Hindawi Publishing Corporation
Modelling and Simulation in Engineering
Volume 2009, Article ID 234529, 8 pages
doi:10.1155/2009/234529

Review Article

Evolutionary Tuning Method for PID Controller Parameters of a Cruise Control System Using Metamodeling

M. N. Ab Malek and M. S. Mohamed Ali

Faculty of Electrical Engineering, Universiti Teknologi Malaysia (UTM), Johor Darul Takzim, 81310 Skudai, Malaysia

Correspondence should be addressed to M. N. Ab Malek, zal_lzura@yahoo.com

Received 11 April 2009; Accepted 8 December 2009

Recommended by Soib Taib

For long time the optimization of controller parameters uses the well-known classical method such as the Ziegler-Nichols and the Cohen-Coon tuning techniques. Despite its effectiveness, these off-line tuning techniques can be time consuming especially for a case of complex nonlinear system. This paper attempts to show a great deal on how Metamodeling techniques can be utilized to tune the PID controller parameters quickly. Note that the plant use in this study is the cruise control system with 2 different models, which are the linear model and the nonlinear model. The difference between both models is that the disturbances were taken into consideration for the nonlinear model, but in the linear model the disturbances were assumed as zero. The Radial Basis Function Neural Network Metamodel is able to prove that it can minimize the time in tuning process as it is able to give a good approximation to the optimum controller parameters in both models of this system.

1. Introduction

It is crucial to tune the controller parameters wisely for any control system in order to obtain the optimum results for its output. That is, the system output response should be as similar as possible due to the corresponding reference.

There were several techniques that commonly used to tune the controller parameters. For example, the Ziegler-Nichols tuning technique, the Cohen-Coon tuning technique, the tuning based on process response, the Damped Oscillation tuning method and so on. Each technique has its own pros and cons. But, for sure these conventional techniques consume lot of time as the tuning process need to be tuned again and again until the best response is met.

Usually, computer simulation is used to model the real complicated system to save cost and time. In spite of the advances in computer technology, the required time to simulate the actual model might still be long and thus it becomes impractical to rely exclusively on simulation for the purpose of design optimization. To overcome this timing issue, the Metamodeling techniques offer good approximation to the actual model due to the usage of simpler model, added on with less computation algorithms.

This paper illustrates how to optimize the PID controller parameters for both models (linear and nonlinear) of the cruise control system.

2. Metamodeling Review

Metamodeling or sometimes called as "model of the model" has been used extensively in many fields to give simpler model of the input and output function that approximates the relationship between system performances and controller parameters of a system. The required significant set of data for each PID controller parameter that fit the actual set of data will give the best results of approximation.

Recently, as studied in [1], Metamodeling had been used to optimize various type of system, included the nonlinear system. Some of the system that were successfully optimized using the Metamodeling technique are the fluid mixing system, the Cartesian coordinates control of hovercraft system and the flexible robot manipulator. Through their study, they also had proved that the Metamodeling technique can optimize various types of controller parameters, for example, the fuzzy logic controller and the PID controller.

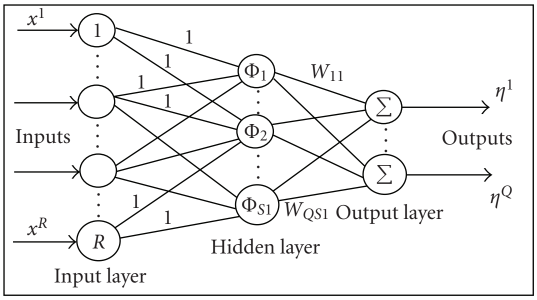

Figure 1: Radial Basis Function Neural Network.

Table 1: Cruise control system variables and values used for simulation.

Variables	Description	Values
τ	Time constant (s)	0.2
T	Real time (s)	1
M	Mass of the vehicle (kg)	1500
$F_{d\ \min}$	Drive force minimum limit (N)	−3500
$F_{d\ \max}$	Drive force maximum limit (N)	+3500
F_a	Air drag force (N)	Variable
F_g	Gravitational force (N)	Variable
C_1	Constant related to engine block (no unit)	743
C_a	Constant related to air-drag force ($\mathrm{Nm^{-1}s^2}$)	1.19
G	Constant related to gravitational force ($\mathrm{ms^{-2}}$)	9.81
v_w	Variable related to wind disturbance	Variable
U	Variable related to desired speed	Variable
θ	Variable related to gravity disturbance	Variable

In another example, in a test Metamodeling for optimization problem in Rashid [2], 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 Neural Network

The RBFs were first used to design Artificial Neural Networks in 1988 by Broomhead and Lowe [3]. Past works reported that the roots of RBF are entrenched in much older pattern recognition techniques as for example potential functions, clustering, functional approximation, spline interpolation and mixture model [4].

In this study, it was used as the Metamodel for matching process between the input-output of the cruise control system. The architecture of the RBF-NN used in this study is illustrated in Figure 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. Equation (1) is used to calculate the output of the RBF NN for $Q = 1$,

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

where $x \in \Re^{R\times 1}$ is an input vector, $\varnothing(\cdot)$ 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 $c_k \in \Re^{R\times 1}$ are the RBF centers in the input vector space. Equation (1) can also be written as,

$$\eta(x, w) = \varnothing^T(x) w, \tag{2}$$

where:

$$\begin{gathered}\varnothing^T(x) = [\varnothing_1(\|x - c_1\|) \cdots \varnothing_{S1}(\|x - c_{S1}\|)], \\ \varnothing(x) = e^{-x^2/\beta^2},\end{gathered} \tag{3}$$

where β 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.

RBF NN offer several advantages compared to the Multilayer Perceptrons. RBF-NN has also been successfully applied to a large diversity of applications including interpolation [3, 5], chaotic time series modeling [6], system identification, control engineering [7], electronic device parameter modeling, channel equalization [8–10], speech recognition [8, 11], image restoration [12], shape from shading [13], 3-D object modeling [5, 14], motion estimation and moving object segmentation [15], data fusion [16], and so forth, The advantages of RBF are they can be trained using fast two stages training algorithm without the need for time consuming nonlinear optimization techniques and an ANN RBF possesses the property of "best approximation" [17]. In Metamodeling, RBF NN has also been successfully used, as reported by [18, 19].

4. Cruise Control System

This section presents the mathematical equation that represents both linear and nonlinear model of the cruise control system.

In this paper, all equations were modeled base on the block diagram given by Dorf and Bishop [20] as shown in Figure 2. The cruise control system properties and values used are given in Table 1.

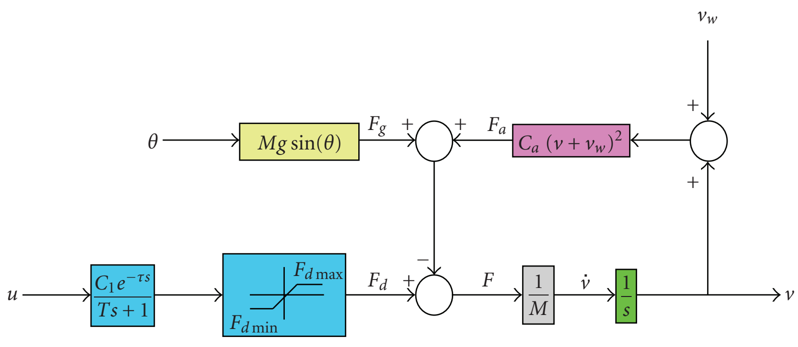

FIGURE 2: A block diagram by Dorf and Bishop [20].

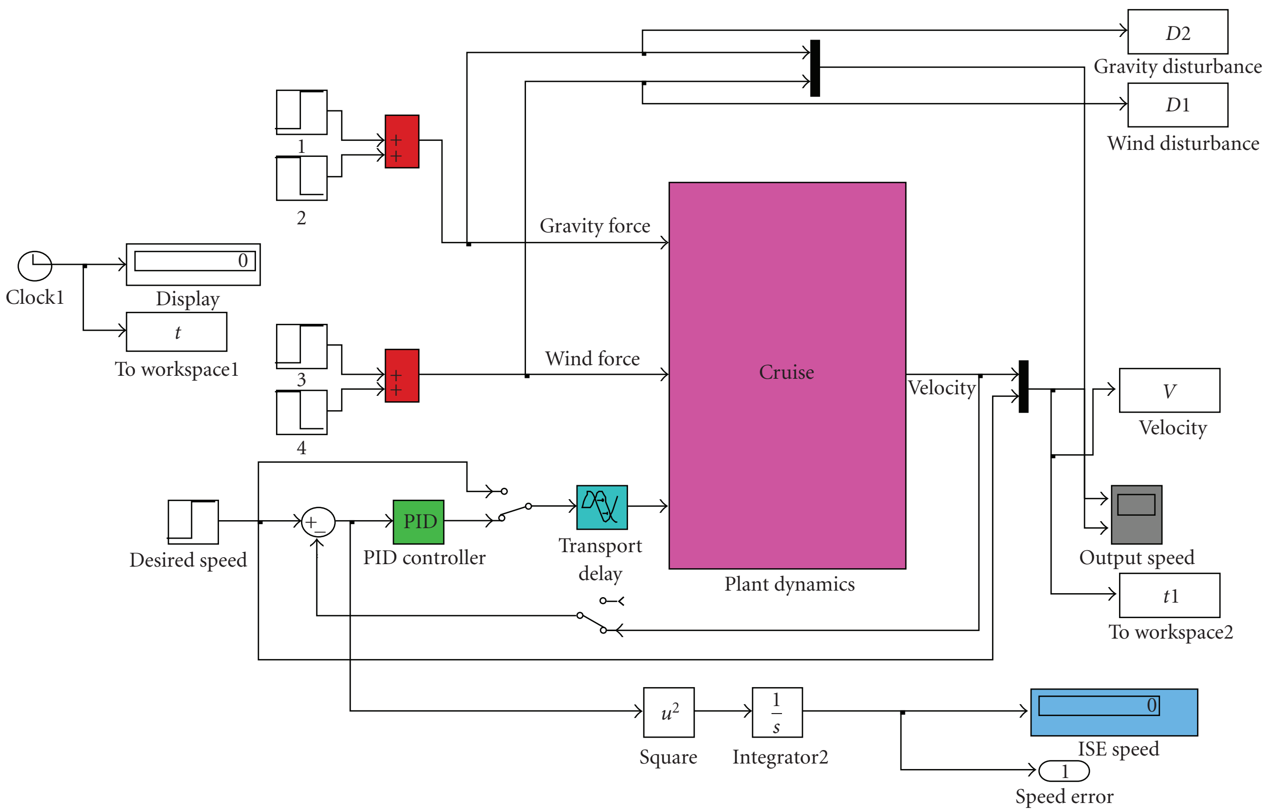

FIGURE 3: MATLAB SIMULINK block diagram for the nonlinear model of cruise control system using PID controller.

TABLE 2: Initial and large data sets.

Initial and Large Data Sets			
	Model type	Nonlinear	Linear
	K_{p1}	$\{1.5, 0.1, \ldots, 2.0\}$	$\{1.525, 0.1, \ldots, 2.0\}$
D	K_{i1}	$\{0.0, 0.15, \ldots, 0.3\}$	$\{0.0, 0.15, \ldots, 0.3\}$
	K_{d1}	$\{0.0, 0.25, \ldots, 1.0\}$	$\{0.0, 0.055, \ldots, 1.0\}$
	Total number of data configuration	90	285
	K_{p1}	$\{1.5, 0.025, \ldots, 2.0\}$	$\{1.5, 0.025, \ldots, 2.0\}$
D'	K_{i1}	$\{0.0, 0.01, \ldots, 0.3\}$	$\{0.0, 0.01, \ldots, 0.3\}$
	K_{d1}	$\{0.0, 0.025, \ldots, 1.0\}$	$\{0.0, 0.025, \ldots, 1.0\}$
	Total number of data configuration	26691	26691

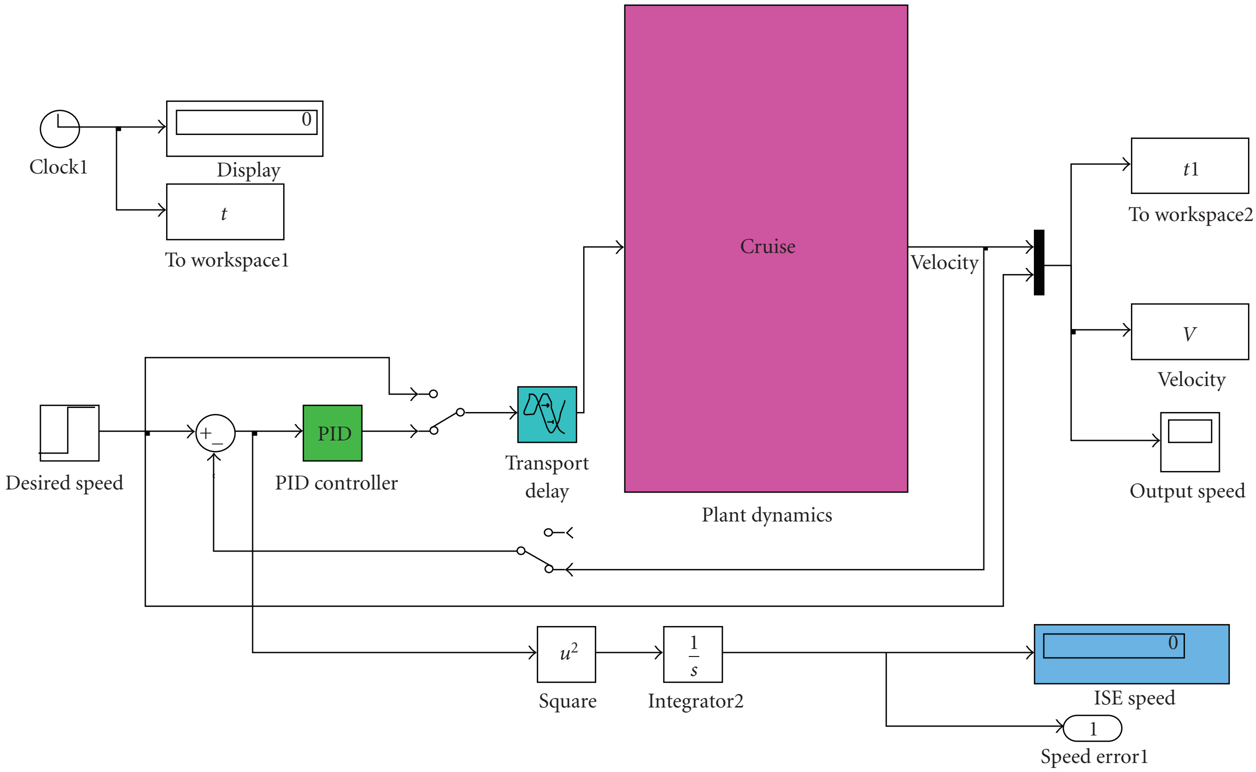

Figure 4: MATLAB SIMULINK block diagram for the linear model of cruise control system using PID controller.

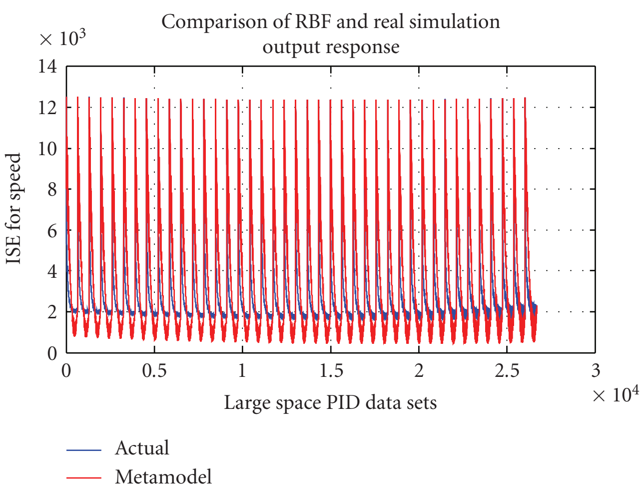

Figure 5: Comparison of metamodel and actual simulation output for nonlinear model.

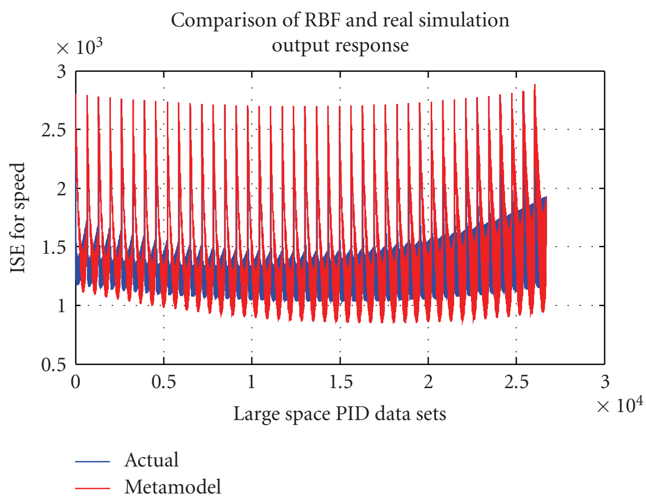

Figure 6: Comparison of metamodel and actual simulation output for linear model.

The following equations model the plant dynamics of nonlinear system:

$$\dot{v} = \frac{1}{M}\left(F_d - F_a - F_g\right), \tag{4}$$

$$\dot{v} = \frac{1}{M}\left(F_d - C_a v^2 - 2C_a v v_w - C_a {v_w}^2 - (-mg\cos\theta)\right), \tag{5}$$

$$\dot{F_d} = \frac{1}{T}\left(C_1 u(t-\tau) - F_d\right). \tag{6}$$

The nonlinear model includes the disturbances parameter into account, therefore all terms include the gravitational force (F_g) and the wind disturbance variable (v_w) were used in (4), (5), and (6). These disturbances are actually opposing forces that reduced the forward force that accelerates the car. Also note that in this paper, the wind gust was assumed to be in the front direction only. Hence, it was calculated into the error signal together with the output signal.

Table 3: Best gains for the cruise control system.

Process	Metamodel		Actual	
Model type	Nonlinear	Linear	Nonlinear	Linear
K_{p1}	1.5	1.5	1.5	1.5
K_{i1}	0.23	0.22	0.3	0.09
K_{d1}	0.75	0.65	0.75	0.55

Take a glance at (5) above. There is squared terms in that equation. To eliminate the squared term in (5), equations (5) and (6) were differentiated both left and right sides to yield:

$$\frac{d}{dt}\dot{v} = \frac{1}{M}(-2C_a v\delta v + \delta F_d - 2C_a\delta v\delta v_w - 2C_a v_w\delta v_w - mg\sin\theta\delta\theta), \qquad (7)$$

$$\frac{d}{dt}\dot{F}_d = \frac{1}{T}(C_1\delta u(t-\tau) - \delta F_d).$$

The inverse Laplace Transform will facilitate the solution of this plant to be modeled in SIMULINK model. Hence, taking the inverse Laplace Transform of (7) yields:

$$s\Delta V(s) - v(0) = \frac{1}{M}\begin{pmatrix} -2C_a v\Delta V(s) + \Delta F_d(s) - 2C_a\Delta V(s)\Delta V_w(s) \\ -2C_a v_w\Delta V_w(s) - mg\sin\theta\Delta\theta(s)\end{pmatrix},$$

$$s\Delta F_d(s) - F_d(0) = \frac{1}{T}(C_1\Delta U(s)e^{-\tau s} - \Delta F_d(s)). \qquad (8)$$

Lastly by equating all initial conditions for (8) to zero, the complete equation that model the nonlinear model become:

$$s\Delta V(s) = \frac{1}{M}\begin{pmatrix} -2C_a v\Delta V(s) + \Delta F_d(s) - 2C_a\Delta V(s)\Delta V_w(s) \\ -2C_a v_w\Delta V_w(s) - mg\sin\theta\Delta\theta(s)\end{pmatrix},$$

$$s\Delta F_d(s) = \frac{1}{T}(C_1\Delta U(s)e^{-\tau s} - \Delta F_d(s)). \qquad (9)$$

In contrast, the linear model did not consider all disturbances parameter into the equation. That is, all terms involving the disturbances parameter such as the gravitational force (F_g) and the wind disturbance variable (v_w) were eliminated. The following equations model the plant dynamics of linear system:

$$\dot{v} = \frac{1}{M}(F_d - C_a v^2),$$
$$\dot{F}_d = \frac{1}{T}(C_1 u(t-\tau) - F_d). \qquad (10)$$

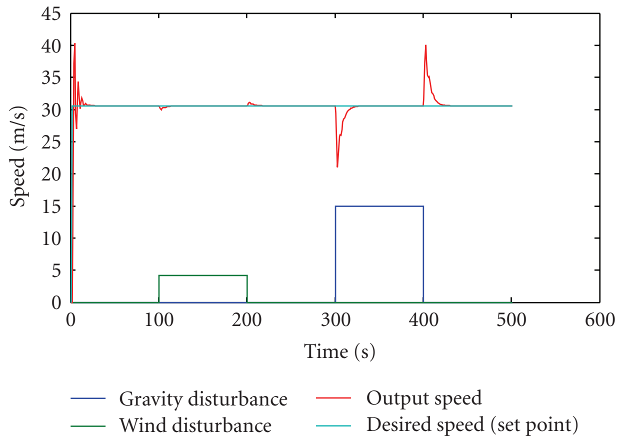

Figure 7: Response of speed for nonlinear model.

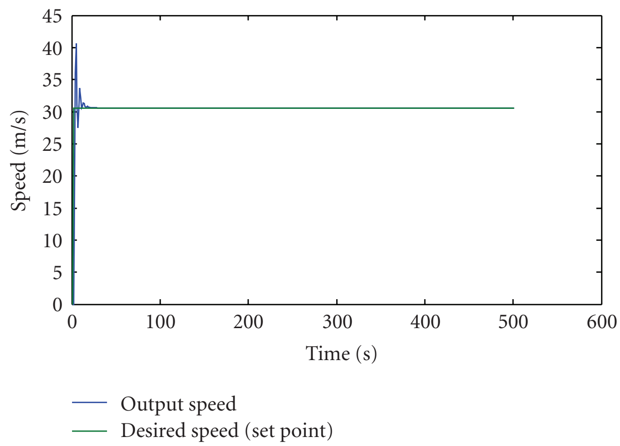

Figure 8: Response of speed for linear model.

Same procedure as before, the nonlinearity functions need to be converted to the linear function. Therefore, (10) above were differentiated to obtain (11) as below:

$$\frac{d}{dt}\dot{v} = \frac{1}{M}(-2C_a v\delta v + \delta F_d),$$
$$\frac{d}{dt}\dot{F}_d = \frac{1}{T}(C_1\delta u(t-\tau) - \delta F_d). \qquad (11)$$

Then, taking inverse Laplace Transform of (11):

$$s\Delta V(s) - v(0) = \frac{1}{M}(-2C_a v\Delta V(s) + \Delta F_d(s)),$$
$$s\Delta F_d(s) - F_d(0) = \frac{1}{T}(C_1\Delta U(s)e^{-\tau s} - F_d(s)). \qquad (12)$$

By setting all initial conditions to zero, the transfer function is as follow:

$$s\Delta V(s) = \frac{1}{M}(-2C_a v\Delta V(s) + \Delta F_d(s)),$$
$$s\Delta F_d(s) = \frac{1}{T}(C_1\Delta U(s)e^{-\tau s} - \Delta F_d(s)). \qquad (13)$$

Table 4: Simulated time for tuning process.

Process	Metamodel		Actual	
Model type	Nonlinear	Linear	Nonlinear	Linear
Time taken	0.4172 min	0.57318 min	52.6874 min	40.8492 min

Table 5: System response characteristics for nonlinear model.

Disturbances type	Characteristics	Actual	Metamodel
Wind force	T_p	103s–203s	103s–203s
	T_s	110s–214s	111s–219s
	%OS	−1.93%–1.93%	−1.58%–1.86%
Gravity force	T_p	303s–403s	303s–403s
	T_s	317s–418s	332s–434s
	%OS	−31.02%–30.99%	−31.11%–31.12%

Table 6: System response characteristics for linear model.

Characteristics	Actual	Metamodel
T_p	5s	5s
T_s	32s	32s
%OS	26.9%	32.92%

Table 7: Best PID controller parameter using actual simulation.

Best gains based on actual simulation		
Model type	Nonlinear	Linear
K_{p1}	1.5	1.5
K_{i1}	0.3	0.09
K_{d1}	0.75	0.55

Table 8: Time taken to optimize PID controller parameter using actual simulation.

Simulated time based on Actual Simulation	
Model type	
Nonlinear	Linear
52.6874 minutes	40.8492 minutes

Table 9: Metamodel and Actual Result Comparison for PID Controller.

Model type	Nonlinear		Linear	
Category	Metamodel	Actual	Metamodel	Actual
min ($\overline{E}$)	1651	1573	1063	1031

The transfer function can be obtained by eliminating $\Delta F_d(s)$ and solving above equations for the ratio of $\Delta V(s)/\Delta U(s)$.

$$\frac{\Delta V(s)}{\Delta U(s)} = \frac{(C_1/MT)(1/\tau)}{(s+2C_a v/M)(s+1/T)(s+1/\tau)} \tag{14}$$

All mathematical equations for nonlinear model and linear model at which were obtained above were modeled in SIMULINK model. This can be illustrated by Figures 3 and 4, respectively.

Note that, the PID controller parameters used in this study are K_{p1}, K_{i1}, and K_{d1} and the performance measure that was used in this case was the Integral Square Error (ISE) given by:

$$\text{ISE} = \int (y_d(t) - y(t))^2 dt, \tag{15}$$

where y_d is the desired output speed (set point) while y is the actual output speed.

5. Simulation Results

The Taylor Series Expansion is utilized to check the system's stability. It should be done before the tuning process started. Referring to the eigenvalues obtained, it shows that the system is indeed stable and hence the control of the system should be possible.

The tuning procedures for Metamodeling technique are listed below:

(1) Define the input design space, D, which consists of a set of initial values of the controller parameters.

(2) Obtain the ISE for speed parameter for all the design space defined in 1.

(3) Create the target data set, T, which consists of the ISE for speed ($\overline{E_s}$).

(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 2. The parameters for the RBF NN used to fit the data D is summarized as:

(i) 16 RBF centers are used. Centers are added one by one until the RBF NN reaches an error goal of 0.1.

(ii) $\beta = 150$.

The limitation of this study is the use of MATLAB on an INTEL PENTIUM M PC to simulate the whole process of optimization. For verification purpose, the actual simulation had been done on D', which consist of 26691 data. The result of actual simulation is then compared with the Metamodeling techniques for each model as illustrated in Figures 5 and 6.

The best controller gain that minimized the error or noted as ISE ($\overline{E}$) for both approaches are given in Table 3. Meanwhile, the time taken to complete the tuning process is noted down as presented in Table 4.

Using the optimized PID gains obtained by Metamodeling technique, the results of the system output response are sketched in Figures 7 and 8.

6. Discussion

Tables 5 and 6 show a comparison of the results in terms of system characteristics. From the results obtained, it can be observed that the adoption of PID controller using the Metamodeling tuning technique can give an almost similar and good response to the system. The effect of each controller K_p, K_i and K_d on a closed-loop system will make that system more stable than the original one. For the case of nonlinear model, the optimized PID controller has proven it can attenuate the disturbances signal that appeared to its system.

As referred to Figures 5 and 6, the best controller gain that minimizes the ISE ($\overline{E}$) for the actual simulation is given in Table 7. Meanwhile, the time taken to complete the actual process for each model is given in Table 8.

Next, the comparison of the minimum errors between the actual process and Metamodel is done and implied in Table 9. Although the results are slightly differed between each others, it can be observed that the Metamodel managed to approximate the global minimum of the error curve fairly well.

7. Conclusion

In this study, the initial data set, D was created based on the background knowledge of the problem. The data contained in the initial data set, D need to be determined carefully, as the data that mismatch much points in the large space, D' will not run the Metamodeling process efficiently. Hence, there is a need of improvement for Metamodeling techniques. An investigation on the possible use of the Worst Case Approach [21] and the Cross Validation technique [22] in data sampling would seems be the appropriate next step forward. It can be simplified that 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.

References

[1] M. Mohamed Sultan, A. Shahrum Shah, and C. Osman David, "Controllers optimization for a fluid mixing system using metamodeling approach," *International Journal of Simulation Modelling*, vol. 8, no. 1, pp. 48–59, 2009.

[2] K. Rashid, *Optimization in electromagnetics using computational intelligence*, Ph.D. thesis, Department of Electrical and Electronics Engineering, Imperial College of Science, Technology and Medicine, London, UK, 2000.

[3] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[4] J. T. Tou and R. C. Gonzalez, *Pattern Recognition*, Addison-Wesley, Reading, Mass, USA, 1974.

[5] S. Matej and R. M. Lewitt, "Practical considerations for 3-D image reconstruction using spherically symmetric volume elements," *IEEE Transactions on Medical Imaging*, vol. 15, no. 1, pp. 68–78, 1996.

[6] M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D*, vol. 35, no. 3, pp. 335–356, 1989.

[7] R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[8] A. G. Bors and M. Gabbouj, "Minimal topology for a radial basis functions neural network for pattern classification," *Digital Signal Processing*, vol. 4, no. 3, pp. 173–188, 1994.

[9] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[10] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Upper Saddle River, NJ, USA, 1994.

[11] M. Niranjan and F. Fallside, "Neural networks and radial basis functions in classifying static speech patterns," *Computer Speech and Language*, vol. 4, no. 3, pp. 275–289, 1990.

[12] I. Cha and S. A. Kassam, "RBFN restoration of nonlinearly degraded images," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 964–975, 1996.

[13] G.-Q. Wei and G. Hirzinger, "Parametric shape-from-shading by radial basis functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 353–365, 1997.

[14] A. G. Bors and I. Pitas, "Object classification in 3-D images using alpha-trimmed mean radial basis function network," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1744–1756, 1999.

[15] A. G. Bors and I. Pitas, "Optical flow estimation and moving object segmentation based on median radial basis function network," *IEEE Transactions on Image Processing*, vol. 7, no. 5, pp. 693–702, 1998.

[16] V. Chatzis, A. G. Bors, and I. Pitas, "Multimodal decision-level fusion for person authentication," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 29, no. 6, pp. 674–680, 1999.

[17] F. Girosi and T. Poggio, "Networks and the best approximation property," *Biological Cybernetics*, vol. 63, no. 3, pp. 169–176, 1990.

[18] A. A. Mullur and A. Messac, "Extended radial basis functions: more flexible and effective metamodeling," *AIAA Journal*, vol. 43, no. 6, pp. 1306–1315, 2005.

[19] N. Dyn, D. Levin, and S. Rippa, "Numerical procedures for surface fitting of scattered data by radial basis functions," *SIAM Journal of Scientific and Statistical Computing*, vol. 7, no. 2, pp. 639–659, 1986.

[20] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Prentice-Hall, Upper Saddle River, NJ, USA, 11th edition, 1998.

[21] S. S. Abdullah, *Experiment design for deterministic model reduction (metamodeling) and neural network training*, Ph.D. thesis, Department of Electrical and Electronics Engineering, Imperial College of Science, Technology and Medicine, London, UK, 2003.

[22] R. Jin, W. Chen, and A. Sudjianto, “On sequential sampling for global metamodeling in engineering design,” in *Proceedings of the ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference*, vol. 2, pp. 539–548, Montreal, Canada, 2002.