

# CONTROLLERS OPTIMIZATION FOR A FLUID MIXING SYSTEM USING METAMODELLING APPROACH

Mohamed Sultan, M.; Shahrum Shah, A. & Osman David, C.

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

E-Mail: sultan\_ali@fke.utm.my; shahrum@utm.my; dauda9@yahoo.com

## Abstract

Offline optimization of controller parameters for complex non-linear processes can be time consuming, even with high performance computers. This paper demonstrates how Metamodelling techniques can be utilized to quickly tune the controller parameters for a non-linear process. The process used in this study is the mixing process which is a multivariable and intrinsically non-linear plant. The Radial Basis Function Neural Network Metamodel founded a good approximation to the optimum controller parameters in this case. This paper proposes an intuitive methodology to use only a small fraction of the design space to create a Radial Basis Function Neural Network Metamodel that is good enough to optimize the system. Comparisons were made between the controllers optimized using the Metamodelling technique and the original large space design.

(Received in May 2008, accepted in October 2008. This paper was with the authors 1 month for 1 revision.)

**Key Words:** Radial Basis Function, Metamodelling, Liquid Mixing Process, Numerical Optimization

## 1. INTRODUCTION

In the design of a control system, one often has a complicated mathematical model of a system that has been obtained from fundamental physics and chemistry. The system usually consists of inputs and outputs. In practice, it is normally desired to find the optimum controller parameter values that would give optimal outputs of the system. These values can be obtained with the help of the Metamodelling.

The simulations required when applying optimization algorithms might be very expensive in terms of computations performed by high performance computers owing to the complexity of the actual model. In spite of the advances in computer technology, the computational time to simulate the actual model might be too long and thus it becomes impractical to rely exclusively on simulation for the purpose of design optimization. Thus Metamodelling is very much needed for the determination of simpler models that involve less computation and gives good approximation results of the complicated model. In this work, a suggestion on how to optimize the controller parameters for a mixing process using Metamodelling techniques is put forward.

## 2. METAMODELLING REVIEW

Recently, Genetic Algorithm is becoming more popular when solving multidimensional optimization problems in different fields of application mainly because of their robustness in finding an optimal solution and ability to provide near optimal solution close to the global minimum [1]. Metamodelling or sometimes called model reduction also has been successfully used in many fields where complex computer models of the actual system exist but running the simulation of the model costs a considerable amount of time. Simulations of models

involving finite element and fluid dynamics analysis and optimization of controllers with many parameters are typical examples of computer programs that require a significant amount of execution time.

In a recent attempt by Samsudin, S. I. [12] to optimize a pole placement controller for a nonlinear plant, the simulation time to search for the optimum pole placement gains took almost 3 full days using a Pentium based computer. In another example, a finite element simulation program solving a microwave passive/active circuit problem took about 8 hours execution time on a Pentium based PC, as reported by Tsai, et al. [6]. With the availability of a simpler model, several design issues such as what-if analysis, prediction of a system output, optimization and verification and validation of simulation models can be done using much less time since computing the output of an optimized Metamodel (say a neural network for example) will just be in a matter of minutes, using an equivalent PC.

As an example, in a test Metamodelling for optimization problem in Rashid [8], training a Fuzzy-Neuro Metamodel on 130 measurement data and the search for its global minimum took only 5.62 minutes of execution time using a Pentium based PC. Hence, although the output of the Metamodel is only an approximate of the actual measurement of the complex model, evaluation of this output value is fast and it usually provides enough information especially during the design phase of a project.

### **3. RADIAL BASIS FUNCTION METAMODEL**

In this paper, a Radial Basis Function Neural Network (RBF NN) was used as the Metamodel to approximate the mapping of the controller gains and the objective function. The radial basis functions were first used to design Artificial Neural Networks in 1988 by Broomhead and Lowe [4]. The architecture of the RBF NN used in this work is illustrated in Fig. 1.

The network consists of three layers: an input layer, a hidden layer and an output layer. Here,  $R$  denotes the number of inputs while  $Q$  the number of outputs. Eq. 1 is used to calculate the output of the RBF NN for  $Q = 1$ :

$$\eta(x, w) = \sum_{k=1}^{S1} w_{1k} \phi(\|x - c_k\|_2) \quad (1)$$

where  $x \in \mathfrak{R}^{R \times 1}$  is an input vector,  $\phi(\cdot)$  is a basis function,  $\|\cdot\|_2$  denotes the Euclidean norm,  $w_{1k}$  are the weights in the output layer,  $S1$  is the number of neurons (and centers) in the hidden layer and  $c_k \in \mathfrak{R}^{R \times 1}$  are the RBF centers in the input vector space. Eq. (1) can also be written as:

$$\eta(x, w) = \phi^T(x)w \quad (2)$$

where:

$$\phi^T(x) = [\phi_1(\|x - c_1\|) \quad \dots \quad \phi_{S1}(\|x - c_{S1}\|)] \quad (3)$$

and:

$$w^T = [w_{11} \quad w_{12} \quad \dots \quad w_{1S1}] \quad (4)$$

The output of the neuron in a hidden layer is a nonlinear function of the distance given by:

$$\phi(x) = e^{-x^2/\beta^2} \quad (5)$$

where  $\beta$  is the spread parameter of the RBF. For training, the least squares formula was used to find the second layer weights while the centers are set using the available data samples.

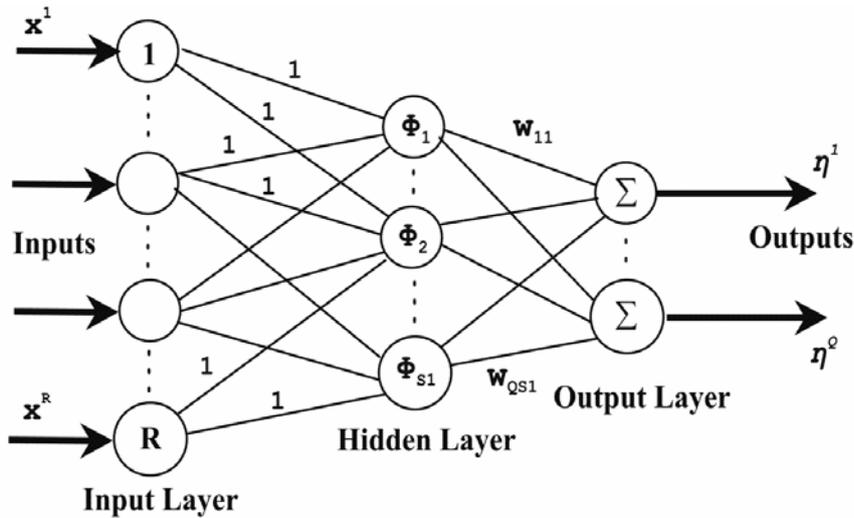


Figure 1: Radial Basis Function Neural Network.

RBF NN offer several advantages compared to the Multilayer Perceptrons. RBF NN has also been successfully used, as reported by [2] and [10]. Two of these advantages are they can be trained using fast 2 stages training algorithm without the need for time consuming non-linear optimization techniques and an ANN RBF possesses the property of ‘best approximation’ [5]. In Metamodelling, RBF NN has also been successfully used, as reported by [2] and [10].

#### 4. MIXING PROCESS

The mixing process studied here was also studied extensively by [3], [7] and [14]. The plant, which is a fluid mixer as shown in Fig. 2, consists of a mixing tank and two auxiliary tanks. The first auxiliary tank contains colored water  $c_1$ , while the second one contains clear water,  $c_2$ . The input flow  $q$  to the mixing tank which contains mixed liquid  $c_0$  is controlled by two valves that regulate the output flows  $q_1$  and  $q_2$  from the auxiliary tanks. The output flow  $q_0$ , taken as a disturbance, has the coloration  $c$  of the resulting mix and is a function of the output pipe cross-section  $ab$ , of the liquid level  $h$  in the mixing tank and of a constant  $C_d$  related to the shape and material of the output pipe.

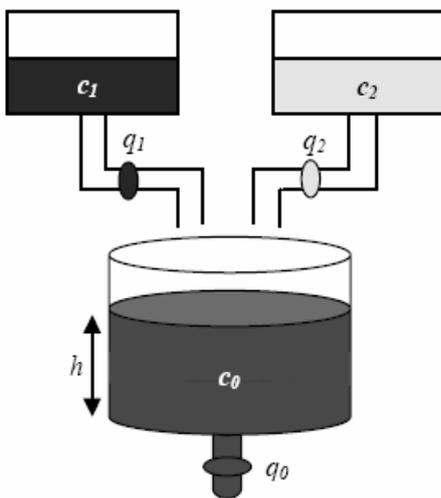


Figure 2: Fluid Mixing Process.

In order to simplify the simulation, it has been assumed that:

- The auxiliary tanks always contain sufficient liquid for the process to keep on running.
- The time necessary to get a uniform mixture is neglected.
- The time delays related to the fluid flow in the pipes are neglected.
- The dynamics of the input valves are neglected.

The mixing tank properties and values used are as given in Table I.

Table I: Mixing Process Variables and Values Used for Simulation.

Variables	Description	Values
$c_0$	Colorations (no unit)	0.5
$c_1$	Colorations (no unit)	1
$c_2$	Colorations (no unit)	0
$q$	Input flow ( $cm^3 s^{-1}$ )	Variable
$q_1, q_2$	Output flows from the auxiliary tanks ( $cm^3 s^{-1}$ )	Variable
$q_0$	Output flow from the mixing tank ( $cm^3 s^{-1}$ )	Variable
$h$	Liquid level in the mixing tank ( $cm$ )	Variable
$h_{mit}$	Initial liquid level in the mixing tank ( $cm$ )	5
$h_{max}$	Maximum allowable height of water in tank ( $cm$ )	100
$ab$	Output pipe cross-section ( $cm^2$ )	0.1
$S$	Area of the liquid surface in the mixing tank ( $cm^2$ )	3600
$C_d$	Constant related to the shape and material of the output pipe.	3
$g$	Gravity acceleration ( $cms^{-2}$ )	980

The following equations model the plant dynamics:

$$q - q_0 = \frac{dV}{dt} = S \frac{dh}{dt} cm^3 s^{-1} \quad (6)$$

$$q = q_1 + q_2 cm^3 s^{-1} \quad (7)$$

Taking into consideration the dynamics of the output valve:

$$q_0 = C_d . ab . \sqrt{2gh} cm^3 s^{-1} \quad (8)$$

where  $g$  is the gravity acceleration,  $V$  is the volume of liquid and  $S$  is the area of the liquid surface in the mixing tank. The color is provided by some ink. The conservation of ink can be expressed by:

$$c_1 q_1 - c_0 q_0 = \frac{d(c_0 Sh)}{dt} = S \left( c_0 \frac{dh}{dt} + h \frac{dc_0}{dt} \right) \quad (9)$$

These equations were modelled using MATLAB<sup>®</sup> SIMULINK<sup>®</sup> as illustrated in Fig. 3. The control system for the fluid mixer is designed in order to maintain the coloration of the output at the desired set-point and the level in the mixer in the neighbouring of its set-point, not allowing it to be empty or to overflow. The output flow is considered as a disturbance.

A common industrial approach in chemical industry where systems tend to be highly nonlinear is to use multiple PID controllers. In this case, a combination of PI and PID controllers are used as shown in Fig. 3 to manipulate the total input flow,  $q$ , and the proportion of ink to the total flow.

$$q_r = \frac{q_1}{q_1 + q_2} \quad (10)$$

thereby controlling  $h$  and  $c_0$ .

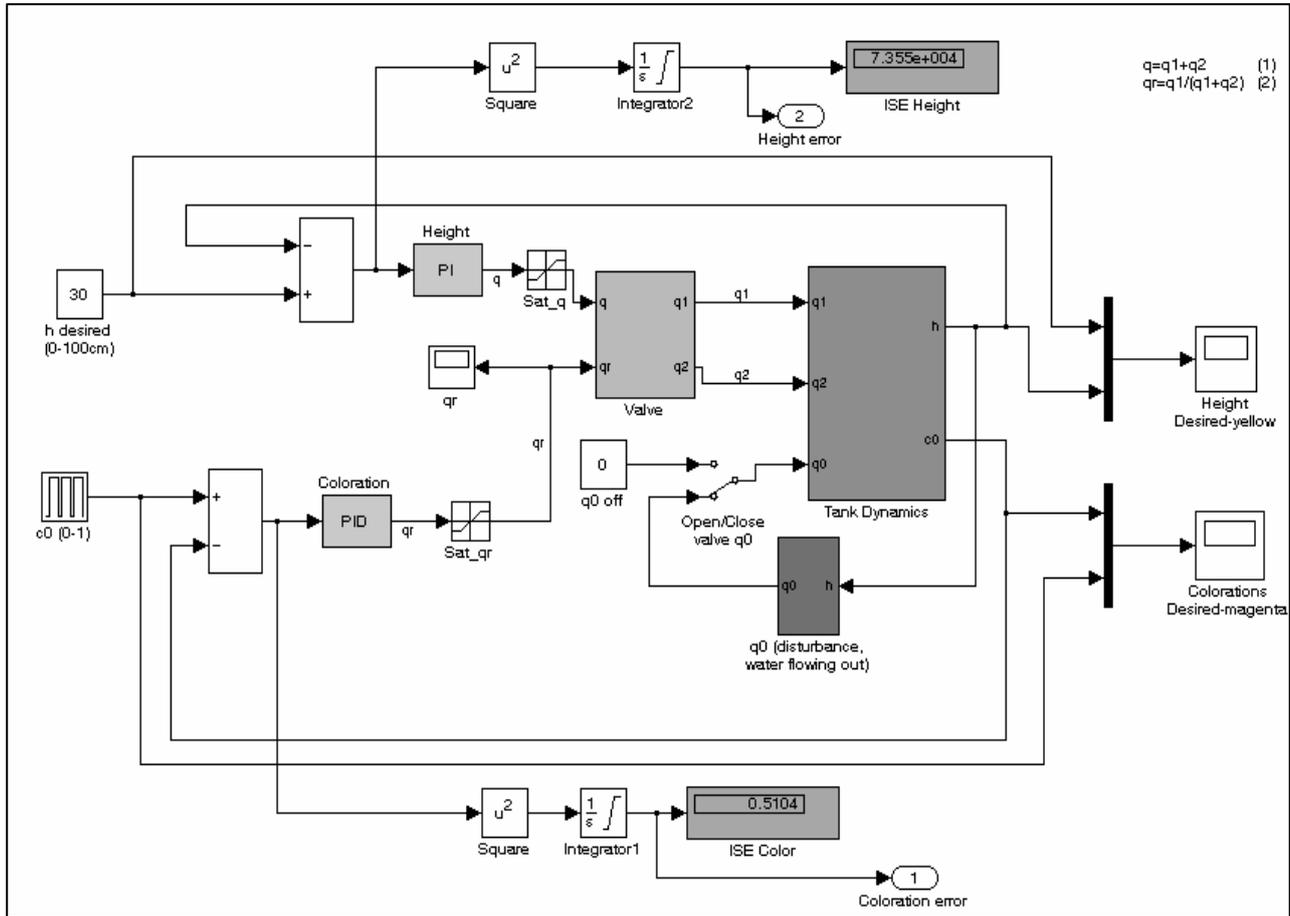


Figure 3: Mixing Process as Implemented in MATLAB® SIMULINK®.

Hence, 5 controller parameters need to be tuned in order to obtain the best performance. These are  $K_{p1}$  and  $K_{i1}$  for the height controller and  $K_{p2}$ ,  $K_{i2}$  and  $K_{d2}$  for the coloration controller. The performance measure that was used in this case was the Integral Square Error (ISE) given by:

$$ISE = \int (y_d(t) - y(t))^2 dt \quad (11)$$

where  $y_d$  is the desired output (set point) while  $y$  is the actual output. This criterion, although not very selective, has been used because of the ease of computing the integral both analytically and experimentally.

## 5. SIMULATION RESULT

Before proceeding with finding the controller parameters, the stability of the system needs to be determined. This has been done in [9] using Taylor expansion and evaluating the eigenvalues of the resulting state space model. It was found out that the system is indeed stable and hence the control of the system is possible.

The approach to optimize the controller parameters is summarized as follows:

1. Define the input design space,  $D$ , which consists of a set of initial values of the controller parameters.
2. Obtain the ISE for both the coloration and height for all the design space defined in  $D$ .
3. Create the target data set,  $T$ , which consists of the normalized ISE for coloration ( $\overline{E}_c$ ) and height ( $\overline{E}_h$ ). Here we use  $\overline{E}$  given by:

$$\overline{E} = \overline{E}_c + \overline{E}_h.$$

4. Fit the RBF NN using  $D$  and  $T$ .
5. Evaluate the RBF NN on a larger input space,  $D'$ .
6. Find the minimum of the RBF NN output (estimated  $\overline{E}$ ). The corresponding controller gains that minimized the RBF output will be the gains to be verified in actual model simulation.
7. Repeat step 1 to 6 should the controller parameter gains are not satisfactory.

In this case,  $D$  and  $D'$  are the sets of discrete values given in Table II. The parameters for the RBF NN that are used to fit the data  $D$  are summarized below:

- 16 RBF centers are used. Centers are added one by one until the RBF NN reaches an error goal of  $0.1$ .
- $\beta=150$ .

The initial data sets need to be properly identified to achieve best approximation by training the Radial Basis Neural Network. If the initial data sets do not cover the maximum and minimum value of the large data sets, the ANN will try to extrapolate which will produce unacceptable results. The initial data sets should not be too small for proper training and should not be too large to minimize the training time. The identified initial and large data sets are as shown in Table II.

The initial data sets are used to obtain the Integral Square Error (ISE) for both height and color by simulation. These ISEs will then be divided by the maximum ISE for coloration and the height respectively and added together to determine the normalized ISE for both outputs. The result of the normalized error against the controller parameter data sets is shown in Fig. 4.

Table II: Initial and Large Data Sets.

Initial Data Sets (D)	
$K_{p1}$	{100, 150, ..., 300}
$K_{i1}$	{0,0.1,0.2}
$K_{p2}$	{100, 150, ..., 300}
$K_{i2}$	{0.001, ...,0.005}
$K_{d2}$	{0.1,0.2}
Total number of data configurations.	750
Large Data Sets (D')	
$K_{p1}$	{100, 120, ..., 320}
$K_{i1}$	{0,0.1,...,0.5}
$K_{p2}$	{100, 120, ..., 320}
$K_{i2}$	{0.001, ...,0.005}
$K_{d2}$	{0.1,0.2}
Total number of data configurations.	8640

The normalized ISE is used to train the Radial Basis Function Network which will then be used as the Metamodel of the Fluid Mixing System to evaluate the ISE for the corresponding large data sets of the controller parameters. In the training stage of the Radial Basis Function Neural Network, the error goal is set at 0.1. The training curve is shown in Fig. 5.

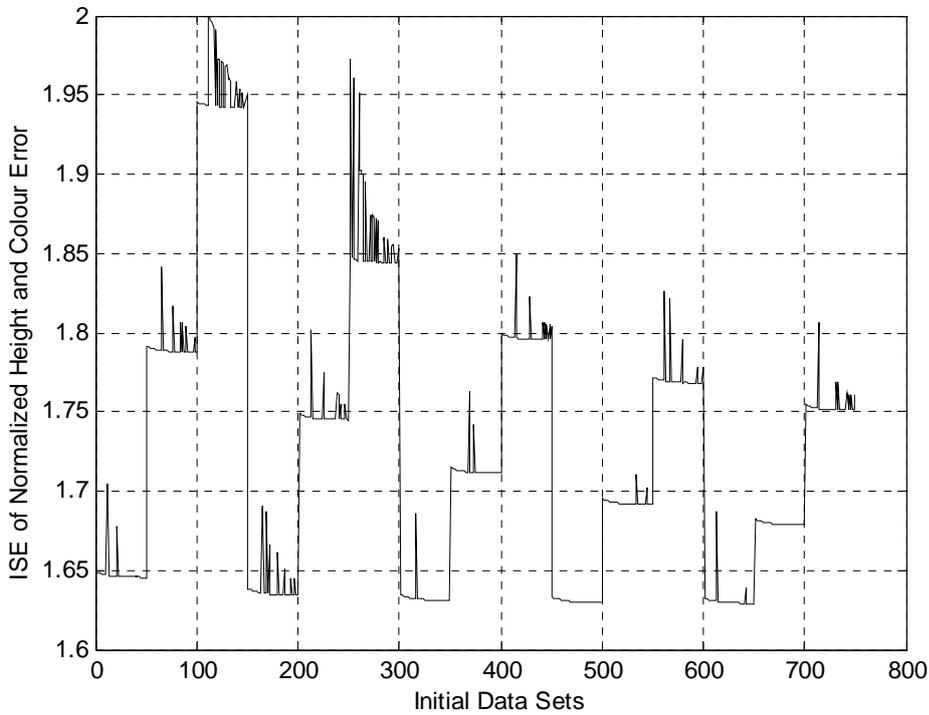


Figure 4: Normalized Height and Color ISE for Initial Data Sets.

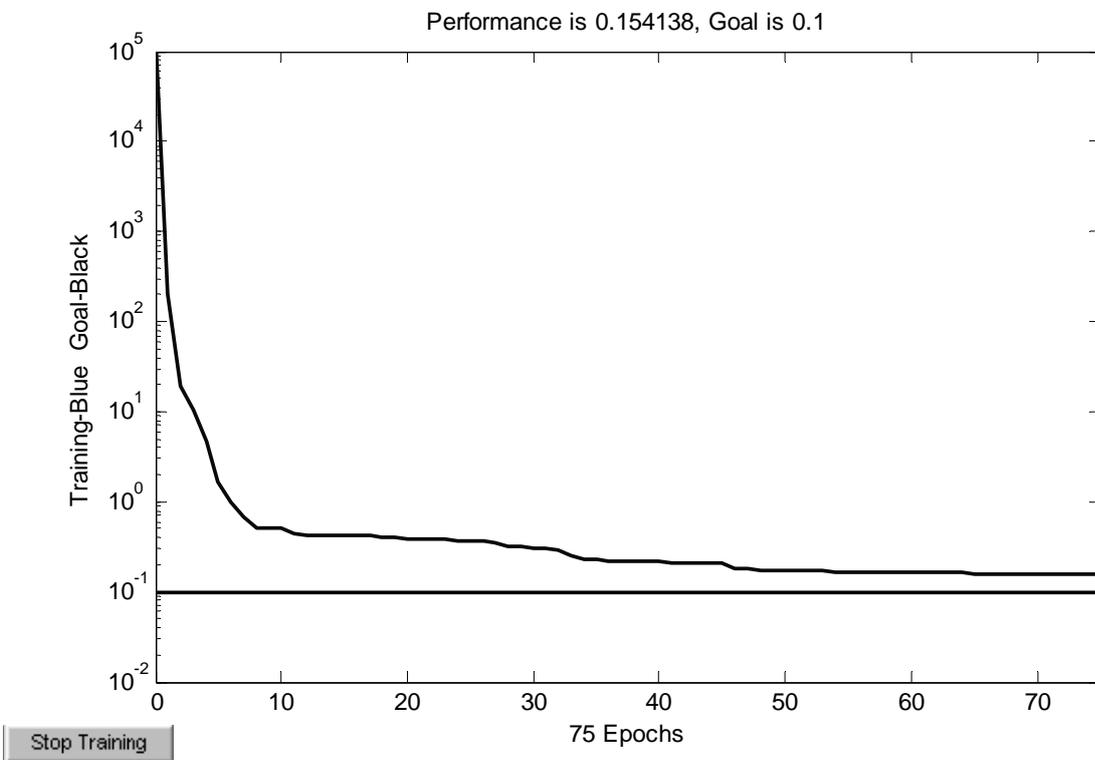


Figure 5: Training of RBF-NN using PI and PID Controller Input Set

From the simulation results of the Radial Basis Function Neural Network, it is clearly shown that only 75 epochs are required to achieve the set goal of error which is 0.1. The smaller is the error goal; the better will be the approximation. However, if the goal is very tight, the training process will take longer time, which could be a disadvantage. The spread value of 150 is used in the training process. This value has to be adjusted to achieve better response. The larger the spread of the data, the smoother will be the function approximation. A large spread implies a lot of neurons will be required to fit a fast changing function. Whereas, a small a spread means less neurons will be required to fit a smooth function, and the network may not generalize well.

After the training stage, Radial Basis Function Neural network is used to perform the simulation for large data space controller parameters sets which consists of 8640 data sets. The result is shown in Fig. 6.

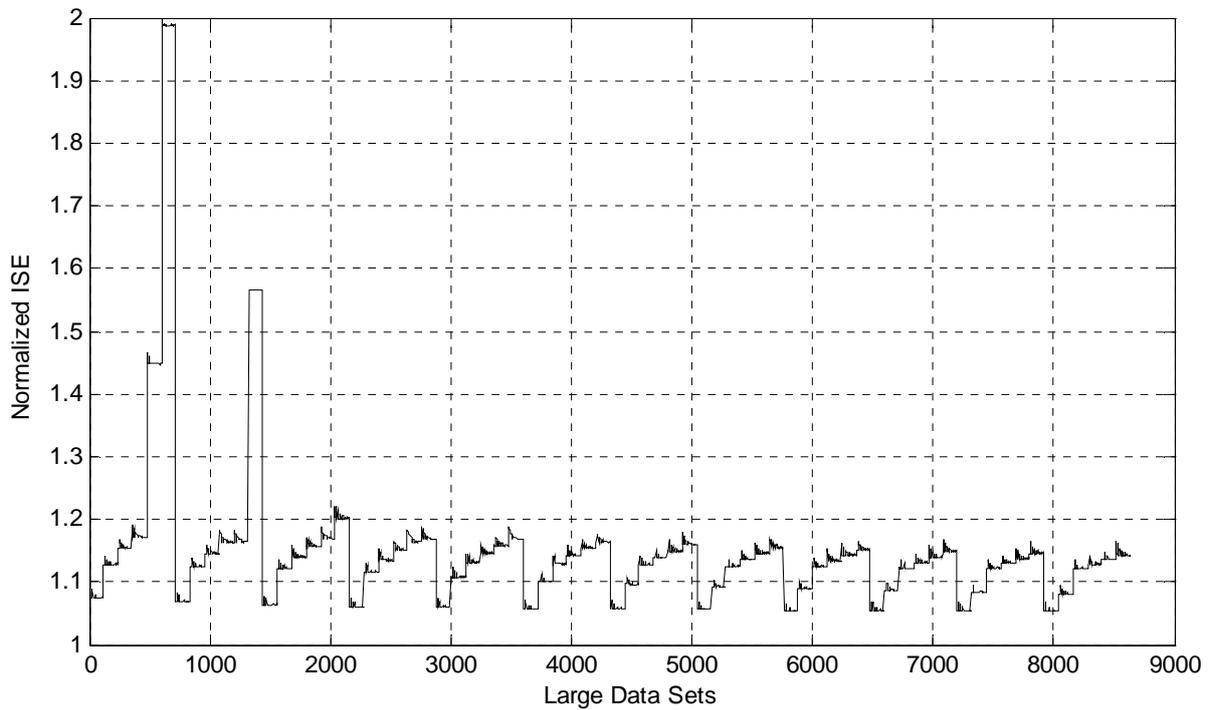


Figure 6: Metamodel output for Large Data Sets of PI and PID Parameters.

Using MATLAB<sup>®</sup> on an INTEL<sup>®</sup> Core<sup>™</sup> 2 Duo PC, it took exactly **1.78** minutes to complete steps 1 to 6 as explained in the previous section. The best controller gain that minimizes the normalized ISE ( $\bar{E}$ ) is given in Table III:

Table III: Best PI and PID Controller Parameter using Metamodel.

Best gains based on RBF NN Metamodel ( <b>1.78</b> minutes)	
$K_{p1}$	300
$K_{i1}$	0
$K_{p2}$	300
$K_{i2}$	0.005
$K_{d2}$	0.2

To verify the Metamodel, the actual SIMULINK model was evaluated for all the 8640 cases in large space data sets ( $D'$ ) using the same PC and the normalized ISE ( $\bar{E}$ ) was also computed. The result is then compared with the actual simulation result as illustrated in Fig. 7.

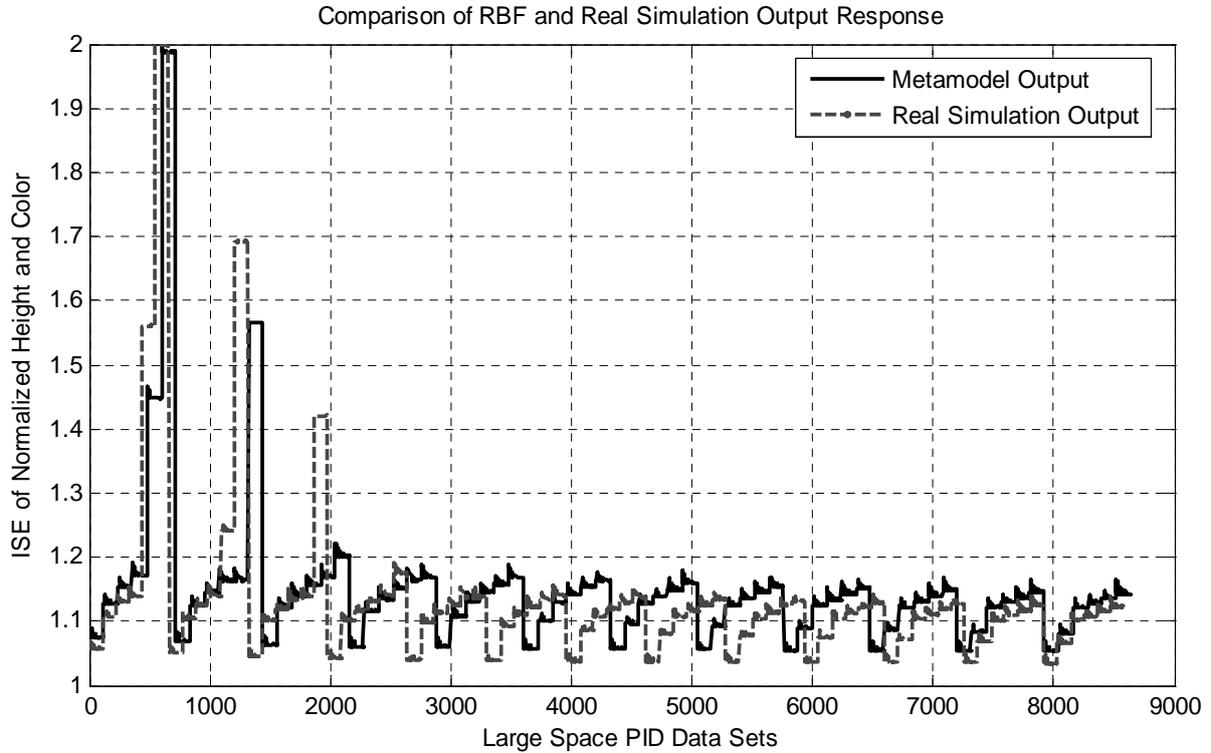


Figure 7: Comparison of Metamodel and Actual Simulation Output for PI and PID Controller.

The simulations took **19.9** minutes to complete, the gain that minimizes  $\bar{E}$  are as given in Table IV.

Table IV: Best PI and PID Controller Parameter using Actual Simulation.

Best gains based on actual evaluations ( <b>19.9</b> minutes)	
$K_{p1}$	320
$K_{i1}$	0
$K_{p2}$	280
$K_{i2}$	0.005
$K_{d2}$	0.1

From the simulation results in Fig. 7 and Table IV, the results obtained by using Metamodel are almost equal to that of using the actual simulation. However, the time of the simulation and the Metamodel differs. The Metamodel only took 1.78 minute to evaluate 8640 input sets compared to 19.9 minutes by actual simulation for the same input sets.

Table V show a comparison of the results obtained, it can be observed that the Metamodel managed to approximate the global minimum of the error curve fairly well. Although the minimum of the normalized error for the Metamodel and the actual model is different (1.6 vs. 1.1), the actual height and coloration error using the parameter values that minimized the errors are the same.

Table V: Metamodel and Actual Result Comparison for PI and PID Controller.

	Metamodel	Actual
$\min(\bar{E})$	1.6293	1.0539
Actual $E_h$ value at $\min(\bar{E})$	$7.35 \times 10^4$	$7.35 \times 10^4$
Actual $E_c$ value at $\min(\bar{E})$	0.514	0.514

The response of the height and coloration based on these controller values are plotted in Fig. 8 and 9.

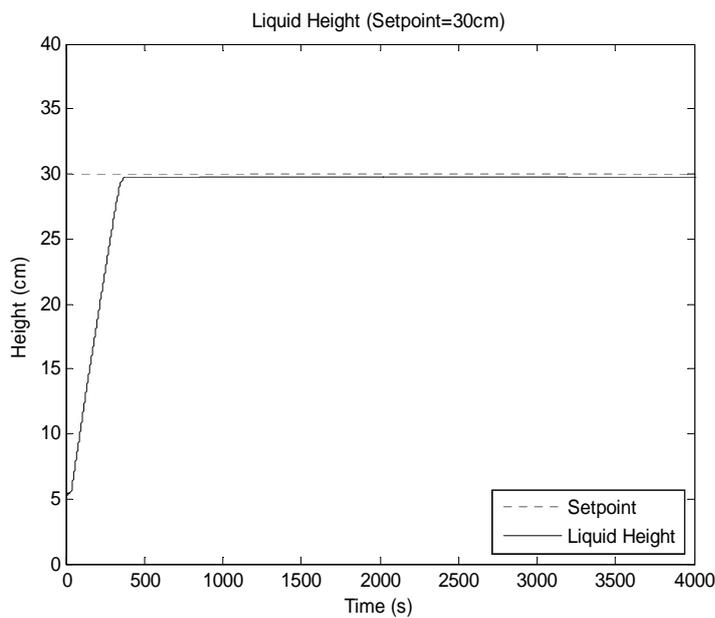


Figure 8: Response of Liquid Height.

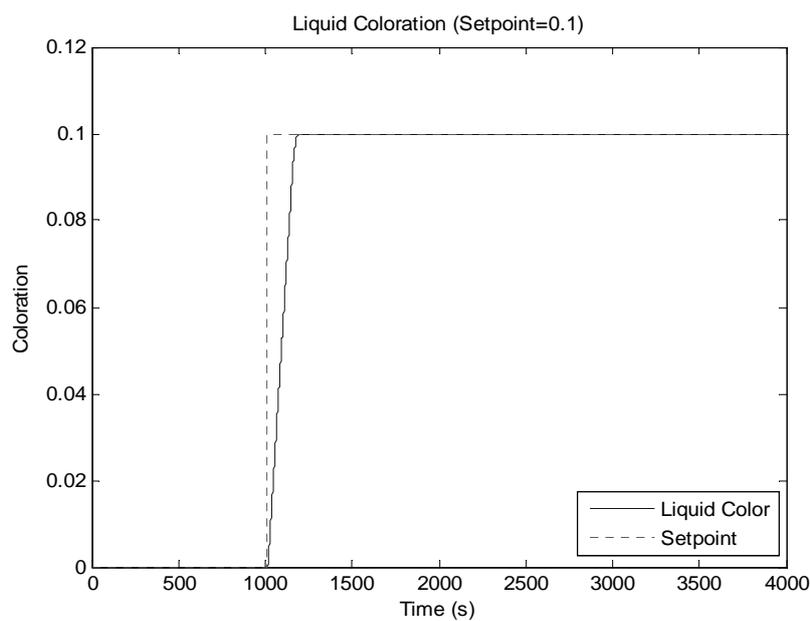


Figure 9: Response of Liquid Coloration.

The controller gave a good response for a height control with settling time of 361 seconds and rising time of 287 seconds. The transient response has no overshoot and that's near to the critical damp. Besides, it also gives a good steady state error of 0.094. Thus, controller parameter obtained for PI controller has an overall good performance.

Fig. 9 shows the simulation results for PID controller which is optimized using Metamodelling technique. As can be observed, this controller has a good capability to control the liquid coloration in fluid mixing system. The response followed the changes of the set point at faster rate. The settling time took only short time to stabilize in its new final value (i.e.  $T_s = 190$  second for 0.1 set point). Both height and coloration output gives a critically damped response.

## **6. CONCLUSION**

RBF NN has proven its effectiveness as a method of controller optimization in this paper. RBF NN has been successfully used to optimize both conventional and intelligent controller. It is able to give quick computational time to obtain best value for the controller (i.e. about 1 minute) as compared to finding the best value by simulating the process for all values in large input space ( $D'$ ) (about 30 minutes). The Metamodelling approach shown is clearly a useful approach and this will become more significant for a larger  $D$  or for a more complicated problem.

Hence, as described earlier, Metamodelling does provide the designer with a quick estimate for a set of good parameters to begin with. Further simulations on the actual system can be done if better parameter values are required. In this example, the data set  $D$  was created simply by choosing the input values in a grid like fashion, based on background knowledge of the problem. A more intuitive approach is to start of with a small number of samples, and then sequentially add more data samples 'intelligently' employing Experimental Design techniques. An investigation on the possible use of the Worst Case Approach [13] and the Cross Validation technique [11] in data sampling would seems be the appropriate next step forward. It is envisaged that a more strategic data location will allow the creation of a more accurate Metamodel using less data, and therefore, less time is required to estimate the best controller parameters.

## **7. ACKNOWLEDGEMENT**

The authors would like to express their gratitude to the Universiti Teknologi Malaysia and Ministry of Science, Technology and Innovation for providing the facilities and financial assistance to conduct this research. The authors would also like to thank the reviewers for their valuable comments.

## **REFERENCES**

- [1] Palmar, G.; Mukarjee, S.; Prasad, R. (2007). Reduced Order Modeling of Linear MIMO System using Genetic Algorithm, *International Journal of Simulation Modelling*, Vol. 6, No. 3, 173-184
- [2] Mullur, A. A; Messac, A. (2005). Extended Radial Basis Functions: More Flexible and Effective Metamodeling, *AIAA Journal*, Vol. 43, No. 6, 1306-1315
- [3] Lanas, I.; Mota, R.; Tanscheit, G; Vellasco, M.; Barreto, J. (1999). Neuro Fuzzy Control of a Multivariable Nonlinear Process, *VIII<sup>th</sup> International Fuzzy Systems Association World Congress (IFSA '99)*, Taiwan
- [4] Broomhead, D. S.; Lowe, D. (1988). Multivariable functional interpolation and adaptive networks, *Complex Systems*, Vol. 2, 321-355

- [5] Girosi, F.; Poggio, T. (1990). Networks and the best approximation property, *Biological Cybernetics*, Vol. 63, 169-176
- [6] Tsai, H.; Wang, Y.; Itoh, T. (2002). An Unconditionally Stable Extended (USE) Finite-Element Time-Domain Solution of Active Nonlinear Microwave Circuits Using Perfectly Matched Layers, *IEEE Transactions on Microwave Theory and Techniques*, Vol. 50, No. 10, 2226-2232
- [7] Barreto, J.; De Neyer, N.; Gorez, R. (1991). Fuzzy control of a nonlinear plant: the case of a fluid mixer, *Proceedings of the 6<sup>th</sup> Mediterranean Electrotechnical Conference*, Vol. 2, 807-811
- [8] Rashid, K. (2000). *Optimization in Electromagnetics using Computational Intelligence*, Ph.D. thesis, Imperial College of Science, Technology and Medicine, Department of Electrical and Electronics Engineering
- [9] Mohamed Ali, M. S. (2008). *Optimization of the Controller Parameters for a Fluid Mixing System*, Master Project Report, Fakulti Kejuruteraan Elektrik, Universiti Teknologi Malaysia
- [10] Dyn, N.; Levin, D.; Rippa, S. (1986). Numerical Procedures for Surface Fitting of Scattered Data by Radial Basis Functions, *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, No. 2, 639-659
- [11] Jin, R.; Chen, W.; Sudjianto, A. (2002). On Sequential Sampling for Global Metamodeling in Engineering Design, *Proceedings of Design Engineering Technical Conference and Computers and Information in Engineering Conference*, Montreal
- [12] Samsudin, S. I. (2006). *Modern and Intelligent Control of Active Magnetic Bearing*, M.Sc. Thesis, Faculty of Electrical Engineering Universiti Teknologi Malaysia
- [13] Abdullah, S. S. (2003). *Experiment Design for Deterministic Model Reduction (Metamodeling) and Neural Network Training*, Ph. D. thesis, Imperial College of Science, Technology and Medicine, Department of Electrical and Electronics Engineering
- [14] Wang, Y.; Deng, H.; Chen, Z. (2005). Adaptive Fuzzy Logic Controller with Rule-based Changeable Universe Of Discourse for a Nonlinear MIMO System, *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*