

Adaptive GRNN for the Modelling of Dynamic Plants

Teo Lian Seng, Marzuki Khalid* and Rubiyah Yusof

Abstract— An integrated General Regression Neural Network (GRNN) adaptation scheme for dynamic plant modelling is proposed in this paper. It possesses several distinguished features compared to the original GRNN proposed by Specht [1], such as flexible pattern nodes add-in and delete-off mechanism, dynamic initial sigma assignment using non-statistical method, automatic target adjustment and sigma tuning. These adaptation strategies are formulated based on the inherent advantageous features found in GRNN, such as highly localised pattern nodes, good interpolation capability, instantaneous learning, etc.. Good modelling performance was obtained when the GRNN is tested on a linear plant in a noisy environment. It performs better than the well-known Extended Recursive Least Squares identification algorithm. In this paper, analysis on the effects of some of the adaptation parameters involving a nonlinear plant is also investigated. The results show that the proposed methodology is computationally efficient and exhibits several attractive features such as fast learning, flexible network sizing and good robustness, which are suitable for the construction of estimators or predictors for many model-based adaptive control strategies.

Index Terms-- General Regression neural network (GRNN), modelling, dynamic process, adaptation, system identification.

I. INTRODUCTION

GRNN has been applied in a number of applications for system control and identification [2-6]. There have been some comparative studies to demonstrate the modelling capability of the GRNN model with respect to other types of neural networks [2, 4, 7]. Although there are some studies on GRNN adaptation methods, the assignment of the sigmas is usually based on the overall statistical calculation from a pre-collected batch of training data [1, 8]. This approach may not be suitable to be applied in a continuous modelling environment as the model needs to be updated continuously due to the changes in plant dynamics or operating conditions. However, there is not much work reported on adaptive GRNN for modelling of dynamical systems, especially for online applications. Furthermore, investigations on the adaptation aspects of GRNN

The authors are from:

Center for Artificial Intelligence and Robotic
(CAIRO) Universiti Teknologi Malaysia, Jalan
Semarak, 54100 Kuala Lumpur, Malaysia. Email
address: marzuki@utmkl.utm.my (All correspondence
should be sent to *)

parameters in dynamic process modelling are still lacking and still in its infancy.

This paper proposes an integrated approach of the GRNN adaptation scheme for dynamical plant modelling. The adaptive GRNN modelling scheme is suitable to be applied in a noisy and dynamical control environment. The proposed adaptive GRNN model is equipped with some distinguished features not found in the original GRNN model [1], such as flexible pattern nodes mechanism with add-in and delete-off features, dynamic initial sigma assignment using a non-statistical method, and automatic adjustment of the targets and sigmas associated with the pattern nodes. These proposed adaptation strategies are basically formulated based on the inherent advantageous features of GRNN, such as expandable and reducible network structure and the exclusive local properties of the pattern nodes [1, 8]. The advantages and rationale of these strategies are experimentally investigated in modelling of linear and nonlinear plants. Relative performance of the proposed adaptive GRNN modelling technique is compared to the popularly known mathematical based Extended Recursive Least Squares identification algorithm (ERLS) [9]. Furthermore, the effects of some adaptation parameters to the overall modelling efficiency are also investigated.

II. GRNN FOR PROCESS PLANT MODELLING

The GRNN paradigm was proposed by Donald Specht as an alternative to the well known back-error propagation training algorithm for feed-forward neural networks [10]. It is closely related to the better-known probabilistic neural network [9]. Regression in this context can be thought of as the least-mean-squares estimation of variables based on available data. From a computational viewpoint, the GRNN is based on the estimation of a probability density function from observed samples using Parzen window estimation [11]. It utilises a probabilistic model between an independent vector random variable X with dimension D , and a dependent scalar random variable Y such that the expected value of Y given x (the regression of Y on x) can be estimated as:

$$E[Y | x] = \frac{\int_{-\infty}^{\infty} Y \cdot f(x, Y) dY}{\int_{-\infty}^{\infty} f(x, Y) dY} \quad \text{Eq.2.1}$$

where x and y are measured values for X and Y , respectively, and $f(X,Y)$ represents the known joint continuous probability density function. It is assumed that $f(X,Y)$ is also known. The computational procedures of GRNN can be viewed as the weighted average of all the observed data using the distance criteria in the input space [11].

A general structure of the GRNN can be illustrated as in Fig. 1. Input layer will simply channel in the input vector to the GRNN, and its distances to the recorded patterns are then calculated in each of the pattern nodes at pattern layer.

GRNN is capable of approximating any arbitrary function, either linear or non-linear relationships between input and output variables, drawing the function estimates directly [2]. It is particularly advantageous with sparse data in a real-time environment, because the regression surface is instantly defined, even by just one sample.

III. ADAPTIVE GRNN MODELLING SCHEME

Basically, the proposed adaptation methodology constitutes four strategies. The roles of these strategies cover the scope of creation of new pattern nodes, dynamic initialisation of new pattern nodes, adjustment of the targets, and tuning of the sigmas.

A. Adaptation strategy I: Creation of New Pattern Nodes

For this purpose, each of the pattern nodes is associated with a merit index ($\eta : 0 \leq \eta \leq 1$), with a value of one assigned upon its creation. The merit index η reflects the accumulated firing strength of that pattern node throughout the iterations. For prediction at the k^{th} sampling instant, the

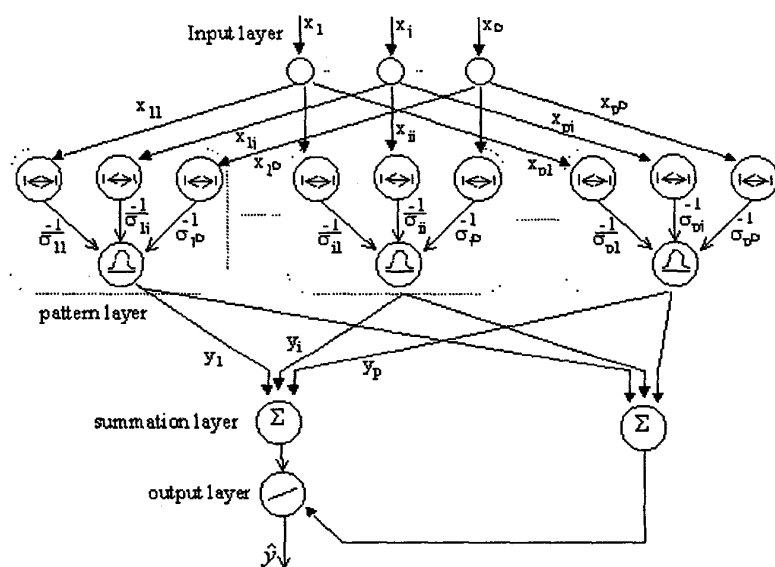


Fig. 1. Structure of the General Regression Neural Network

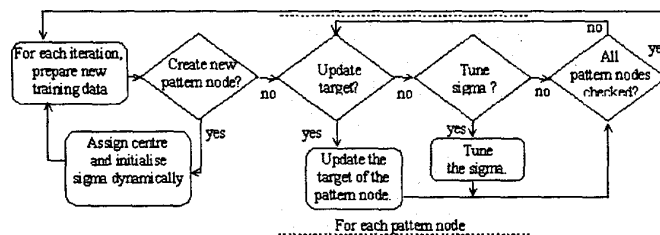


Fig. 2. Flowchart of GRNN model strategies

index η for the i^{th} pattern node is updated as follows:

$$\eta_i(k) = 0.99 * \eta_i(k-1) + f_i(k) \quad \text{Eq.3.1}$$

where $f_i(k)$ is the firing level of the i^{th} pattern node, and computed as:

$$f_i(k) = \exp \left[- \sum_{j=1}^D \left(\frac{x_i - x_{ij}}{\sigma_{ij}} \right)^2 \right] \quad \text{Eq.3.2}$$

The merit index can be used to indicate the worthiness of a particular pattern node in the network. The pattern nodes that are seldom used or are outdated would eventually have a comparatively lower merit index, and vice-versa. Besides updating new knowledge to the network within a particular network size, the replacement actions as discussed also serve as a simple and effective strategy to gradually phase out the out-dated pattern nodes.

B. Adaptation Strategy II: Dynamic Sigma Initialization

Finding an appropriate sigma for each of the variables of the pattern nodes could be a difficult task [1, 7]. It depends on the network input variables and also the plant characteristics. This paper introduces a simple, fast and dynamic sigma initialisation method based on the dynamic states of the plant, and without the need for statistical calculation. This scheme is able to assign the centre and width of the Gaussian kernel for each of the input variables effectively with less computation procedures compared to other clustering method [12, 13]. The initialisation of the sigma is based on the distance, i.e. the changing rate of the variable at the time when the new node was created. For the sigma of the i^{th} variable (x_i) of the j^{th} pattern node created at the k^{th} sampling instant, the initialisation method can be written as follows:

$$\begin{aligned} &\text{if } \{a_i + b_i | x_i(k) - x_i(k-1) \} < c_i \\ &\quad \sigma_{ij} = a_i + b_i | x_i(k) - x_i(k-1) | \\ &\text{else} \\ &\quad \sigma_{ij} = c_i \end{aligned} \quad \text{Eq.3.3}$$

where a_i and c_i are the defined lower and upper limits of the sigma to x_i respectively, and b_i is the slope rate of the sigma. The initial value of the sigma is thus bounded, i.e., ($a_i \leq \sigma_{ij} \leq c_i$), where a_i is a small positive value for the minimum sigma allowed. The parameter c_i prevents the sigma from becoming too large which may cause prediction inaccuracy in a finite sample modelling, such that to be over generalised.

C. Adaptation strategy III: Adjustment of the targets

The idea here is to merge the existing target with the feed-in training target. However, adjustment will only apply to pattern nodes which have high matching degrees, i.e. the firing levels are closed to 1. In this paper, a pattern node is qualified for adjusting targets if its firing level is above a ‘‘target update’’ threshold (γ). The target vector of the i^{th} pattern node to the i^{th} output variable is then updated as follows:

$$\begin{aligned} y_{it_new} &= (1 - \lambda) \frac{y_{it_training}}{f_i(k)} + \lambda \cdot y_{it_old} && \text{if } \left| y_{it_training} \right| \leq \left| y_{it_old} \right| \\ y_{it_new} &= (1 - \lambda) \cdot y_{it_training} \cdot f_i(k) + \lambda \cdot y_{it_old} && \text{else} \end{aligned} \quad \text{Eq.3.4}$$

where λ is the adjustment rate of the target, and $f_i(k)$ is the firing level of the i^{th} pattern node as in Eq(3.2). The parameter $y_{it_training}$ is the training target vector that is associated with the current input vectors, and y_{it_old} is the target vector currently associated with the i^{th} pattern node. The target update threshold ($0 \leq \gamma \leq 1$), is set closed to 1. Generally, the adaptation of the targets serves two functions, firstly, it helps to improve prediction error due to noise contamination; secondly, it helps to update the network if the plant characteristics vary over time.

D. Adaptation Strategy IV: Tuning of Sigma

The sigma initialisation method provided by Strategy II serves as a convenient way to assign the sigma appropriately, however, there are no means of generating an optimal sigma. Although it is reported that the GRNN algorithm does not get trapped in local minimas [2], tuning of the sigma would be necessary to further refine the prediction accuracy, especially in a dynamic modelling environment. The optimisation of the sigma is intended to minimise the prediction squared-error of the system. The sigma of the j^{th} variable for the i^{th} pattern node at the k^{th} sampling instant is then updated as follows:

$$\sigma_{ij}(k) = -\phi \frac{\partial e(k)}{\partial \sigma_{ij}} + \sigma_{ij}(k-1) \quad \text{Eq.3.8}$$

where ϕ is the learning rate of the sigma, which usually is a small constant.

While it is necessary to continuously improve the accuracy of the GRNN model, tuning the sigmas for all the pattern nodes in each sampling instant are rather time consuming which could cause problems in an online adaptation. On the other hand, the same objective can still be achieved by tuning only a group of identified pattern nodes that

contributed significantly to the model predicted output, where the number of nodes is usually small. In a serial computer, this tuning scheme can greatly save the time for tuning of the sigmas, and this is experimentally demonstrated in the following sections. In this paper, a minimum significant level (ψ) is defined to select the pattern nodes to be tuned. Only the pattern nodes that fire above the ψ level qualify for the gradient tuning procedure. In other words, only the sigma of those significantly contributed nodes, i.e., which are closely related to the input patterns, will be tuned.

In the adaptation strategies of the GRNN as discussed, the tuning of the pattern nodes is determined and the tuning magnitude is calculated locally based on its firing level. The overall local adaptation action of the pattern nodes using Strategy III and IV can be summarised as follows:

If the particular pattern node is fired above the target update threshold (γ), i.e. $f_i(k) > \gamma$, then its targets will be adjusted without tuning its sigmas. Else, if it is fired above the tuned sigma minimum level (ψ), i.e. $\psi < f_i(k) < \gamma$, the sigma of the pattern node will be tuned.

Note that, there will be no adaptation for the pattern nodes that are fired below ψ , i.e. $f_i(k) < \psi$. This highly localised GRNN adaptation strategy is, therefore, computationally efficient and is favourable for parallel processing.

IV. EXPERIMENTS AND RESULTS

Several experiments were carried out employing all of the proposed GRNN strategies [16] both on linear and nonlinear plants. Comparisons to the proposed GRNN technique in the modelling of dynamic plants are made where a number of popular identification techniques are also investigated on the same plants. However, due to space, we do not discuss in this paper. The next section discusses the application of the proposed GRNN technique on a benchmark nonlinear plant.

Figure 3 shows a typical configuration where GRNN is used in modelling a plant with randomly excited inputs. The inputs of the GRNN use only the delayed value of the plant inputs (u) and outputs (y). The parameter d is the plant time delay. These GRNN inputs also serve as the training vector for adaptation of the model. The parameter y_e is the measured plant output that is distorted by the noise, i.e. $\{y_e = y + \varepsilon\}$, where ε is the Gaussian noise. Although the network takes in y_e , the evaluation of the prediction performance is measured by the prediction error ($\hat{\varepsilon}$), which is the difference between the predicted system response (\hat{y}) and the actual system response (y). The prediction error, which is the accumulated normalised root mean square error ($ARMSE$) is represented as follows:

$$ARMSE = \sum_i^L (y_i - \hat{y}_i)^2 \quad \text{Eq. 4.1}$$

where L is the total number of data in the pre-generated data set. The maximum number of pattern nodes generated allowed in all the experiments is limited to 300 nodes.

A. Modelling of a nonlinear plant

By modelling a nonlinear plant, the effects of some of the adaptation parameters involved can be investigated, such as the “create new node” minimum threshold (α), targets adjustment threshold (γ) and sigma tuning rate (ϕ), to the overall modelling performance and the GRNN structural growth. In these experiments, we assume the nonlinearities of the plant are unknown.

For our investigation purposes, a dynamical process which has been used as a benchmark for validating other neural networks and fuzzy algorithms in plant identification [17-19] is used. Its dynamics is modified such that it has a higher nonlinearity and a more complex plant dynamics and is given as follows:

$$y(k+1) = \frac{y(k)y(k-1)[2.5 + y(k-1)]}{1 + y^2(k) + y^2(k-1)} + 2.0 \frac{e^{-u(k)} - 1}{e^{-u(k)} + 1} + \varepsilon(k) \quad \text{Eq. 4.2}$$

where $u(k)$ and $y(k)$ are the plant input and output signals, respectively, and $\varepsilon(k)$ is the Gaussian noise with zero mean. The GRNN model has four inputs and outputs.

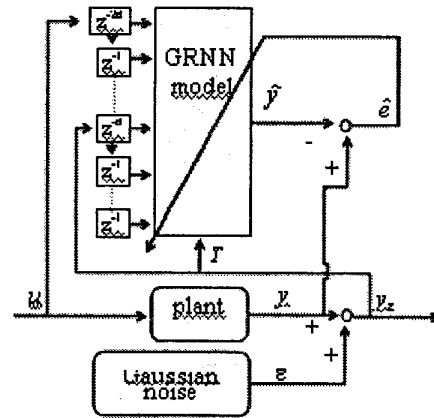


Fig.3 Training scheme of modelling with random plant inputs.

The inputs consist of a single delayed value of the plant input signal and two delayed values of the plant output.

The default adaptation parameters of the GRNN model are selected judiciously. Figure 4 shows the effects of different strategies in the modelling environment, which consists of Gaussian noise with normalised amplitude of 0.05. Figure 5 indicates the progress of the GRNN learning with different cases of noise amplitudes. It was found that the effects of employing different GRNN adaptation strategies for the linear (not-discussed in this paper) and nonlinear plants are basically similar. Figure 6 shows the output by the GRNN model when compared with that of the actual plant. It was found that Strategy II can effectively initialise each of the sigmas without much computation required.

V. CONCLUSION

This paper has discussed an effective methodology in using GRNN for the modelling of dynamic plants in which four adaptation strategies have been proposed and investigated experimentally. The proposed GRNN model was able to evolve from a null network to an appropriate size that can be practically used for modelling purposes without any loss in performance. Furthermore, the model is equipped with a dynamic and fast adaptation algorithm. The experimental results have shown that the GRNN is superior for modelling in a dynamical and even a noisy environment. In conclusion, the four strategies that have been proposed can be summarised to have the following advantages. Strategy I helps to create an appropriate network size as well as maintaining the updated pattern nodes. Strategy II overcomes the difficulty of initialising the sigma of the pattern nodes. Strategy III and IV further improve the prediction accuracy by continuously tuning the targets and sigmas of the pattern nodes, respectively. Further research in trying to apply the methodology to real physical systems are currently being investigated with some measurable success.

VI. REFERENCES

1. Specht, D.F., A General Regression Neural Network, *IEEE Transactions on Neural Network*, Vol.2, No.6, Nov, p568-76, 1991.
2. Hyun, B.G. and Nam, K., Faults Diagnoses of Rotating Machines by Using Neural Networks: GRNN and BPN, *Proc. of the 1995 IEEE IECON*, Vol.2, p1456-1461, 1995.
3. Specht, D.F. and Romsdahl, H., Experience With Adaptive Probabilistic and General Regression Neural Networks, *Proc. Of IEEE World Congress on Computational Intelligence*, Orlando, USA Vol.2, p1203-1208, 1994.
4. Marquez, L. And Hill, T., Function Approximation Using Backpropagation and General Regression Neural Networks, *Proc. 26th Hawaii Int. Conf. On Systems Sciences*, Vol.4, p607-15.
5. Patton, J.B., Brushless DC Motor Control Using a General Regression Neural Network, *Proc. Of the 1995 IECON*, Vol.2, p1422-1427, 1995.
6. Schaffner, C. And Schroder, D., An Application of General Regression Neural Network to Nonlinear Adaptive Control. *Proc. Of Fifth European Conf. On Power Electronic and Applications*, Brighton, UK, Sept, Vol.4, p219-224, 1993.
7. Chen, C.H., Neural Networks for Financial Market Prediction, *Proc. Int. Conf. On Neural Networks* Vol.2, June 1994, p1199-1202.
8. Timothy, M., *Advanced Algorithms for Neural Networks: A C++ Sourcebook*, John Wiley & Sons, Inc., Canada, 1995.
9. Specht, D.F., *Fuzzy Logic and Neural Network Handbook : Chapter 3 - Probabilistic and General Regression Neural Networks*, McGraw-Hill Companies, Inc. 1996.
10. Rumelhart D.E. et. al., "Parallel Distributed Processing: Explorations in the Micro Structure of Cognition Vol. I and II", MIT Press, Cambridge, MA, 1986.

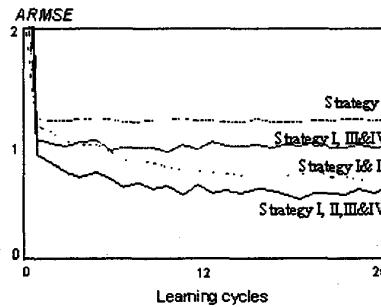


Fig. 4: ARSME for cases of Gaussian Noise of 0.05

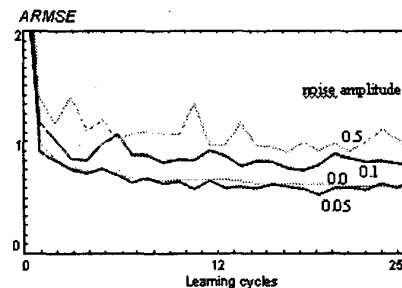


Fig. 5: Performance of GRNN models incorporating Strategies I,II,III & IV under various noise conditions.

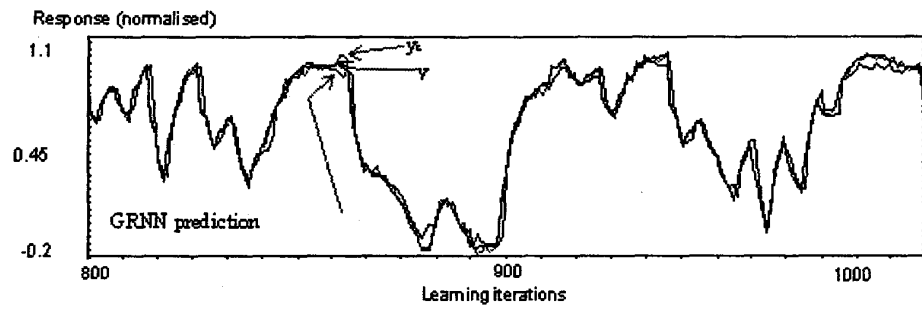


Fig. 6: Performance of the GRNN predictor model of the nonlinear plant with a noise amplitude of 0.05.