

MODELLING & SIMULATION OF HUMAN POWERED FLYWHEEL MOTOR FOR FIELD DATA IN THE COURSE OF ARTIFICIAL NEURAL NETWORK – A STEP FORWARD IN THE DEVELOPMENT OF ARTIFICIAL INTELLIGENCE

A. R. Lende¹, J. P. Modak²

¹Ex. Assistant Professor, Mechanical Engineering, DMIETR, Wardha, Maharashtra, India,
²Emeritus Professor & Dean (R&D), Mechanical Engineering, PCE, Nagpur, Maharashtra, India
 tanuja.chandak04@gmail.com, jpmodak@gmail.com

Abstract

As per geographical survey of India about 65% of human population is living in rural areas where urban resources like electricity, employment accessibility, etc are very deprived. The country is still combating with fundamental needs of every individual. The country with immense population living in villages ought to have research in the areas which focuses and utilizes the available human power. Some Authors of this paper had already developed a pedal operated human powered flywheel motor (HPFM) as an energy source for process units. The various process units tried so far are mostly rural based such as brick making machine (both rectangular and keyed cross sectioned), Low head water lifting, Wood turning, Wood strips cutting, electricity generation etc. This machine system comprises three sub systems namely (i) HPFM (ii) Torsionally Flexible Clutch (TFC) (iii) A Process Unit. Because of utilization of human power as a source of energy, the process units have to face energy fluctuation during its supply. To evaporate this rise and fall effect of the energy, the concept of use of HPFM was introduced. During its operation it had been observed that the productivity has great affection toward the rider and producing enormous effect on quality and quantity of the product. This document takes a step ahead towards the development of a controller which will reduce system differences in the productivity. This paper contributes in development of optimal model through artificial neural network which enables to predict experimental results accurately for seen and unseen data. The paper evaluates ANN modeling technique on HPFM by alteration of various training parameters and selection of most excellent value of that parameter. The mathematical model of which then could be utilized in design of a physical controller.

Keywords: Artificial Neural Network, HPFM, MATLAB

1. OVERVIEW OF HPFM OPERATED PROCECSS UNIT

1.1 Working of Human Powered Flywheel Motor Energized Process Unit

This machine system comprises three sub systems namely (i) HPFM [11] (ii) Torsionally Flexible Clutch (TFC) (iii) A Process Unit. The process units tried so far are mostly rural based such as brick making machine [1] [3](both rectangular and keyed cross sectioned), Low head water lifting, Wood turning, Wood strips cutting, electricity generation etc.

The Fig-1 shows the schematic arrangement of pedal operated flywheel motor which comprises of following elements
 R= Rider
 M = mechanism (01-OA-B-02-01)

BSC = Big Sprocket Chain Drive

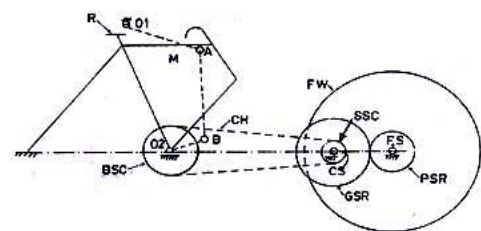


Fig -1: Schematics of Human Powered Flywheel Motor

SSC = Small Sprocket Chain Drive
 GSR = Gear of Speed Rise
 PSR = Pinion of Speed Rise
 FW= Flywheel

CH = Chain
 CS = Counter Shaft
 FS = Flywheel Shaft

1.2 Study of already Available Experimental Data

The various parameters involved [11] in the experimentation are

Table -1: Independent Variables and their symbols

Sr. No.	Independent Variable	Symbol
1	Moment of Inertia of Flywheel	I
2	Input by the Rider	R
3	Time	T
4	Mechanical Efficiency	ME
5	Gear Ratio	G
6	Angular Velocity of Flywheel	ω

Table -2: Range of Variation of Independent parameter i.e. Rider(R)

Range of age	20-25Years
Height	155-170cm
Weight	40-55
Blood Pressure	140-70
Pulse rate	68-80/min.

The observations recorded during the experimentation are as below

Table -3: Experimental observations

Sr. No.	Independent Variables			Dependent variable
	Log(I/RT ²)	Log(ME)	Log(G)	Log(ω T)
1	-7.4270	0.00	0.3010	3.6305
↓	↓	↓	↓	↓
23	-7.1792	0.0662	0.0010	3.5570
↓	↓	↓	↓	↓
50	-7.2694	0.0600	0.0010	3.5004
↓	↓	↓	↓	↓
82	-6.1549	0	0.301	3.0767
↓	↓	↓	↓	↓
141	-5.9717	0	0.301	2.8587
↓	↓	↓	↓	↓
171	-7.2414	0	0.1139	3.4463
↓	↓	↓	↓	↓
200	-7.2694	0	0.0569	3.4107

1.3 Mathematical Model [12]

The experimental Independent variables were reduced by evaluating dimensionless pi terms by Buckingham pi theorem and a mathematical equation was generated by traditional method to predict the experimental findings. The equation is as shown.

$$\omega T = 1.288 (I/RT^2)^{-0.46} (ME)^{-0.87} (G)^{0.40}$$

2. DETECTION OF DILEMMA AND ELECTED APPROACH

The plot has been drawn to observe the prediction of experimental evidences by the traditional empirical model. The figure 2 evaluates and compares the results. Figure 2 undoubtedly express the inaccuracy in prediction of experimental evidences by traditional mathematical model. The existing equation is deficient to predict the desired experimental findings. Consequently we find the need of another modelling technique which can predict the evidences more accurately.

The figure 3 highlights the percentage error in prediction which is of the order of 20% to 80%. In view of the fact that independent variables involved are high in numbers, it becomes a very tedious and inaccurate work to build an accurate mathematical model. ANN i.e. Artificial Neural Network has shown its strength in the field of learning and prediction [6] [7] of the desired results when the input variables are high in numbers. On the other hand ANN model can also give more reliable solution with unseen data.

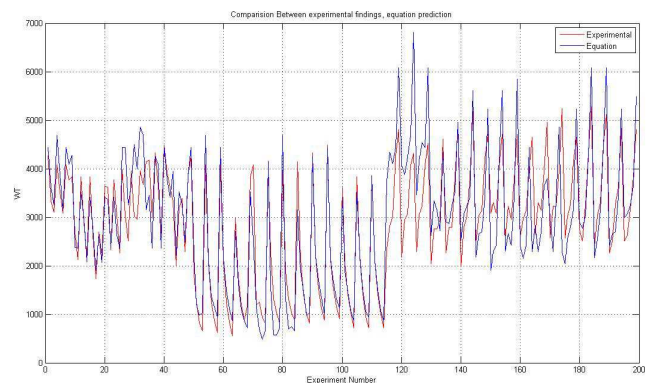


Fig - 2: Prediction of Experimental Findings through Mathematical Model

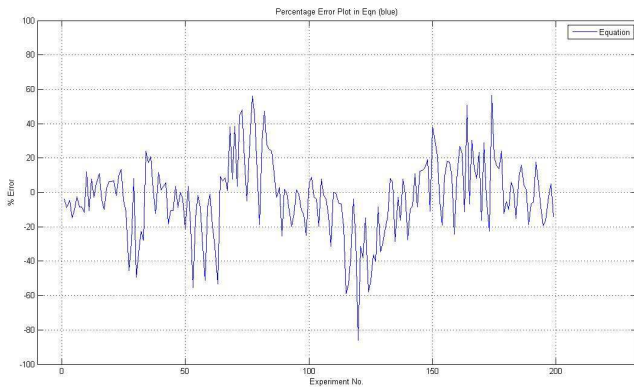


Fig - 3: Percentage Error in Prediction of Experimental Findings through Empirical Model

3. DECIDING THE SEQUENCE OF EXECUTION OF PROPOSED ANN MODELLING PROCEDURE

Modelling a system through ANN simulation [9] involves use of ANN parameters appropriately. A topology is nothing but the complete architecture of network formed through the use of ANN parameters. The ANN parameters should be varied systematically in an attempt to identify best topology for a specified problem. The number of layers was restricted to two as the variables involved were high in number. A table for evaluation of modelling technique is formed [5] as below. The shaded column indicates the variation of that particular parameter and shaded row shows the value of that parameter.

Table -3: Sequence of variation of ANN Parameters

Training Number	Hidden layer Size	Type of Training Function	Performance Function	Types of transfer function		Type of Learning Algorithm
				Layer 1	Layer2	
T1	20	trainlm	mse	tansig	purelin	learngd
T2	50	trainlm	mse	tansig	purelin	learngd
T3	100	trainlm	mse	tansig	purelin	learngd
T4	150	trainlm	mse	tansig	purelin	learngd
T5	250	trainlm	mse	tansig	purelin	learngd
T6	300	trainlm	mse	tansig	purelin	learngd
T7	500	trainlm	mse	tansig	purelin	learngd
T8	600	trainlm	mse	tansig	purelin	learngd
T9	700	trainlm	mse	tansig	purelin	learngd
T10	500	trainb	mse	tansig	purelin	learngd
T11	500	trainbfg	mse	tansig	purelin	learngd
T12	500	trainlm	mse	tansig	purelin	learngd

T13	500	trainbr	mse	tansig	purelin	learngd
T14	500	traingdm	mse	tansig	purelin	learngd
T15	500	traingb	mse	tansig	purelin	learngd
T16	500	traingcf	mse	tansig	purelin	learngd
T17	500	traingcp	mse	tansig	purelin	learngd
T18	500	trainlm	mse	tansig	purelin	learngd
T19	500	trainlm	mae	tansig	purelin	learngd
T20	500	trainlm	sse	tansig	purelin	learngd
T21	500	trainlm	mae	tansig	purelin	learngd
T22	500	trainlm	mae	logsig	purelin	learngd
T23	500	trainlm	mae	tansig	logsig	learngd
T24	500	trainlm	mae	tansig	Purelin	Learncn
T25	500	trainlm	mae	tansig	Purelin	Learngd
T26	500	trainlm	mae	tansig	Purelin	Learnh
T27	500	trainlm	mae	tansig	Purelin	Learnk

4. ILLUSTRATION OF RESULTS

The graphs for each program are generated which illustrate the effect of variation of each parameter on prediction of model. The percentage error in prediction is also plotted to compare and select the best of the topology amongst these topologies.

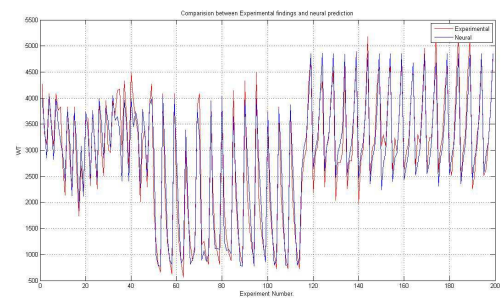


Fig - 4: Neural response with 20 Neurons

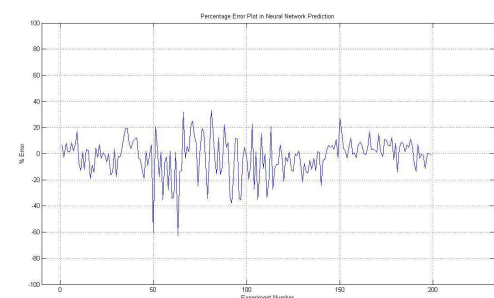


Fig - 5: Percentage error in prediction with 20 Neurons

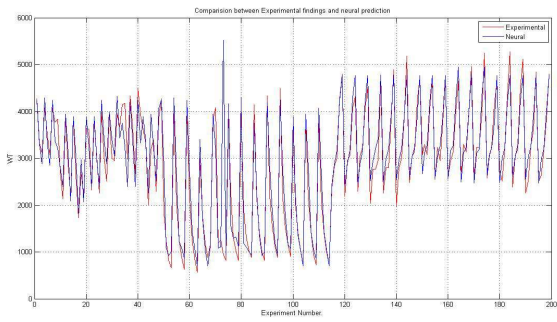


Fig - 6: Neural response with 20 Neurons

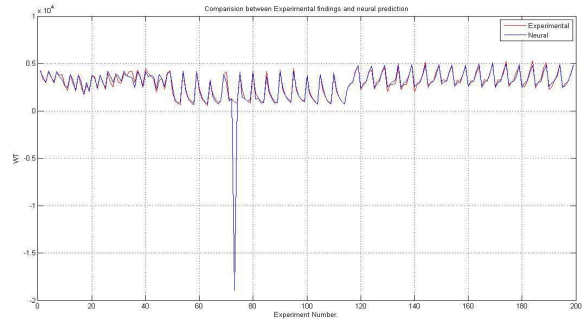


Fig - 10: Neural response with 200 Neurons

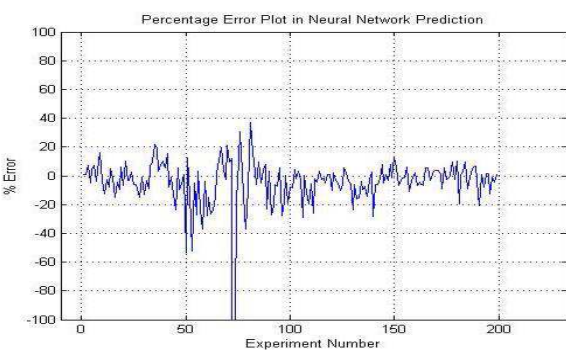


Fig - 7: Percentage error in predication with 50 Neurons

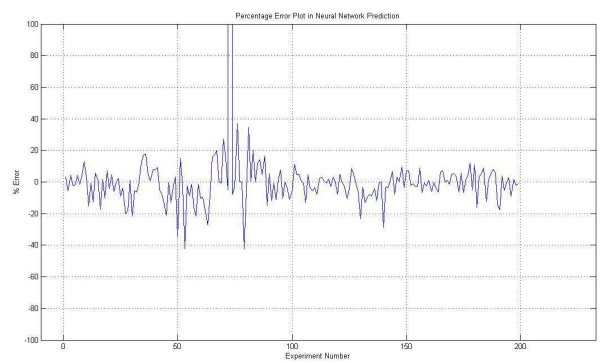


Fig - 11: Percentage error in predication with 200 Neurons

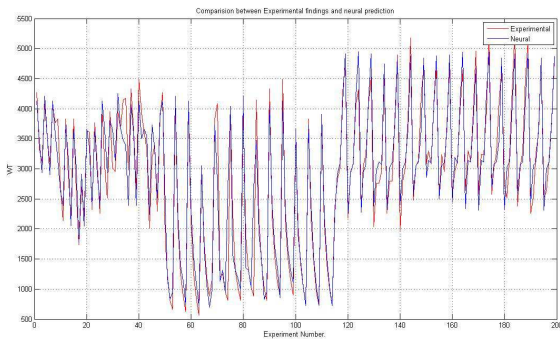


Fig - 8: Neural response with 20 Neurons

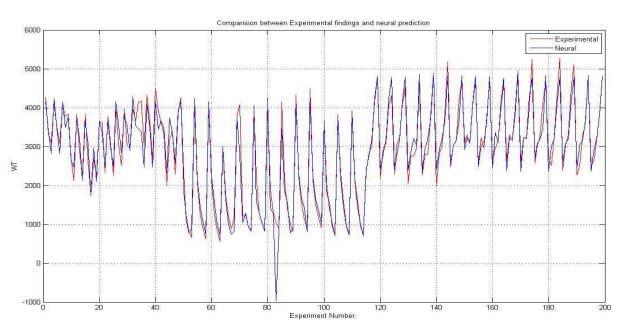


Fig - 12: Neural response with 250 Neurons

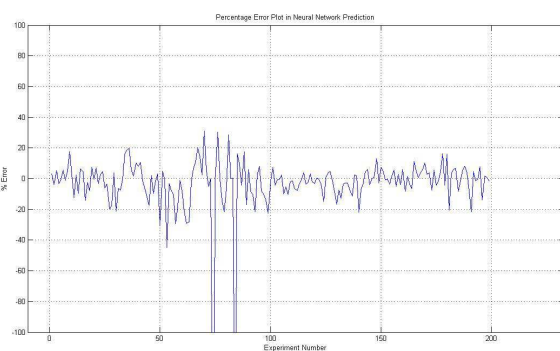


Fig - 9: Percentage error in predication with 150 Neurons

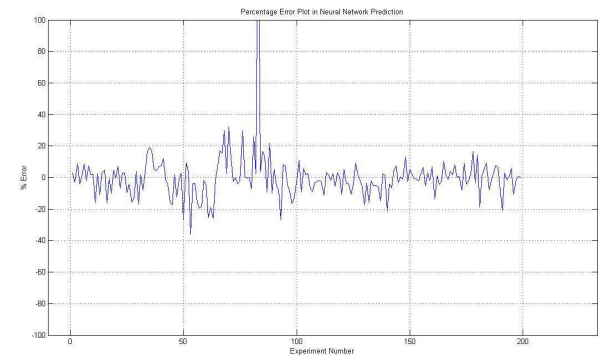


Fig - 13: Percentage error in predication with 250 Neurons

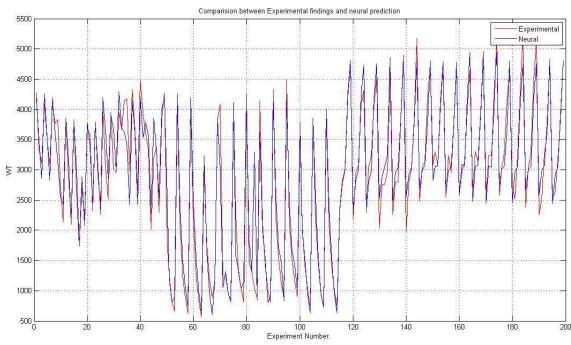


Fig - 14: Neural response with 300 Neurons

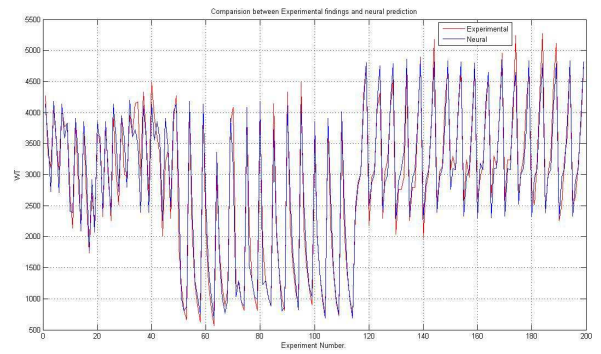


Fig - 18: Neural response with 600 Neurons

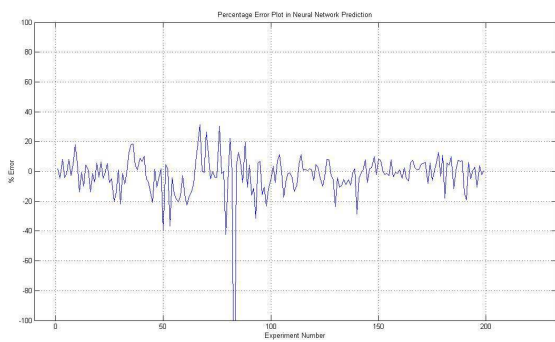


Fig - 15: Percentage error in predication with 300 Neurons

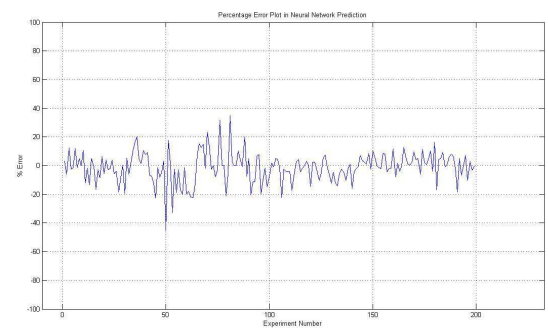


Fig - 19: Percentage error in predication with 600 Neurons

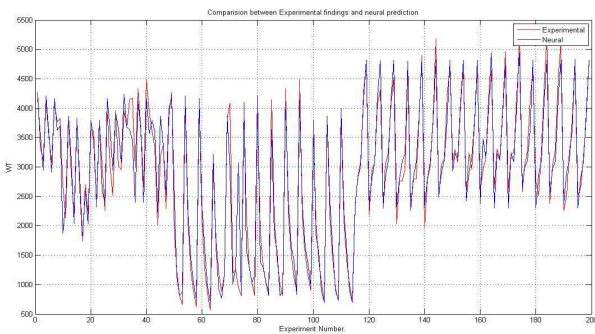


Fig - 16: Neural response with 500 Neurons

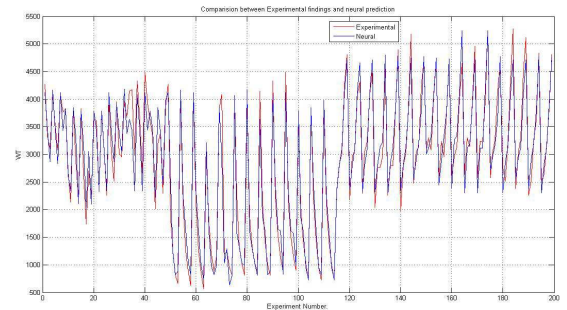


Fig - 20: Neural response with 700 Neurons

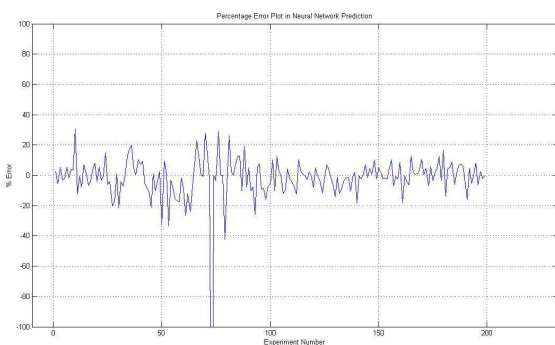


Fig - 17: Percentage error in predication with 500 Neurons

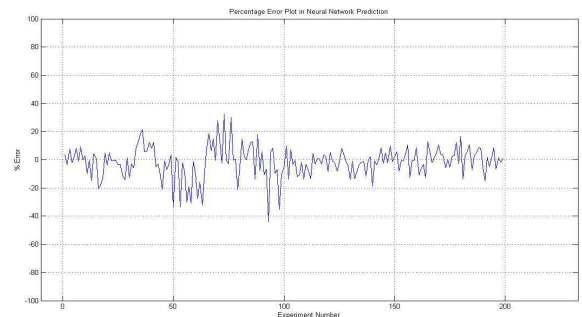


Fig - 21: Percentage error in predication with with 700 Neurons

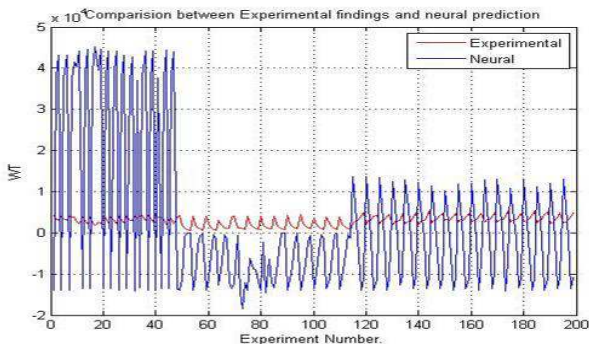


Fig - 22: Neural response with training Function “trainb”

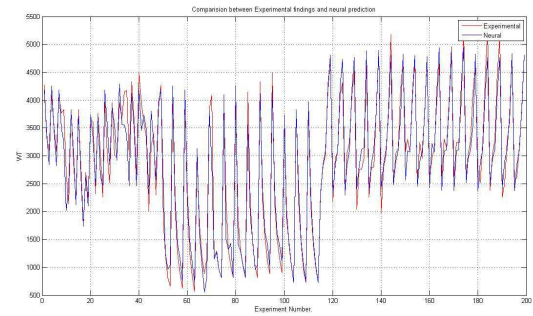


Fig - 26: Neural response with training Function “trainlm”

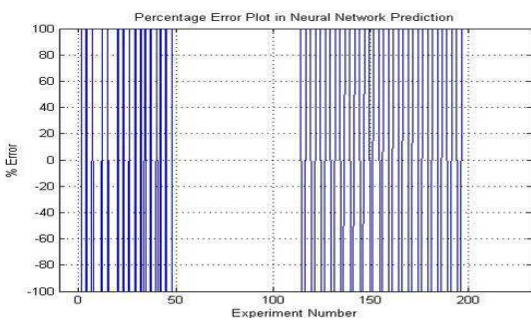


Fig - 23: Percentage error in predication with training Function “trainb”

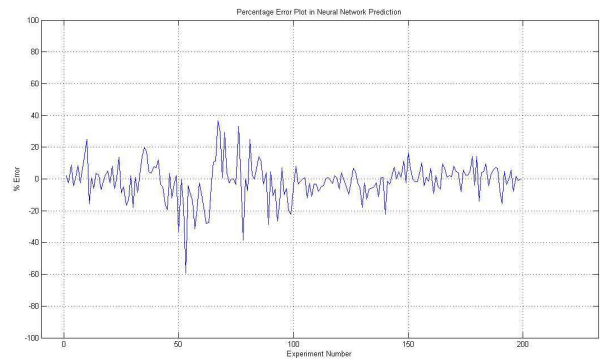


Fig - 27: Percentage error in predication with training Function “trainlm”

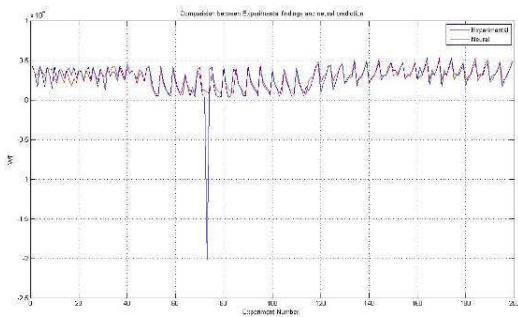


Fig - 24: Neural response with training Function “trainbfg”

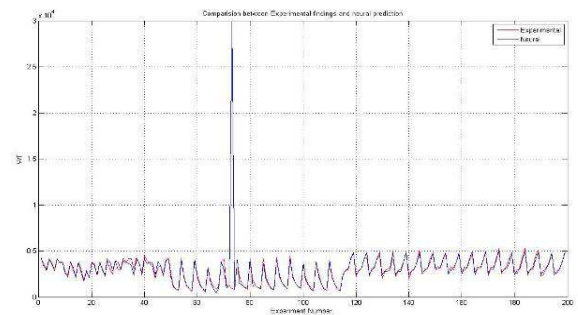


Fig - 28: Neural response with training Function “trainbr”

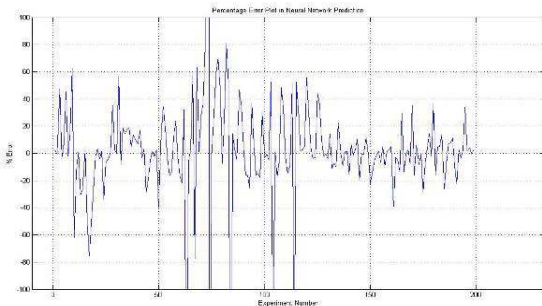


Fig - 25: Percentage error in predication with training Function “trainbfg”

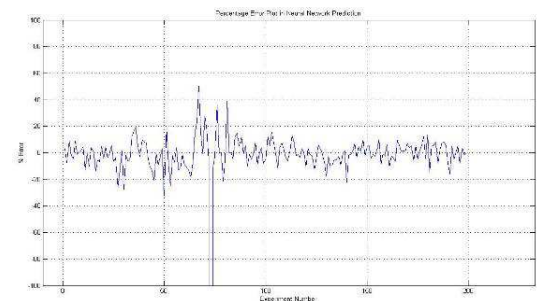


Fig - 29: Percentage error in predication with training Function “trainbr”

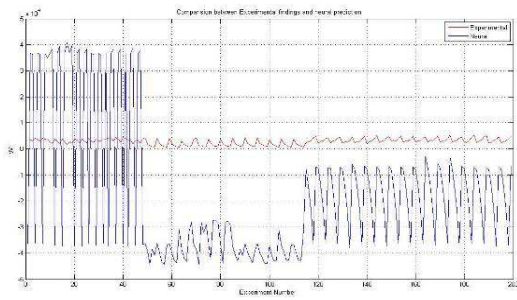


Fig - 30: Neural response with training Function “traingdm”

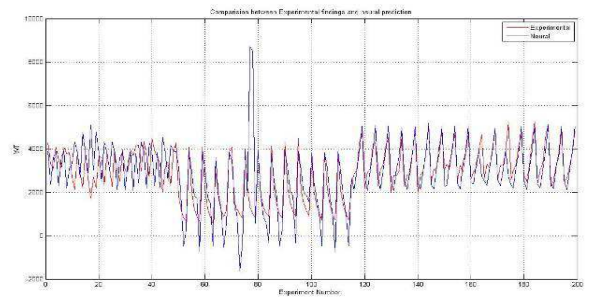


Fig - 34: Neural response with training Function “traincgf”

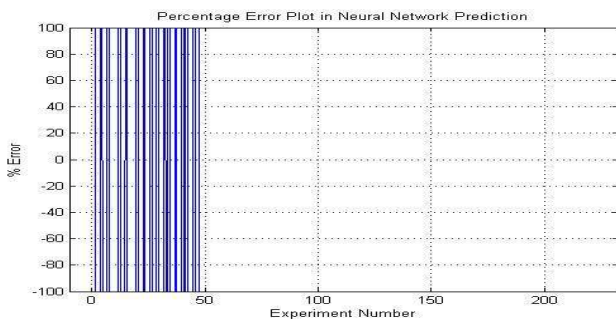


Fig - 31: Percentage error in predication with training Function “traingdm”

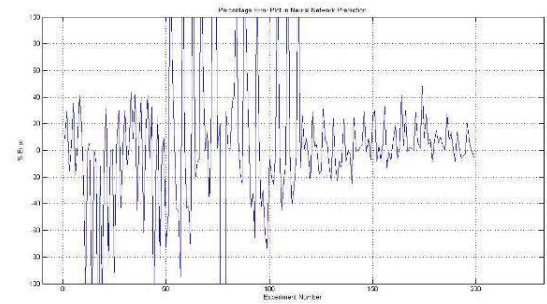


Fig - 35: Percentage error in predication with training Function “traincgf”

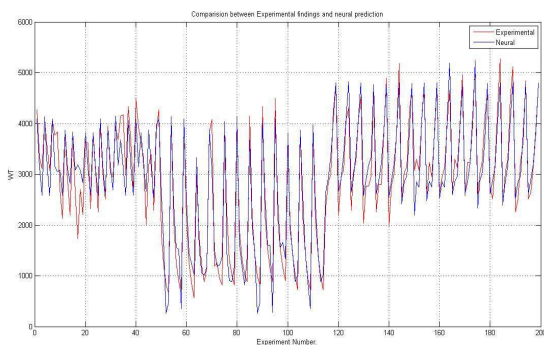


Fig - 32: Neural response with training Function “traincgb”

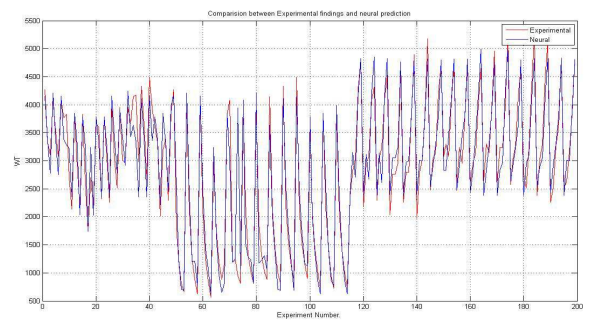


Fig - 36: Neural response with training Function “traincgp”

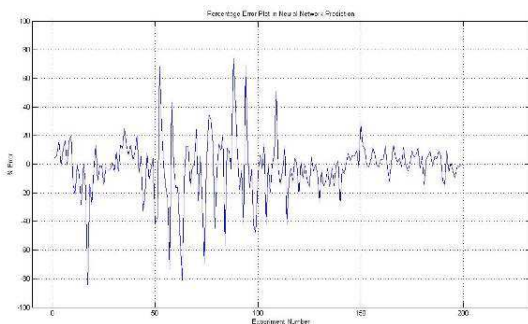


Fig - 33: Percentage error in predication with training Function “traincgb”

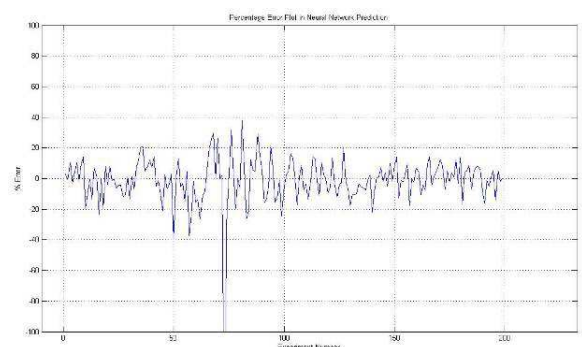


Fig - 37: Percentage error in predication with training Function “traincgp”

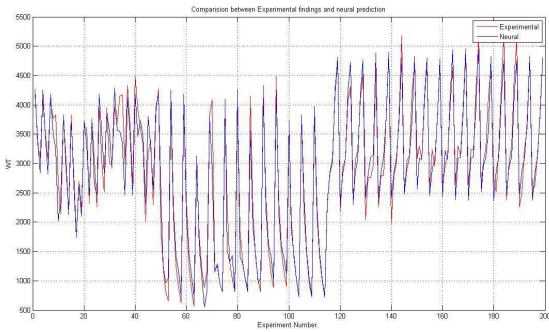


Fig - 38: Neural response with performance Function “mse”

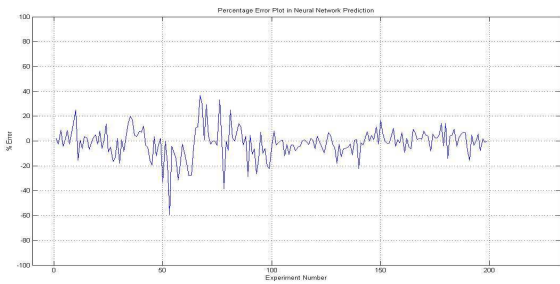


Fig - 39: Percentage error in predication with performance Function “mse”

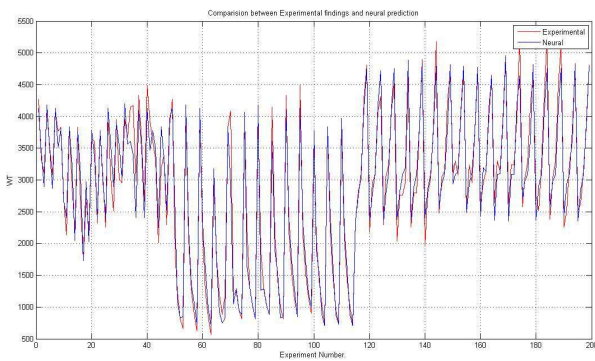


Fig - 40: Neural response with performance Function “mae”

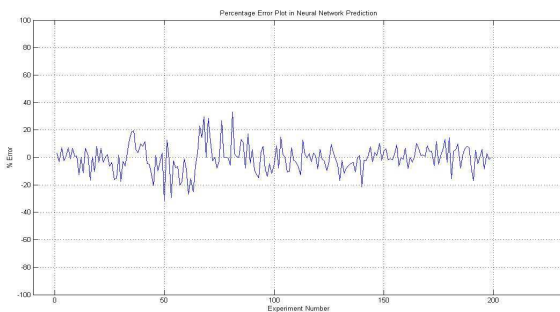


Fig - 41: Percentage error in predication with performance Function “mae”

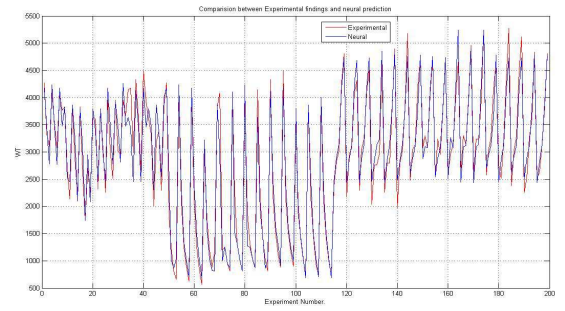


Fig - 42: Neural response with performance Function “sse”

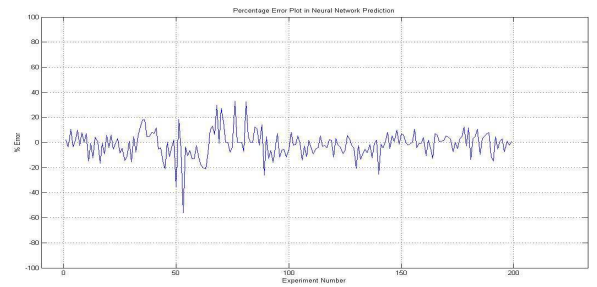


Fig - 43: Percentage error in predication with performance Function “sse”

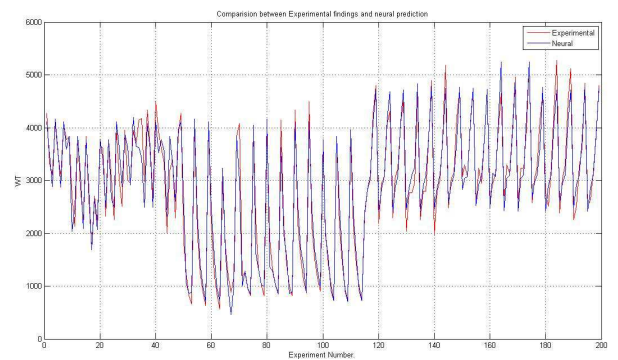


Fig - 44: Neural response with transfer Function “tansig, purelin”

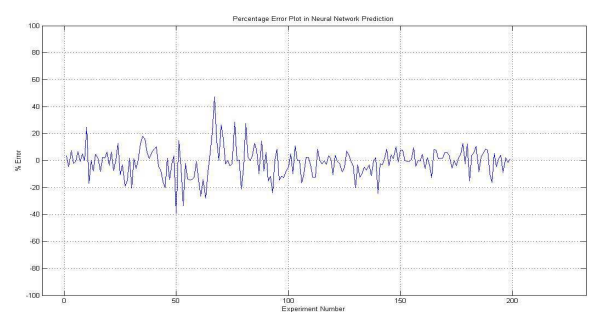


Fig - 45: Percentage error in predication with transfer Function “tansig, purelin”

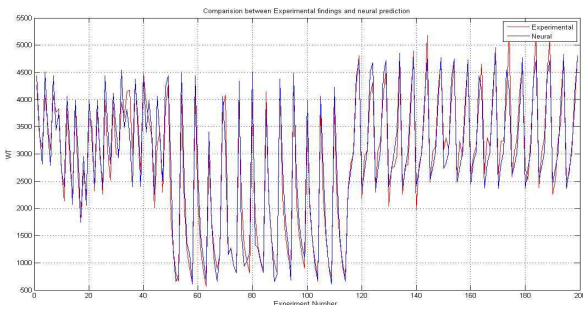


Fig - 46: Neural response with transfer Function “logsig, purelin”

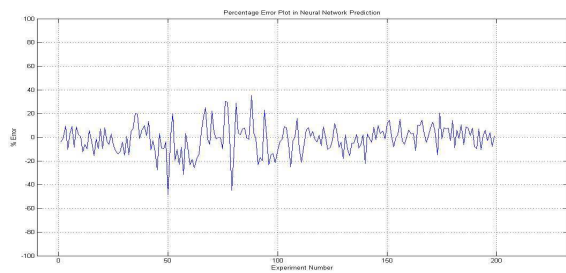


Fig - 47: Percentage error in predication with transfer Function “logsig, purelin”

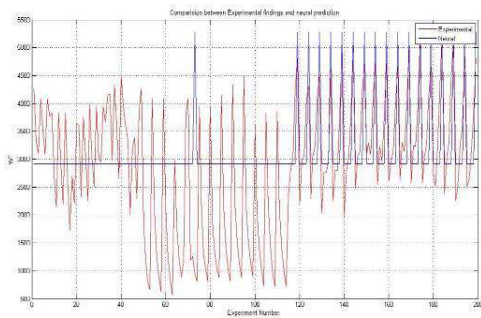


Fig - 48: Neural response with transfer Function “tansig, logsig”

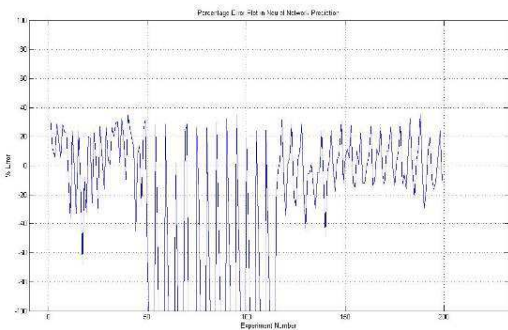


Fig - 49: Percentage error in predication with transfer Function “tansig, logsig”

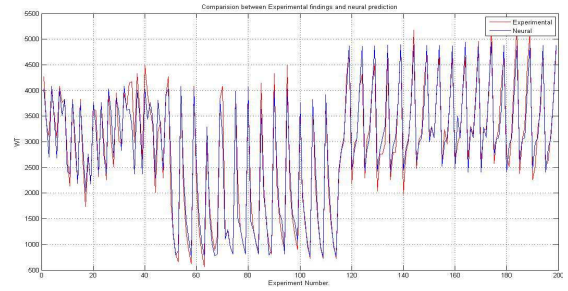


Fig - 50: Neural response with learning Function “learncon”

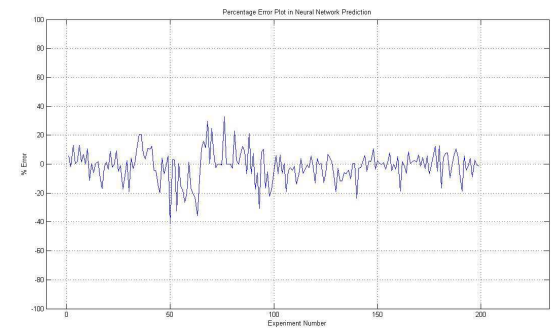


Fig - 51: Percentage error in predication with learning Function “learncon”

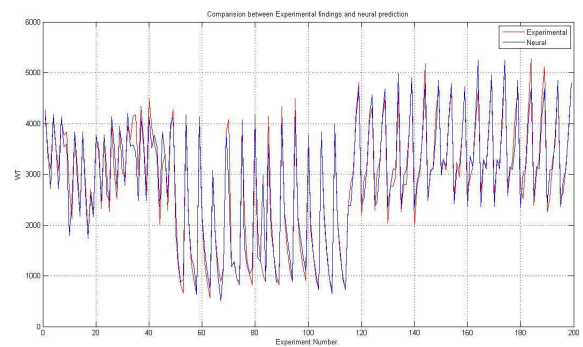


Fig - 52: Neural response with learning Function “learnrd”

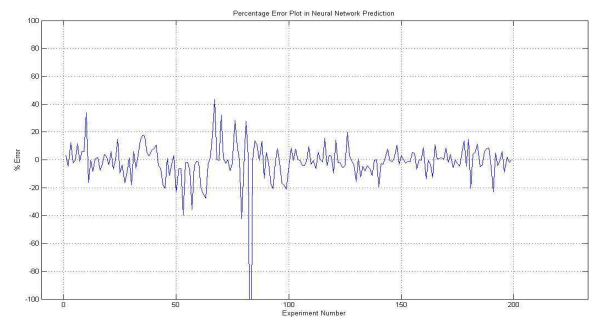


Fig - 53: Percentage error in predication with learning Function “learnrd”

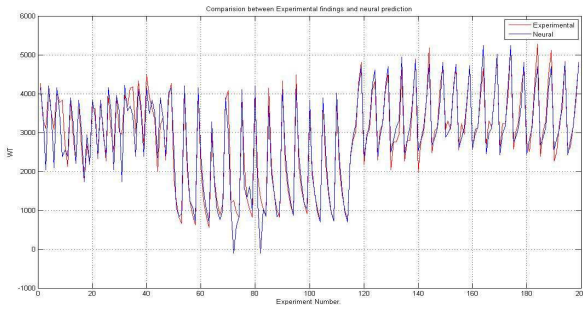


Fig - 54: Neural response with learning Function “learnh”

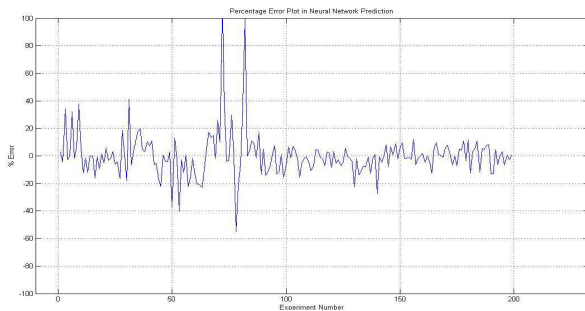


Fig - 55: Percentage error in prediction with learning Function “learnh”

5. DISCUSSION OF RESULTS

5.1 Effect of Variation of Number of Neuron in Hidden Layer

- Figure 4 to 23 shows that increase of number of neuron gives better prediction.
- But as the number goes to 700 it takes very large time to train the network
- Hence Layer size is limited to 500.

5.2 Effect of Variation Training Styles

- Figure 24 to 37 gives the results when training styles are changed.
- The results are too bad with “trainb” & “trainc”
- As a result it can be concluded that training styles affect the performance of network to great extent.
- It is been observed that back propagation training functions are better for fitting function and the performance seems superior with “trainlm”

5.3 Effect of Variation of Performance Function

- Figure 38 to 43 displays the results with change of performance function
- The change of performance function has shown little effect on prediction

- Performance function “mae” has shown better results for this case.

5.4 Effect of Variation Transfer Function to Hidden Layer

- Figure 44 to 49 shows the outcome when transfer functions to hidden layers were changed.
- There are too many transfer functions to use and network has two hidden layers. Hence various combinations of these transfer functions were use to see the effect on performance.
- It is been observed that the outer most layer with linear transfer function “purelin” gives better results
- The combination of “tansig, purelin” transfer functions is found best for this case.

5.5 Effect of Variation of Learning Function

- Figure 50 to 55 put view on the results with variation of learning function.
- It clearly shows the is very mild effect of variation learning function on performance of network
- Out of those learning functions “learncon” was better

CONCLUSIONS

The optimization methodology adopted is unique and rigorously derives the most optimum solution for field data available for Human Powered Flywheel Motor. The effect on prediction of network is observed very consciously with variation each ANN parameter.

It could also be concluded from the above results that for a fitting function type of situation “purelin” is the best transfer function at the outermost layer of the network. Deciding numbers of neuron in a layer are also crucial for performance of the network and this number depends on number of independent variable counting for the output.

The learning algorithm is also playing a vital role in performance of the network. Hence precise learning algorithm should be identified for a particular field problem. If compared with empirical model prediction ANN is predicting much better and the error is reduced to 20%.

FUTURE SCOPE

- Prediction of ANN model may be compared with the available empirical model.
- The best ANN model may be further used to develop another mathematical model.
- The ANN model could be validated through unseen data.
- This mathematical model then could be utilized to develop a physical controller.

REFERENCES

- [1]. Modak J. P. and Askhedkar R. D. "Hypothesis for the extrusion of lime flash sand brick using a manually driven Brick making machine", Building Research and Information U.K., V22,NI, Pp 47-54, 1994
- [2]. Modak J. P. and Bapat A. R. "Manually driven flywheel motor operates wood turning machine", Contepory Ergonomics, Proc. Ergonomics Society annual convension13-16April, Edinburg, Scotland, Pp 352-357, 1993.
- [3]. Sohoni V. V., Aware H. V. and Modak J. P. " Manual Manufacture of Keyed Bricks", Building Research and Information UK, Vol 25, N6, 1997, 354-364.
- [4]. Modak J. P."Design and Development of Manually Energized Process Machines having Relevance to Village/Agriculture and other productive operations, Application of manually energized flywheel motor for cutting of wood strip", Human Power, send for Publications.
- [5]. H. Schenck Junior "Theory of Engineering Experimentation", MC Graw Hill, New York.
- [6]. A. R. Lende, "Modelling of pedal driven flywheel motor by use of ANN", M. Tech. Thesis, PCE, Nagpur
- [7]. S. N. Shvanandam, "Introduction to Neural Network using Matlab 6.0", McGraw Hill publisher.
- [8]. Stamtiou V. Kartaplopoulos , Understanding Neural Networks and Fuzzy Logics, IEEE Press
- [9]. Neural Network Toolbox TM 7 User's Guide R2010a, Mathworks.com
- [10]. Rudra Pratap, "Getting Started with Matlab7," Oxford, First Indian Edition 2006.
- [11]. A. R. Bapat, "Experimental Optimization of a manually driven flywheel motor", M.E. Thesis, VNIT, Nagpur.
- [12]. A. R. Bapat, "Experimentation of Generalized experimental model for a manually droven flywheel motor", PhD Thesis, VNIT, Nagpur.

BIOGRAPHIES



Ms. A. R. Lende, Ex. Asst. Professor, MIT, Kothrud, Pune, MH, India. Ex. Asst. Professor, DMIETR, Wardha, MH, India. Qualification: BE. (Mech. Engg.), M.Tech.(Mech. Engg. Design) Ph.D. (Pursuing)



Dr. J. P. Modak, Emeritus Professor and Dean (R&D), PCE, Nagpur, MH, India