

Application of Neural Net Modeling and Inverse Control to the Desulphurization of Hot Metal Process

“Moh’d Sami” S. Ashhab *

Department of Mechanical Engineering, The Hashemite University, Zarqa, 13115, Jordan

Abstract

Optimization techniques are powerful in determining inputs to a process that will drive its output to a desired target. This inverse control problem reduces to minimizing a positive cost function that measures the difference between the output and its target value. In this paper, we present a method for inverse control that uses a combination of artificial neural net modeling and optimization. We apply this method to the desulphurization of hot metal process. In this steel industry application, the sulphur content of hot metal, obtained at the end of calcium carbide powder injection into 400 ton torpedo ladles is predicted as a function of hot metal weight, treatment time, initial sulphur content, gas flow rate, and powder injection rate. Based on the prediction model, the optimization algorithm coordinates the five inputs or part of them to achieve a desired sulphur content in the hot metal.

© 2007 Jordan Journal of Mechanical and Industrial Engineering. All rights reserved

Keywords: Inverse control; Neural network; Constrained optimization; Desulphurization;

Nomenclature

Alphabetic symbols

B_i	: column vector of size n that contains the biases from the input to the hidden layer of the neural net model
b_o	: bias (scalar) from the hidden layer to the output of the neural net model
d	: desired output value
J	: cost function
Lb	: lower input bound
M	: number of points of the simplex in the complex method algorithm
p	: number of process inputs
p_c	: number of process inputs that can be changed by the controller
U	: column vector of size p that contains the p inputs of the process.
Ub	: upper input bound
U^o	: initial input vector for the complex method algorithm
W_o	: row vector of size n that contains the weights of the neural net model from the hidden layer to the output
W_i	: matrix that contains the weights of the neural net model from the inputs to the hidden layer and has n rows and p columns

y : neural net model output

Greek symbols

α	: reflection parameter for the complex method algorithm
ε	: termination parameter for the complex method algorithm
δ	: termination parameter for the complex method algorithm

1. Introduction

In many industrial applications, there is a need to calculate the inputs to a process that will drive its output in a desired way and thus achieve some optimum (desired) goal. In such applications a mathematical input-output model for the process is usually derived. The model could be based on the physical phenomena or available (historical) data about the process. Once the model is developed, control techniques can be used to determine the inputs to the process that will satisfy a certain given criteria.

In this paper, we deal with the inverse control problem of a multi-input-single-output process whose input/output data is given. The objective of the controller is to determine the set of inputs that will make the output of the process go to its desired target value. The input/output data is used to train an artificial neural net model with one hidden layer. This kind of identified model is an input-

* Corresponding author. e-mail: sami@hu.edu.jo

output model for which control techniques can be applied. To achieve our objective we build a positive cost function that measures the difference between the output and its target value. We use the complex method optimization to minimize this cost function. The resultant set of inputs will drive the output as close as possible to its target value. Clearly, the output will achieve its target value if and only if the minimum of the cost function is equal to zero.

Control of linear systems is well developed and neural network modeling is not needed in this case even though the procedure will give good results. A method to identify the inverse of linear systems was presented in [1]. For non-linear systems, the use of neural net modeling is helpful, especially, for the cases where a mathematical model based on the physical phenomena is difficult to obtain and process input/output data is available. The inverse model can be learned by an artificial neural network (ANN) [2]. The process inputs are predicted if the target outputs are passed to the net as inputs. There has been a lot of research on using ANN for control [3-5]. In this paper we calculate the inputs to a process based on the forward (input to output) neural net model. After proper training, the net can be used to calculate particular inputs when all the other inputs and the target output are known. This is done using optimization techniques. Thus, the ANN model functions as a versatile means for process control.

The developed inverse controller is applied to the desulphurization of hot metal process. The desulphurization of carbon saturated hot-metal in torpedo ladles is performed by injecting powdered calcium diamide through a submerged lance. Nitrogen acts as a carrier gas. The interaction between the injected gas plus powder and the liquid metal is complex because of a variety of reaction interfaces. Seven different reaction zones can be identified according to the reactor model for the ladle injection. The powder particles, on entering the gas stream, are accelerated by the high velocity of the carrier gas. On leaving the nozzle, the particles travel almost in a straight line downwards until the gas jet breaks up into a swarm of rising gas bubbles as a result of the resistance offered by the metal buoyancy forces. The final sulphur content of hot metal can be considered to be a function of five input variables, namely, treatment time, hot metal weight, initial mass content of sulphur, carrier gas flow rate, and powder injection rate [6].

A neural net model for the desulphurization of hot metal process described above is derived. The model predicts the final sulphur content of hot metal as a function of the five inputs. The error between the real and predicted outputs is small. Based on the developed neural net model of the desulphurization process an inverse controller is derived. The controller objective is to determine the five inputs or part of them that will achieve a predetermined value of the final sulphur content of the hot metal. We call this value the desired or target value of the output. The controller uses given initial inputs and the target value of the output to calculate the corresponding optimum inputs. If the optimization algorithm is successful, these inputs are mapped by the neural net model to the desired output. It should be clear that the optimum set of inputs will cause the true desulphurization process to have approximately the target value of the final sulphur content if the developed neural net model can predict the output accurately over the given range of inputs. Therefore, accurate models of the process are significant in order to obtain good performance of the controller.

2. Neural Net Modeling Experiment Design

Neural networks were originally inspired as being models of human nervous system. They have been shown to exhibit many abilities, such as learning, generalization, and abstraction [7]. Useful information about neural networks can also be found in [8]. These networks are used as models for processes that have input/output data available. The input/output data allows the neural network to be trained in a way that minimizes the error between the real output and the estimated (neural net) output. The model is then used for different purposes among which are estimation and control.

The neural net structure that we will consider in this paper is shown in Figure 1. The inputs are linked to a hidden layer which in turn is linked to the output. The components of the hidden layer are called nodes or neurons. Each link is associated with a weight. These weights are determined from given input/output data about the process such that the least square error between the given output data and the model output is minimized. The optimization process for calculating the weights of the neural net is called training. If the data used for training is huge or the number of hidden nodes is large then the training may take a long time. Once the net is trained it can be tested on a different set of data than that used for training. It is a good approach to divide the given input/output data into two parts: one part is used for training whereas the other usually smaller part is used for testing the neural net model.

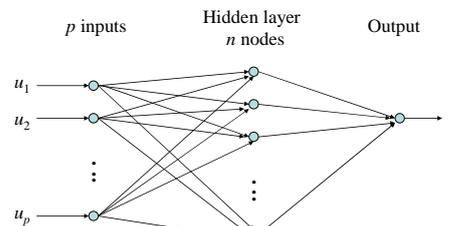


Figure 1: Neural net structure

Note that Figure 1 shows a neural net with one hidden layer. This is the case that we are going to consider since a number of results have been published showing that a feed-forward network with only a single hidden layer can well approximate a continuous function [9,10]. In practice, most processes are continuous.

An artificial neural net mathematical model that represents the structure shown in Figure 1 is written as

$$y = f(U) = W_o * \tanh(W_i * U + B_i) + b_o \quad (1)$$

where,

- y : is the output of the neural net model.
- U : is a column vector of size p that contains the p inputs of the process. If there is only one input the U becomes a scalar.
- W_o : is a row vector of size n that contains the weights of the neural net model from the hidden layer to the output.
- W_i : is a matrix that contains the weights of the neural net model from the inputs to the

hidden layer. This matrix has n rows and p columns.

B_i : (not shown in Figure 1) is a column vector of size n that contains the biases from the input to the hidden layer of the neural net model.

b_o : (not shown in Figure 1) is the bias (scalar) from the hidden layer to the output of the neural net model.

$\tanh(W_i*U+B_i)$: is a column vector that contains the hyperbolic tangent of the elements of the vector W_i*U+B_i

Each input $u_j, j=1,2,\dots,p$ has lower and upper bounds, Lb_j and Ub_j , respectively. These bounds are calculated from the given input data. Lb_j is the minimum value of the j th input over the given data whereas Ub_j is the maximum value of the j th input over the given data. If all the inputs lie within their lower and upper bounds then the estimated output by the neural net should lie within the given output data range. In other words, the output bounds are dictated by the neural network and the input bounds.

The number of hidden neurons affects the performance of the neural net over the training and test sets of data. More neurons make the fitting of data more accurate over the training region. It is more important to check the performance of the model over the test set of data since it was not used to calculate the parameters of the model. Overtrained neural net may perform badly over the test set. The number of nodes is usually chosen by trying different values and selecting the one that performs best over both the training and test sets of data.

3. Inverse Control

The inverse control of a process is to find the inputs to that process that will drive its output as close as possible to a given desired goal. In order to solve this problem a mathematical model of the process is needed. In this paper we deal with processes whose input/output data is available. We model the process using artificial neural networks based on the available input/output data. Optimization techniques are then used to determine the inputs to the process that will drive its output as close as possible to the given desired goal. Figure 2 illustrates these ideas. A desired signal d is passed to the inverse controller. The controller determines the input signal U to the process such that the output y of the neural net model is as close as possible to the desired signal d . The real output y_r of the process will not be equal to y unless there are no modeling errors. In practice, the performance of the controller will depend on the accuracy of the controller and the neural net model.

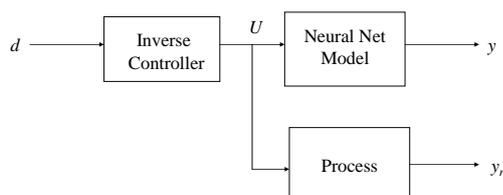


Figure 2: Inverse controller

The controller operation is characterized by minimizing a positive cost function that measures the difference between the model output and its desired value. If there is a set of inputs at which the cost function is equal to zero (minimum) then this is an optimum input. Generally speaking, the output is not equal to its target value at the initial inputs. Thus, the inverse controller coordinates the inputs or part of them to achieve the desired output if possible. Optimization over a subset of the inputs is a feature of the developed algorithm that leaves the freedom to the user to determine the other inputs in a way such that other essential requirements of the process are met. In this paper, we test our optimization algorithm on the developed neural net model of the desulfurization process (see next section).

Given a neural net model of a process as described in the previous section, we define the cost function J as:

$$J = (d - y)^2 \quad (2)$$

The cost function is expressed explicitly in terms of the model output. However, since the output is a function of the process inputs as given in the equation of the neural net model which is available in the previous section, the cost function is expressed implicitly in terms of the inputs. Note that the cost function J is non-negative and is equal to zero if and only if the output is at its desired value. In order to drive the output as close as possible to its target value we need to minimize J . Optimization techniques are best in tackling such problems.

We are now ready to define our inverse control problem: Given a process with p inputs and a single output, and a neural net model for this process. Assume that p_c of the p inputs (p_c is less than or equal to p) are allowed to change by the controller. These are called the control inputs. Minimize the cost function J subject to the following constraints

$$Lb_i \leq u_i \leq Ub_i, i = 1, 2, \dots, p \quad (3)$$

We are going to use the complex method to solve this constrained optimization problem. This technique will be applied to the control inputs only. The other inputs will stay constant at their given initial values. These inputs are usually set at values (within their given constraints) that satisfy some other desired conditions for the process. The complex method is based on the generation and maintenance of a pattern of search points and the use of projections of undesirable points through the centroid of the remaining points as the means of finding new trial points. In the presence of inequality constraints, it is evident that if the new point is infeasible, then it is a simple matter to retract it toward the centroid until it becomes feasible. Box proposed in [11] the complex method as a better alternative to the simplex direct search-method. The considerations that led him to the simplex modification are:

1. The destruction of the regularity of the simplex.
2. Many of the points of the regular initial simplex will be infeasible.

3. The points must be generated sequentially rather than being essentially simultaneously defined using the formula for a regular simplex.

Box proposed in [11] that the set of P trial points be generated randomly and sequentially. N samples are required to define a point in N dimensions. Each newly generated point is tested for feasibility, and if infeasible it is retracted toward the centroid of the previously generated points until it becomes feasible. The total number to be used, P , should be no less than $N+1$ but can be larger. The complex method can be summarized as follows [12]:

Given an initial strictly feasible point U^o (set of inputs), reflection parameter α , and termination parameters ε and δ .

1. Generate the initial set of M feasible points. For each point $m = 1, \dots, M-1$,
 - (a) Sample p_c times to determine the point U^m . Note that the inputs that are not allowed to change are left constant at their initial values.
 - (b) If U^m is infeasible, calculate the centroid U_a of the current set of points and reset

$$U^m = U^m + \frac{1}{2}(U_a - U^p)$$
 Repeat until U^m becomes feasible.
 - (c) If U^m is feasible, continue with (a) until M points are available.
 - (d) Evaluate $J(U^m)$, for $m = 0, 1, \dots, M-1$.
2. Carry out the reflection step.
 - (a) Select the point U^R such that

$$J(U^R) = \max J(U^m) \equiv J_{\max}$$
 - (b) Calculate the centroid U_a and the new point

$$U^n = U_a + \alpha(U_a - U^R).$$
 - (c) if U^n is feasible and $J(U^n)$ is greater than or equal to J_{\max} , retract half the distance to the centroid U_a . Continue until $J(U^n)$ becomes less than J_{\max} .
 - (d) If U^n is feasible and $J(U^n) < J_{\max}$, go to step 4.
 - (e) If U^n is infeasible, go to step 3.
3. Adjust for feasibility
 - (a) Reset violated variable bounds:

$$\text{If } u_i^n < Lb_i, \text{ set } u_i = Lb_i.$$

$$\text{If } u_i^n > Ub_i, \text{ set } u_i = Ub_i.$$
 Note that the subscript i denotes the i th input. This step will be applied only to the control inputs since the other inputs are set within their constraints initially.
 - (b) If the resulting U^n is infeasible, retract half the distance to the centroid. Continue until U^n is feasible, then go to step 2(c).
4. Check for termination.
 - (a) Calculate

$$J_a = \frac{1}{M} \sum J(U^m)$$
 and

$$U_a = \frac{1}{M} \sum U^m.$$
 - (b) If

$$\sum_m (J(U^m) - J_a)^2 \leq \varepsilon$$
 and

$$\sum_m \|U^m - U_a\| \leq \delta$$
 Terminate. Otherwise, go to step 2(a).

The search is terminated when the pattern of points has shrunk so that the points are sufficiently close together and/or when the differences between the function values at the points become small enough.

Box has performed numerical experiments in [11] with this algorithm, and on this empirical basis recommends using $\alpha = 1.3$ and $M = 2p_c$. Biles [13], on the other hand, reports good results with $M = p_c + 2$. The $\alpha > 1$ compensates for shrinking of the complex caused by halving the distances, while the large number of vertices is intended to prevent the complex from collapsing and flattening out along the constraints. The setting at the value of the bound is intended to avoid unnecessary contraction of the complex.

The complex method requires that the feasible region be a convex set. This requirement arises in two places: in the calculation of the centroid and in the retraction of a feasible but unsatisfactory point. For nonconvex regions the method could fail to converge satisfactorily. In practice, the method is widely used and has successfully solved numerous nonconvex problems. Thus, the above situation must arise with low probability.

4. The Desulphurization of Hot Metal Process

The primary objective of introducing powdered reagents into liquid steel, with the help of either cored wire or a carrier gas, is for desulphurization, deoxidation, alloy dissolution and inclusion shape control. Injection technology and equipment are well advanced and available on commercial basis [6].

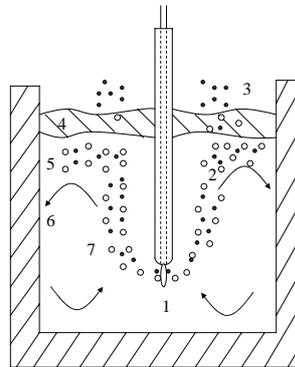


Figure 3: Reactor model for ladle injection. The different reaction zones are: (1) jet zone in front of lance outlet; (2) plume zone containing bubbles and particles rising in metal; (3) breakthrough zone where bubbles emerge; (4) slag zone; (5) dispersion zone where gas-slag-metal emulsion can form; (6) lining zone where metal may react with lining; and (7) intermediate zone with lowest stirring intensity [6]

The desulphurization of carbon saturated hot-metal in torpedo ladles is performed by injecting powdered calcium diamide through a submerged lance (see Figure 3). Nitrogen acts as a carrier gas. The interaction between the injected gas plus powder and the liquid metal is complex because of a variety of reaction interfaces. Seven different reaction zones can be identified according to the reactor model for the ladle injection. The powder particles, on entering the gas stream, are accelerated by the high velocity of the carrier gas. On leaving the nozzle, the

particles travel almost in a straight line downwards until the gas jet breaks up into a swarm of rising gas bubbles as a result of the resistance offered by the metal buoyancy forces. The final sulphur content of hot metal can be considered to be a function of five input variables, namely, treatment time, hot metal weight, initial mass content of sulphur, carrier gas flow rate, and powder injection rate.

The desulphurization process is very complex and is affected by many parameters as mentioned above. It is almost impossible to derive a mathematical model for such a chemical process based on physical phenomena. The input/output data for the system is available which allows modeling with neural networks. The data for the desulphurization of hot metal process is provided in [14] and has 51 patterns. Each pattern includes the data about the five inputs and the corresponding output which is the final sulphur content, namely, each pattern includes data for the five inputs:

1. Treatment time
2. Hot metal weight
3. Initial mass content of sulphur
4. Carrier gas flow rate
5. Powder injection rate
6. and the single output which is the final sulphur content.

Forty of the provided 51 data patterns are used to train a neural net model for this process, whereas the remaining 11 data patterns are used to test the performance of the model. We ran experiments for different numbers of hidden neurons. It was observed that the quality of the results depends on this number. The least square error versus the number of hidden neurons is plotted for the training and test regions in Figure 4. We choose the ANN with 25 hidden neurons since it produces the minimum least square error over the test region and a very small least square error over the training region. The real and predicted (ANN model) sulphur content are plotted in Figure 5 as a function of the pattern index which starts from 1 and ends at 51. Note that in the training region (index < 41) the predicted values are very close to the real ones. The corresponding least square error is 0.0032 and standard deviation is 0.0005. On the other hand, there are some errors at indices 43 and 48 within the test region. The test least square error and standard deviation are 0.0110 and 0.0034, respectively. The results are acceptable compared to the ones generated by neural models in the literature. In [14] the authors used the same training patterns to obtain a neural net model that predicts the final sulfur content. Their generalization results over the same test set we have above are shown in Figure 6. The corresponding least square error and standard deviation are 0.0116 and 0.0037, respectively. Note that our results over the test region are slightly better which shows that our ANN model is satisfactory and thus will be utilized by the proposed inverse controller later in this section.

The neural network that represents the mathematical model for the desulphurization process consists of 25 hidden neurons and there are five inputs. In this case, there are more parameters than data points. Over fitting is expected under these conditions. The available data for the desulphurization process consists of 51 patterns. Part of this data must be used for training the neural net and the rest of data is used for testing the performance of the neural network.

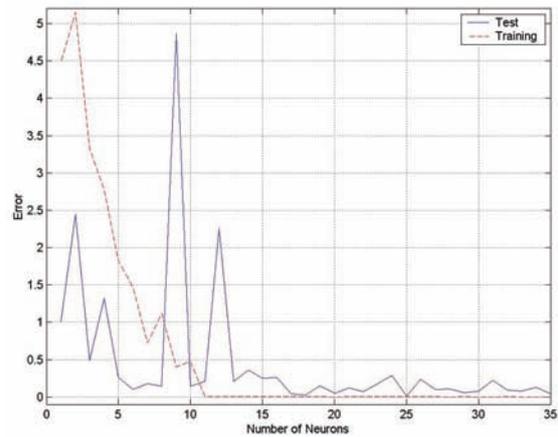


Figure 4: The least square error versus the number of hidden neurons for the training and test regions

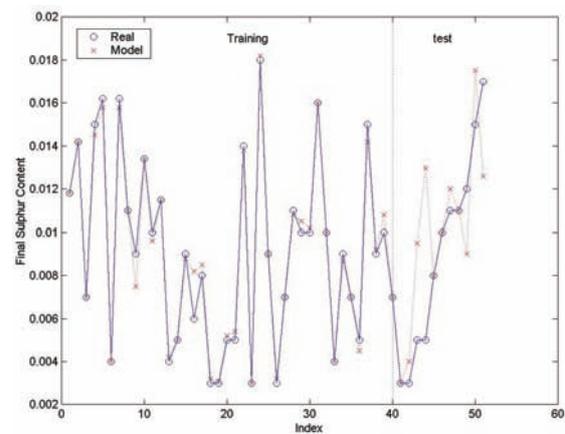


Figure 5: Neural net model results

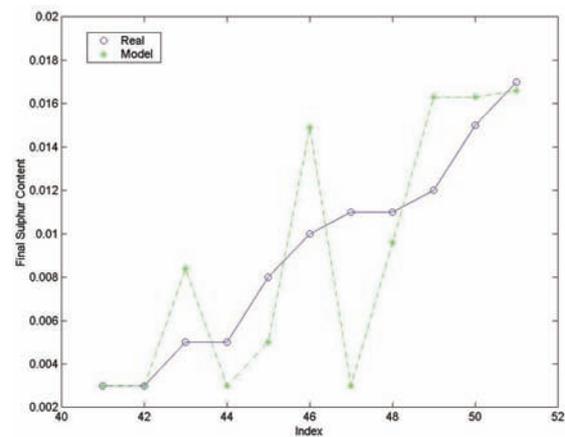


Figure 6: Neural net model results of [14]

We use about 80% of the patterns for training and 20% of the patterns for testing the neural network. Over fitting is avoided when neural nets with small number of neurons are used. However, it was observed that the performance over the test region is bad in these cases. Increasing the number of neurons causes over fitting in the training region. Therefore, the quality of the neural net is judged by its performance over the test region. In the case of the desulphurization process the best performance over the test region was obtained with 25 hidden neurons. It is worth to point out that in order to construct better models for the

desulphurization process more data points should be generated.

The optimization algorithm described earlier in this paper uses given initial inputs to the desulphurization of hot metal process and a target (desired) value of the final sulphur content in order to calculate the optimum set of inputs that will cause the actual final sulphur content to be equal to its desired value. As discussed above, the algorithm represented in this paper is based on the neural net model of the process. Optimization over a subset of the inputs is an important feature of the algorithm. This allows choosing the other inputs such that some essential requirements can be satisfied. We now present the simulation results that demonstrate the optimization algorithm ability in driving the final output sulphur content from a given initial value to a final desired one by coordinating the inputs to the desulphurization process. We pick one of the available data patterns to be the initial state. The inputs thus have the following initial values:

1. Treatment time = 430 seconds
2. Hot metal weight = 267 tons
3. Initial sulphur content = 0.021%
4. Gas flow rate = $0.677 \text{ m}^3 / \text{min}$.
5. Powder injection rate = $50 \text{ kg} / \text{min}$.

The corresponding initial value of the sulphur content is 0.015% as calculated from the neural net model. We would like to reduce the sulphur content to a final value of 0.0117 by regulating only two out of the above five inputs, namely, treatment time and gas flow rate. The remaining three inputs will be fixed at their initial values. The simulation results

of the optimization algorithm in this case are shown in Figure 7. The three subplots show history of the two inputs and output in terms of the iteration number of the complex method. The inverse controller achieves its goal of moving the final sulphur content to 0.0117 in ten iterations. The corresponding optimum values of the inputs are treatment time = 699 seconds and gas flow rate = $0.6 \text{ m}^3 / \text{min}$. Iteration 0 represents the initial condition for the optimization algorithm or inverse controller.

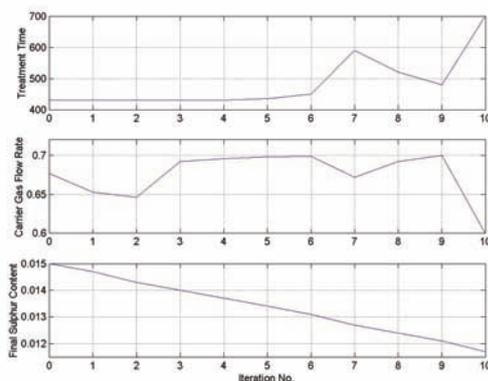


Figure 7: Inverse controller performance

5. Conclusions

A method for inverse control of processes with available input/output data was developed. The method consists of neural net modeling and constrained optimization using the complex method. The developed inverse controller was tested on the desulphurization of hot metal process. The controller achieves its objective.

References

- [1] B. Widrow, J. McCool, B. Medoff, "Adaptive control by inverse modeling". Twelfth Asilomar Conference on Circuits, Systems, and Computers, 1979.
- [2] A. Karniel, R. Meir, G. Inbar, "Best estimated inverse versus inverse of the best estimator". *Neural Networks*, Vol. 14, 2001, 1153-1159.
- [3] K. Narendra, K. Parthasarathy, "Identification and control of dynamical systems using neural networks". *IEEE Transaction On Neural Networks*, Vol. 1, No. 1, 1990, 4-27.
- [4] D. Bullock, S. Grossberg, F. Guenther, "A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm". *Journal of Cognitive Neuroscience*, Vol. 5, No. 4, 1993, 408-435.
- [5] A. Karniel, G. Inbar, "Human motor control: Learning to control a time-varying non-linear many-to-one System". *IEEE Transactions on Systems, Man, and Cybernetics Part C*, Vol. 30, No. 1, 2000, 1-11.
- [6] Deo B, Boom R. *Fundamentals Of Steelmaking Metallurgy*. New York: Prentice Hall, 1993.
- [7] Patterson D. *Introduction To Artificial Intelligence And Expert Systems*. New York: Prentice Hall, 1990.
- [8] Haykin S. *Neural Networks: A Comprehensive Foundation*. New York: Prentice Hall, 1998.
- [9] G. Cybenko, "Approximations by superpositions of a sigmoidal function". *Math. Control Signal Systems*, Vol. 2, 1989, 303-314.
- [10] K. Funahashi, "On the approximate realization of continuous mappings by neural networks". *Neural Networks*, Vol. 2, 1989, 183-192.
- [11] M. Box, "A new method of constrained optimization and a comparison with other methods". *Computer Journal*, Vol. 8, 1965, 42-52.
- [12] Reklaitis G. *Engineering Optimization*. New York :John Wiley & Sons; 2002.
- [13] W. Biles, "Optimization of multiple response simulation models". Final Report, ONR-Contract N00014-76-C-1021, University of Notre Dame, 1978.
- [14] A. Datta, M. Hareesh, P. Kalra, B. Deo, R. Boom, "Adaptive neural net (ANN) models for desulphurization of hot metal and steel". *Steel Research*, Vol. 65, No. 11, 1994, 466-471.