$$

Thus formula (4) becomes

$$(I_\mu + H_\mu)(\delta x \delta y) = J^T \varepsilon, \tag{6}$$

where $H_a = J^T J$ is the approximated Hessian matrix and $I_\mu = uI$, u is considered to be the learning rate when simultaneously adjusting or updating the x, y values. Formulas (4), (5), and (6) belong to M-LMA. In this work, the parameters x and y are considered as the weight and bias of the BPNN (McCulloch and Pitts, 1943; Rochester, et al., 1956; Fukushima, 1980). Therefore, the updated processing of weight and bias in BPNN becomes a PDP, simulated more like the processing of a biological neuron using M-LMA.

TESTS AND RESULTS

A nonlinear dynamic system of y as an input u in Figure.1 has been simulated by BPNN using LMA and M-LMA to minimize back propagation Errors with the learning rates from 0 to 2 simultaneously. The frame of BPNN included 2 hidden layers and there were 10 neurons in each layer. Figure.2 has shown the results of LMA and M-LMA with same learning rate of 0.15. The learning rate of 0.15 was best using LMA and M-LMA in the range of 0 to 2. The simulated results of M-LMA were better because of the smaller simulated errors.

ISSN: 2186-8476, ISSN: 2186-8468 Print
www.ajsc. leena-luna.co.jp

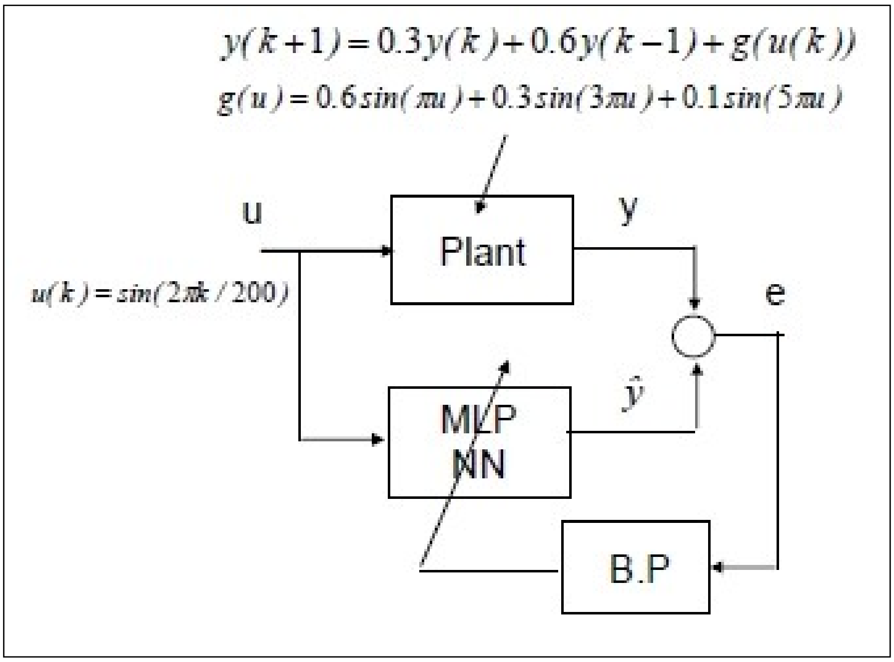
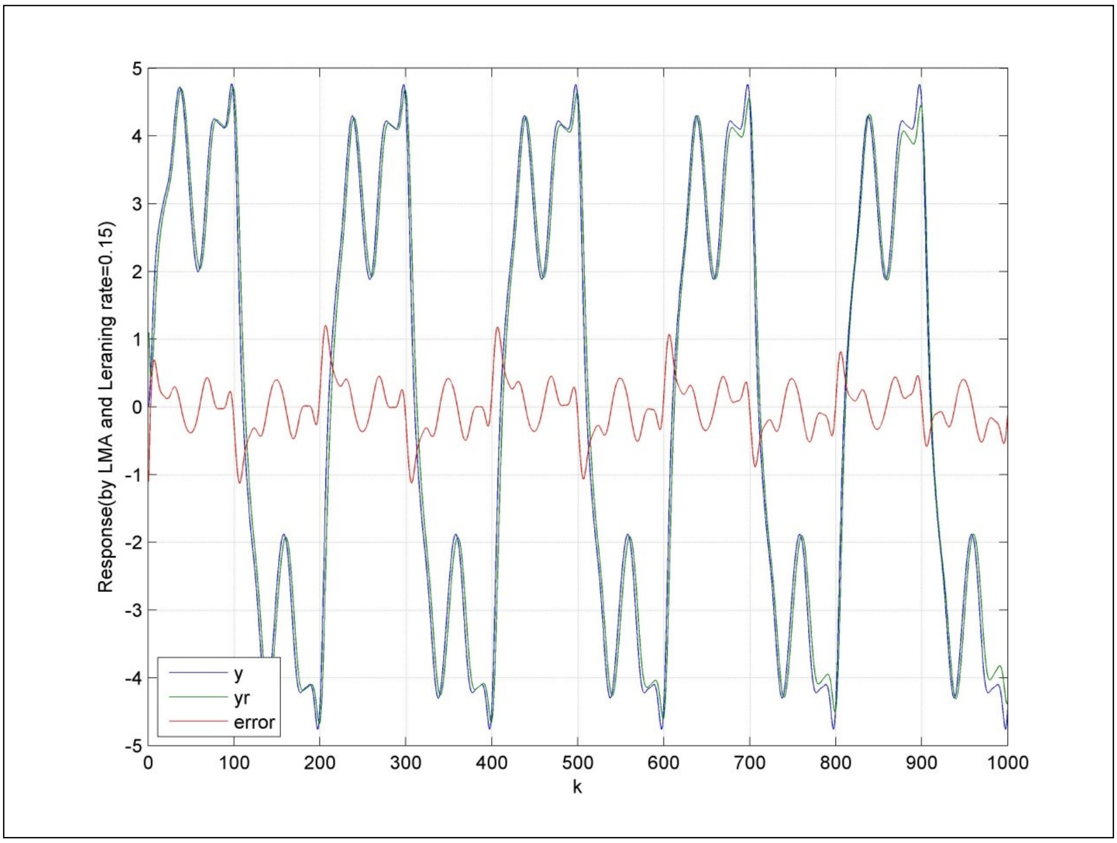


Figure.1 This figure shows the flowchart of a nonlinear dynamic system of y simulated by BPNN with the output $\hat{y}$.



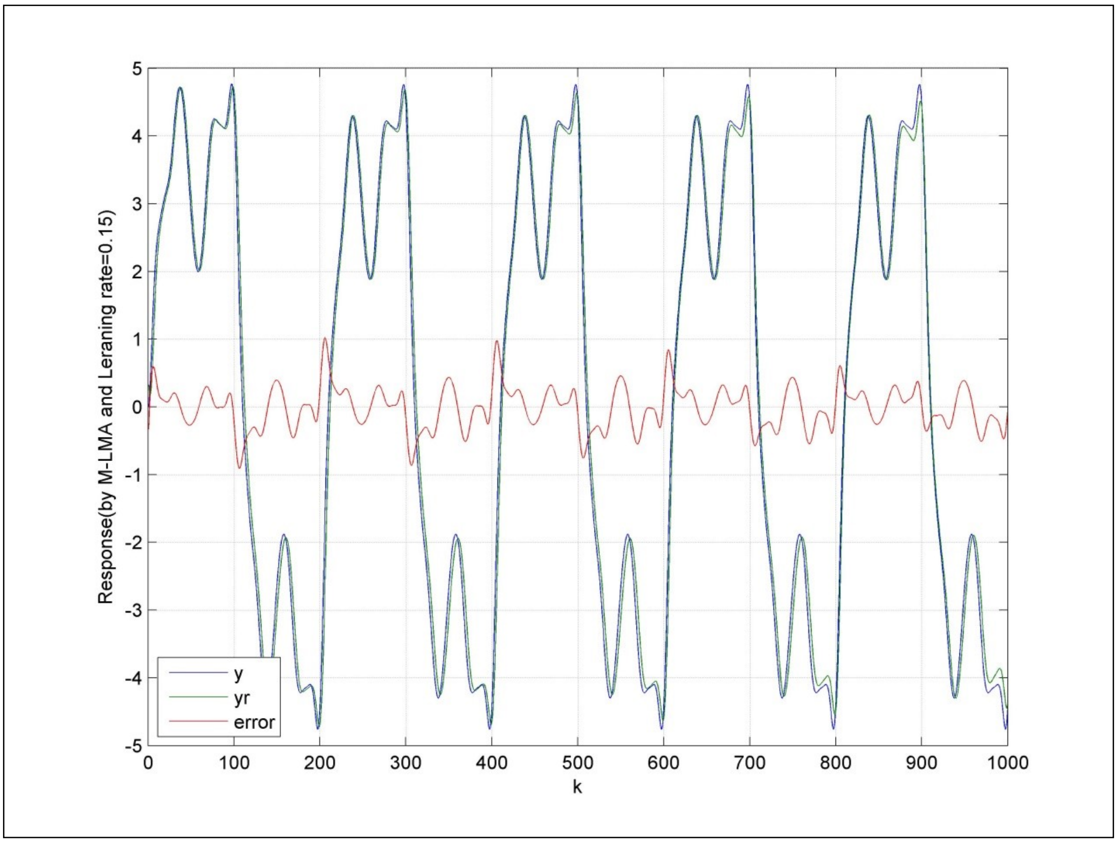


Fig.2 This figure shows the results of LMA and M-LMA with same learning rate of 0.15 simultaneously. The simulated errors of M-LMA are smaller. The blue line indicates the true outputs. The green line indicates the simulated outputs. The blue line indicates the simulated errors.

CONCLUSION

The LMA and M-LMA have been simultaneously performed to minimize back propagation errors for a simulating a nonlinear dynamic system by BPNN with same learning rate of 0.15. The M-LMA has shown the smaller simulated errors.

ACKNOWLEDGEMENTS

The authors are grateful to the support of Department of Electrical Engineering, Southern Taiwan University of Science and Technology.

ISSN: 2186-8476, ISSN: 2186-8468 Print
www.ajsc. leena-luna.co.jp

REFERENCES

[1] Alippi, C. (2014). *Intelligence for embedded systems: A methodological approach*. London: Springer International Publishing.

[2] Bird, A. (2003). Kuhn, nominalism, and empiricism. *Philosophy of Science, 70* (4), 690–719.

[3] Ferrier, D. (1876). *The functions of the brain*. London: Smith, Elder and Company.

[4] Finsterle, S., & Kowalsky, M. B. (2010). A truncated Levenberg–Marquardt algorithm for the calibration of highly parameterized nonlinear models. *Computers & Geosciences, 37*(6), 731-738.

[5] Fu, X., Li, S., Fairbank, M., Wunsch, D. C., & Alonso, E. (2015). Training recurrent neural networks with the Levenberg–Marquardt algorithm for optimal control of a grid-connected converter. *IEEE Transactions on Neural Networks and Learning Systems, 26* (9), 1900-1912.

[6] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36* (4), 93–202.

[7] Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks, 5* (6), 989-993.

[8] Hagan, M. T., Demuth, H. B., & Beale, M. (1997). *Neural network design*. MA, USA: PWS Publishing Co.

[9] Ham, F.M., & Kostanic, I. (2001). *Principles of neurocomputing for science and engineering*. USA: McGraw-Hill.

[10] Haykin, S. (2008). *Neural networks and learning machines (3rd Ed.)*. USA: Prentice Hall.

[11] Ipsen, I. C. F., Kelley, C. T., & Pope, S. R. (2011). Rank-deficient nonlinear least squares problems and subset selection. *SIAM J. Numer. Anal., 49*(3), 244–1266.

[12] Kecman, V. (2001). *Learning and soft computing: Support vector machines, neural networks, and fuzzy logic models*. Cambridge, MA: Massachusetts Institute of Technology Press.

[13] Kwon, S., Lynch, M., & Prokop, M. (2005). Decoupling PI controller design for a normal conducting RF cavity using a recursive Levenberg-Marquardt algorithm. *IEEE Transactions on Nuclear Science, 52* (1), 440-449.

[14] Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P., & Bottou, L. (2009). Exploring strategies for training deep neural networks. *Journal of Machine Learning Research, 1*, 1-40.

[15] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics, 2*, 164–168.

[16] Lin, J. W. (2017). Artificial neural network related to biological neuron network: A review. *Advanced studies in medical sciences, 5* (1), 55 – 62.

[17] Manolis, I., & Lourakis, A. (2005). A brief description of the Levenberg-Marquardt algorithm implemented by Levmar. Heraklion, Crete, Greece: Institute of Computer

Science, Foundation for Research and Technology.

[18] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics, 11*(2), 431–441.

[19] McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5* (4), 115–133.

[20] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. USA: MIT Press.

[21] Ngia, L. S. H., & Sjoberg, J. (2000). Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm. *IEEE Transactions on Signal Processing, 48* (7), 1915-1927.

[22] Ortega, J.M., & Rheinboldt, W.C. (2000). *Iterative solution of nonlinear equations in several variables, series: Classics in applied Mathematics*. Philadelphia: Society for Industrial and Applied Mathematics (SIAM).

[23] Pickies, J. (1994). *Ground truth: The social implications of geographic information systems*. USA: Guilford Press.

[24] Pourbagher, R., & Derakhshandeh, S. Y. (2016). Application of high-order Levenberg–Marquardt method for solving the power flow problem in the ill-conditioned systems. *IET Generation, Transmission & Distribution, 10* (12), 3017-3022.

[25] Pradeep, T., Srinivasu, P., Avadhani, P.S., & Murthy, Y.V.S. (2011). Comparison of variable learning rate and Levenberg-Marquardt back-propagation training algorithms for detecting attacks in Intrusion Detection Systems. *International Journal on Computer Science and Engineering, 3* (11), 3572-3581.

[26] Rao, N.D., & Tripathy, S. C. (1980). Power system static state estimation By the Levenberg-Marquardt algorithm. *IEEE Transactions on Power Apparatus and Systems, 99* (2), 695-702.

[27] Richter, C. F. (1935). An instrumental earthquake magnitude scale. *Bulletin of the Seismological Society of America, 25* (1), 1-32.

[28] Rochester, N., Holland, J.H., Habit, L.H., & Duda, W.L. (1956). Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions on Information Theory, 2* (3), 80–93.

[29] Rumelhart D. E., & McClelland, J. L. (1986). *Parallel distributed processing, explorations in the microstructure of cognition*. Cambridge, MA: MIT Press

[30] Shan, S. (2008). *A Levenberg-Marquardt method for large-scale bound-constrained nonlinear least-squares*. Vancouver: The Faculty of Graduate Studies (Computer Science), The University of British Columbia.

[31] Shawash, J., & Selviah, D. R. (2013). Real-time nonlinear parameter estimation using the Levenberg–Marquardt algorithm on field programmable gate arrays. *IEEE Transactions on Industrial Electronics, 60* (1), 170-176.

[32] Suratgar, A., Tavakoli, M. B., & Hoseinabadi, A. (2007). Modified Levenberg-Marquardt method for neural networks training. *International Journal of Computer, Electrical, Automation, Control and Information Engineering, 1* (6), 1745- 1747.

Leena and Luna International, Chikusei, Japan.
(株) リナアンドルナインターナショナル, 筑西市,日本
ISSN: 2186-8476, ISSN: 2186-8468 Print
www.ajsc. leena-luna.co.jp

[33] Wagarachchi, N. M., & Karunananda, A. S. (2014). *Towards a theoretical basis for modeling of hidden layer architecture in artificial neural networks*. USA: Institute of Research Engineers and Doctors

[34] Wilamowski, B. M., & Yu, H. (2010). Improved computation for Levenberg–Marquardt training. *IEEE Transactions on Neural Networks, 21* (6), 930-937.