

Use of Artificial Neural Network and Fuzzy Logic in Optimization of Collapsible structure

Tabish Izhar¹, Kirti Singh², Neha Mumtaz³

¹Assistant Professor, Department of Civil Engineering, Integral University, Lucknow (U.P.), India

²M.Tech Student, Department of Civil Engineering, Integral University, Lucknow (U.P.), India

³Assistant Professor, Department of Civil Engineering, Integral University, Lucknow (U.P.), India

Corresponding Author: TabishIzhar

ABSTRACT:

Collapsible roof system is a new emerging technology of structural system which will be used for auditoriums according to the weather condition. It will open during fair and closed weather. Artificial neural network and fuzzy logic system are very effective and fast process of optimization which will be used for the non-linear data. Optimization is a process of selecting best solution from the set of available alternatives. The proposed ANN model is trained with observation data for predicting the optimization of collapsible roof. In this research work, optimization of collapsible roof structural design using real coded of artificial neural network and fuzzy logic from the MATLAB. This can optimize the structure in a very less period of time with effective and efficient results. These results are verified by using fuzzy logic toolbox of MATLAB in which the available crisp data is firstly converted into the fuzzy which is called as fuzzification and then optimization is performed by use of membership functions and conditions then after that the defuzzification will occur to convert the data into normal results.

KEYWORDS: Artificial Neural Network (ANN), Collapsible Roof, Fuzzy Logic, Optimization.

Date of Submission: 25-04-2019

Date of acceptance:05-05-2019

I. INTRODUCTION

Collapsible roof are the structures that can change shape from a compact to an expanded form. Thus their advantage over conventional roof is adaptability. Because there are certain situations where having moving components allows the structure to change shape according to changing need. The property of this roof is that that it can be used just once and multiple times according to the requirement. Light weight collapsible roof have number of applications, usually as a weather cover so that roof can remain open during fair weather and closed during poor weather. It can be frequently used for sports facility like in stadiums. These are made of flexible material, or be made of rigid panels attached with hinges. It is also termed as foldable or retractable roof. For the stability of retractable type of structures we have to provide extra members and locking systems for open and closed configuration.

Classification of collapsible roof structure:

These structures are in principle classified as sliding type and foldable type depending upon the type of movement and movable elements. While the fig. shows below sliding and movement type reactions and membrane attached to a cable showing movement of the roof to open or close position.

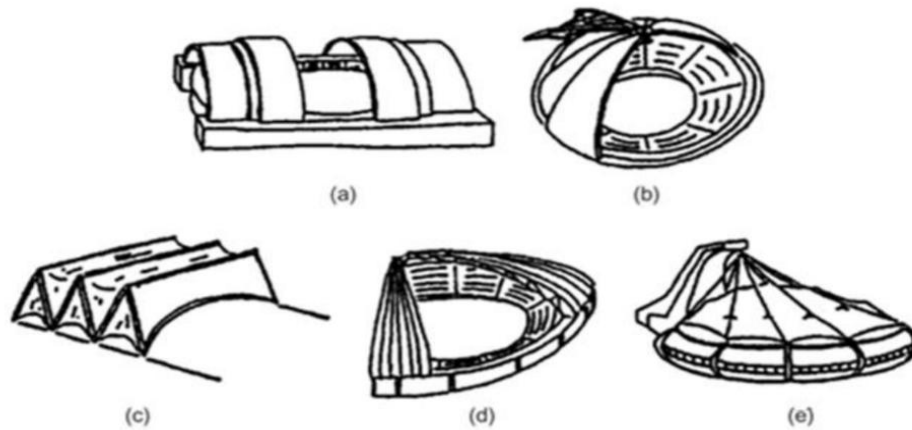


Figure-1: Classification of retractable roof structures

(Kassabian P.E, You Z. and Pellegrino S. (1999)

1.1.2 Performance of collapsible roof:

Performance of collapsible structures will totally depend on the site area and the type of requirement of structure. These structures are easy to transport due to light weight and having an open-closed configuration.

II. OBJECTIVE

From the case studies of literature review it is found that optimization by artificial neural network and fuzzy logic toolbox has played very important role in collapsible type of structure. Now days many types of optimization like for shape, size and topology is done by these advanced type of toolbox of MATLAB. Optimization of structure will depend on the area of structure, loading conditions and the members used in collapsible type of structure.

So by using these effective toolbox of MATLAB optimization can be done for the-

1. Self weight of the structure in stable condition
2. Use of minimum members in the structure.

III. OPTIMIZATION PROBLEM

A 25-bar scissor shown in “Figure-2” was considered to be optimized. The members are grouped in eight sets and contain set of discrete cross-sections with 32 elements. Table-2 contains all the design data. The effect of buckling was not included.

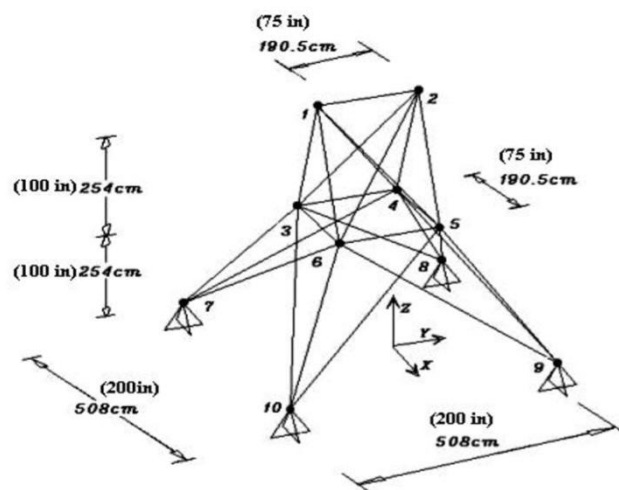


Figure -2: A 25- bar space scissor structure

(A. kaveh and saeedshojaee (2003))

“Table-1” and “Table-2” provides the all information about the grouping details, member properties and all design variables which will used in the structure.

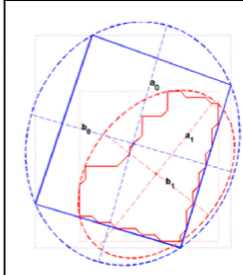
Table - 1: Details for grouping of the 25- bar space truss (A, kaveh and saeedshojaee(2003))

Group numbers	Design variables	Nodal number of the members	Member numbering
1	X ₁	1-2	1
2	X ₂	1-4,2-3,1-5,2-6	2,3,4,5
3	X ₃	2-5,2-4,1-3,1-6	6,7,8,9
4	X ₄	3-6,4-5	10,11
5	X ₅	3-4,5-6	12,13
6	X ₆	3-10,6-7,4-9,5-8	14,15,16,17
7	X ₇	3-8,4-7,6-9,5-10	18,19,20,21
8	X ₈	3-7,4-8,5-9,6-10	22,23,24,25

Table - 2: Design data used in the structure

Constraints data	Displacement constraints: In the direction of X and Y axis $ \Delta j \leq 0.889\text{cm}(0.35\text{in})$ Stress constraints: $ \sigma_j \leq 275.6 \text{ Mpa (40Ksi)}$ $i = 1,2,\dots,25$			
Loading data	Nodal number	Px (KN)	Py(KN)	Pz(KN)
	1	4.545	-44.537	-44.537
	2	0	-44.537	-44.537
	3	2.227	0	0
	6	2.672	0	0
Material properties	Modulus of elasticity $E = 6.895 \times 10^4 \text{ Mpa (104 Ksi)}$ Weight density of the material $\rho = 0.0272 \text{ N/cm}^3 (0.1\text{lb/in}^3)$			

The table below illustrates the information of shape factor and their values which will use during optimization process.

	Shape Feature	Value
	<i>RoofsAreaRatio</i>	0.5105
	<i>BoundingBoxRatio</i>	0.5009
	<i>BoundingBoxFilling</i>	0.6458
	<i>MajorAxesRatio</i>	0.7587
	<i>MinorAxesRatio</i>	0.6309
	<i>DirectionStability</i>	0.7546
	<i>EllipseFilling</i>	0.6867

The table -3 provides the information of several nodes conditions in the binary form which will use in optimization process. By using this table the data will load in the neural network fitting tool in the form and input and output vector. After the loading of data in MATLAB code threshold function is selected for training of data we will uses the LEVENBEG-MARQUARDT back propagation algorithm (trainlm) function. And provides the number of hidden layers or the number of iteration for which the program run. Then after normalizing of input and output data will be done and training pair values are given to the system. In which its 70 percent is used for training of data and remaining 30 percent will use for the test and validation.

Table -3: Nodal information and several nodes of the structure

Nodal Information											
Nn	Nx	Ny	ux	Uy	tz	Px	Py	Mz	Fx	Fy	Fr
1	0	0	0	0	0	0	0	0	1	1	0
2	5	0	0	0	0	0	0	0	1	1	1
3	1	5	0	0	0	0	0	0	0	0	0
4	5	5	0	0	0	0	0	0	0	0	0

IV. RESULTS AND DISCUSSION

The above data which will presented in the tables will be used for the optimization process in neural network and fuzzy logic toolbox of MATLAB by using these data optimization results will be generated in the form of graphs at several point and R- value.

Optimization results of neural network toolbox in MATALB:

In the neural network toolbox data should uploaded in the vector forms in the input and output form and number of iteration and hidden layers provided manually.

“Figure 1” illustrate about the training histogram of above data with 20 bins. Where bins are represent about number of vertical bars in the structure or graph at the time of optimization. The blue, green and red bars represent training data, validation data and testing data respectively. The zero error line provides the information about the square error value. Similarly “Figure 1” illustrate about the error histogram training data with 20bins at several instances.

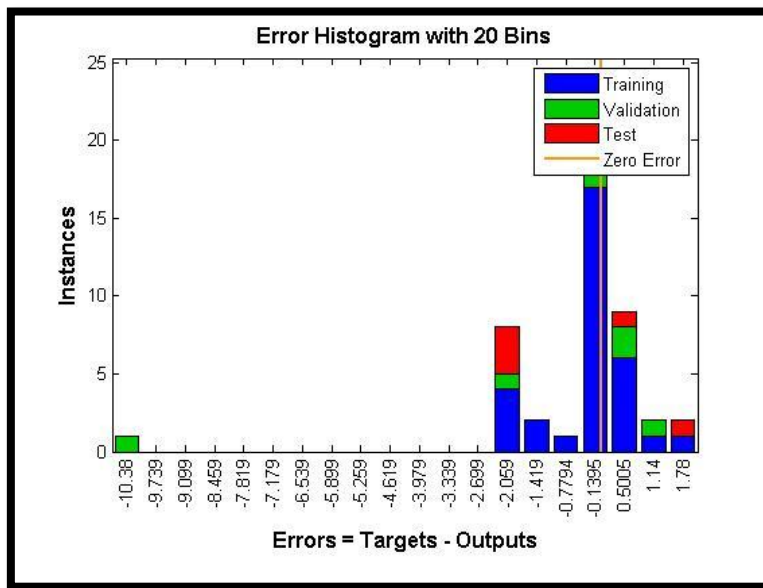


Figure-1: Neural network training error histogram epoch 5 minimum gradients reached

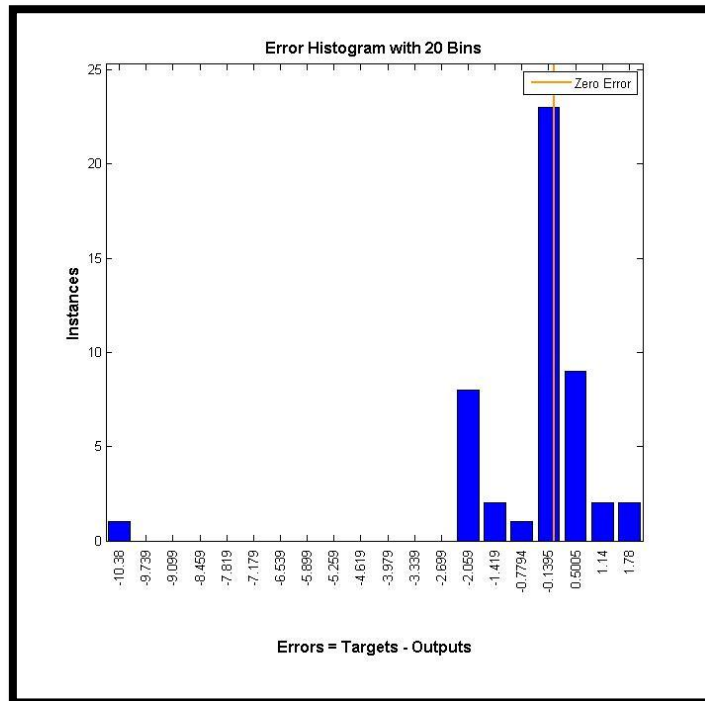


Figure-2: Error histogram of training data with 20 Bins

“Figure 3” and figure 4” will illustrates the regressions plots for the training, testing and validation data. Regressions plots tell us about the how independent variables will relate to the dependent variables and data will fit with the random pattern of regression which is best for the model.

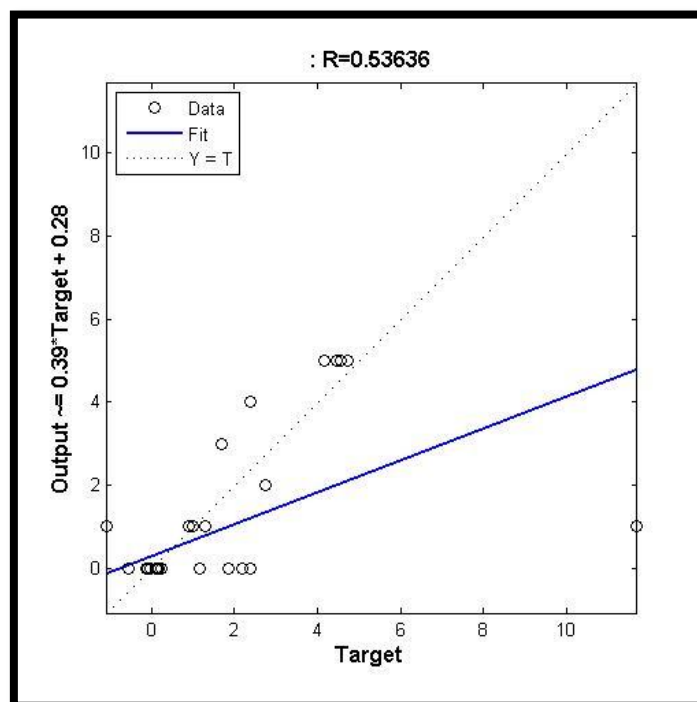


Figure-3: Regression plot for output data

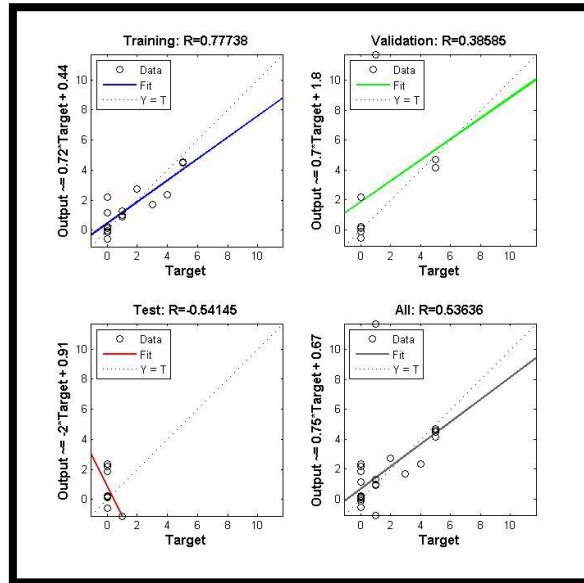


Figure-4: Regression plots for training validation and test

“Figure 5” illustrate validation check graphs for the full optimization process it will shows the training progress of data and the gradients at 5 epoch. Where epoch is the one complete iteration used to update the weight values.

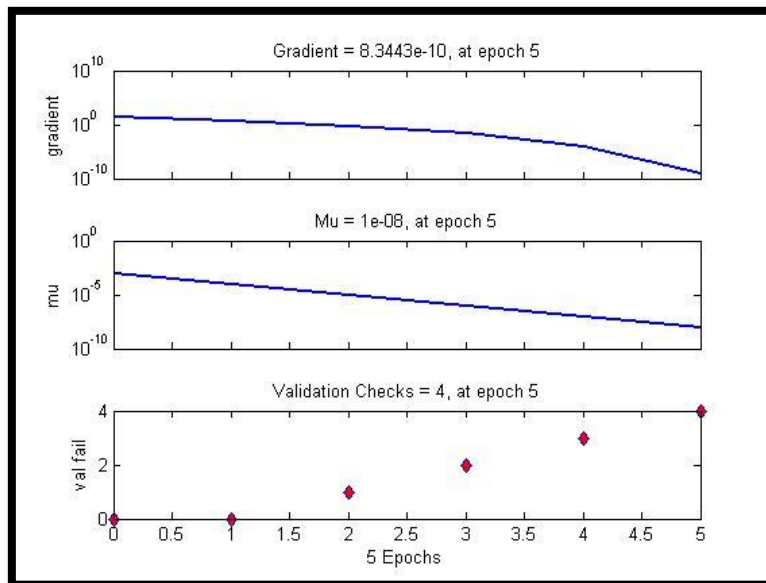


Figure-5: Neural network training state validation plot

Optimization results generated by fuzzy logic toolbox of MATLAB:

In the fuzzy logic toolbox of MATLAB optimization is done by making the membership functions and by providing its range.

“Figure 6” illustrate optimization results of fuzzy inference system (FIS) with membership where the conditions for load case and weights are applied. It will present about the train data and also sample index of mean square value and root mean square value. Similarly “figure 7” and Figure 8” illustrate the optimization results for different error and standard deviation values.

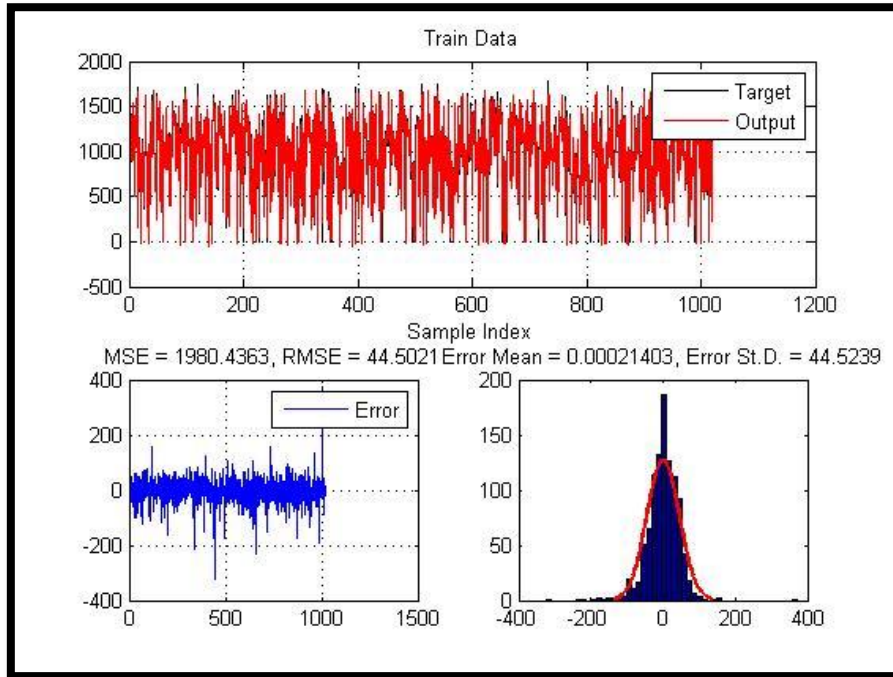


Figure-6: Fuzzy behaviour of training data at several points

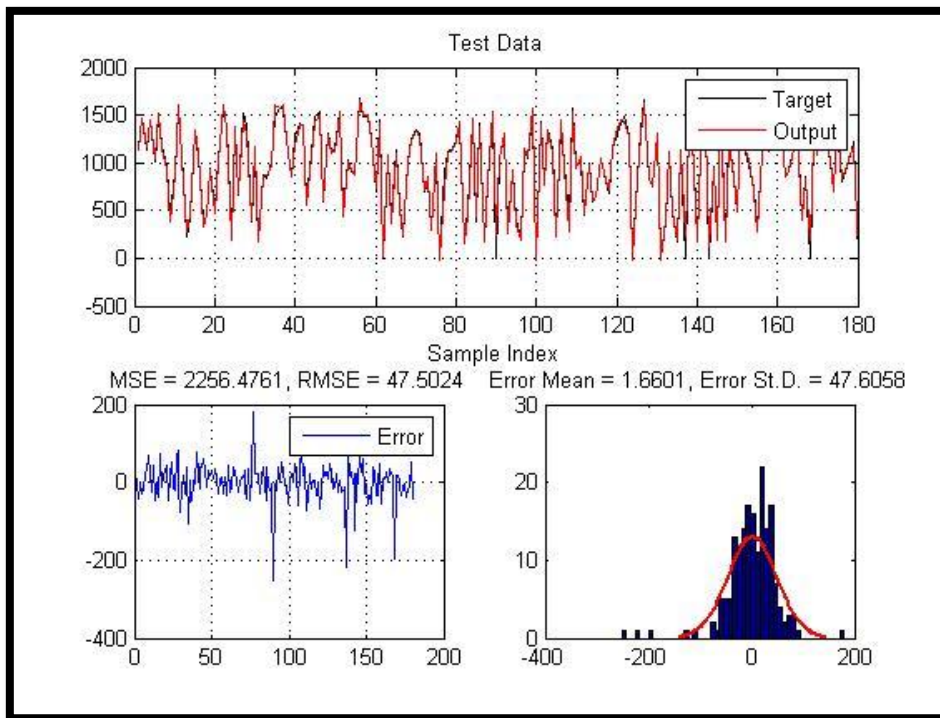
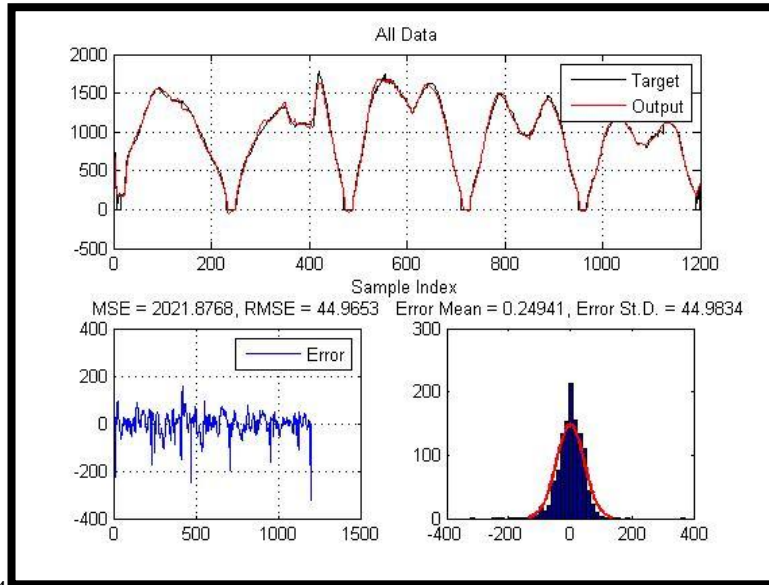


Figure-7: Optimization results for test data at error 1.6601



4

Figure-8: Optimization result for error mean of 0.24941

“Figure 9” illustrate about the regression plots of train data and test data with R- value 0.99546 and 0.99463 respectively.

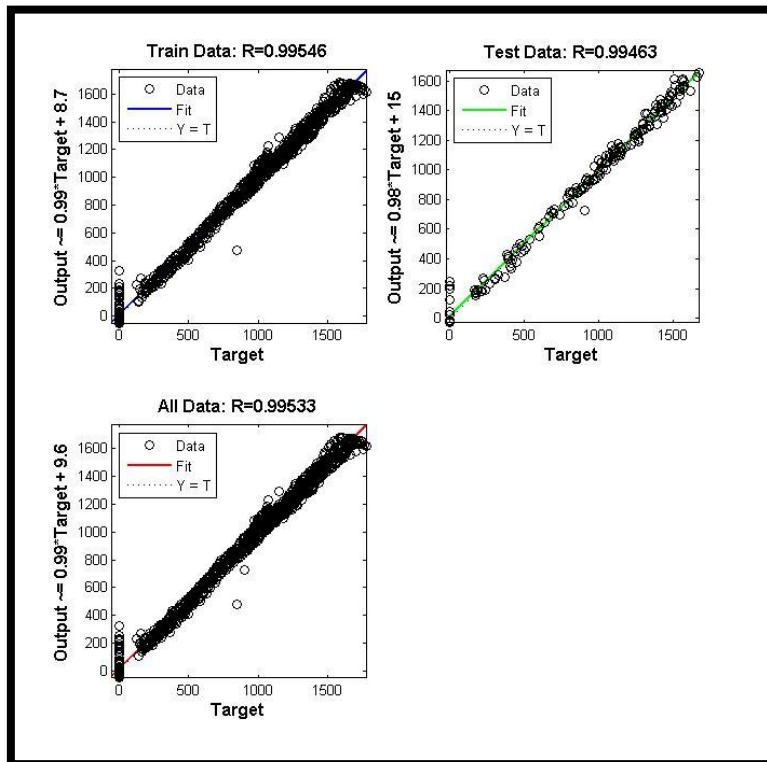


Figure-9: Regressions plots of training and testing data

The simulink behaviour of the optimization results is generated by fuzzy logic which will shown in the “Figure 10”.

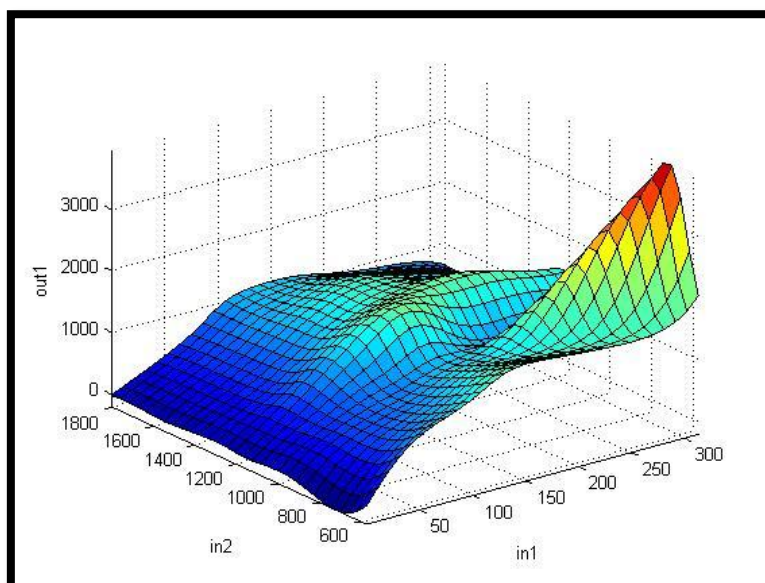


Figure-10: SIMULINK behaviour of optimization data

V. CONCLUSION

After the neural network optimization and fuzzy optimization we can conclude that these are very effective toolboxes of MATLAB for the optimization process because it will take less time and work should be completed in a very efficient way so these can conclude for the optimization problem. The data present above provide the good optimal designs and convergence. The optimization results by fuzzy are more appropriate when the qualitative statements are made from the objective problem. In this the proposed algorithms is based on the FIS representation of object in terms of neural network and fuzzy sets.

ACKNOWLEDGEMENT

Authors of this paper would like to extent their gratitude to Integral University for supporting this research work.

REFERENCES

- [1]. Paolo dadone, design optimization of fuzzy logic system, doctoral dissertation, virgina polytechnic institute ans state university (2001).
- [2]. A. Kaveh, Saeed Shojaee, optimal design of scissor-like foldable structures using genetic algorithm. Asian journal of civil engineering, vol. 4, 115-133(2003).
- [3]. Andrew H. Frazer Design consideration of retractable-roof stadia, masters thesis, Massachusetts institute of technology (2005).
- [4]. E.Salajegheh, J. Salajegheh, S.M. Seyedpoor and M. Khatibinia, Optimal Design Of Geometrically Nonlinear Space Truss Using Adaptive Neuro-Fuzzy Intercence System. Department of civil engineering, University of Kerman, vol.16, 403-414(2009).
- [5]. A.R. Gaiduk, Y.A>VErshinin, M.J. West, Neural networks and optimization problem, IEEE international conference on control applications (2002).
- [6]. Achille Massac, Optimization in practice with MATLAB for engineering students and professionals, Cambridge university press (2015).
- [7]. Howard Demuth Mark Beale, Neural network toolbox, for use with MATALB, version 4, (release 13), (2012) online only.
- [8]. RudraPratp, Getting started with MATLAB, Oxford university press, Newyork (2010).
- [9]. S.N. Sivanandam, S. Sumathi and S.N. Deepa, introduction to fuzzy logic using matlab, Springer (2007).
- [10]. Christos Xenophontos, A beginner's guide to matlab, Clarkson University, (1999).
- [11]. Javier Ferreiro-Cabello, Esteban fragile-Garcia, Eduardo Martinez de PisonAscacibar, Fco. Javier Martinez de PisonAscacibar. Metamodels-based design optimization of structural one-way slabs based on deep learning neural networks to reduce environmental impact. Engineering structures 155, 91-101(2018).
- [12]. Thomas Bhul, Frank V. Jensen, Sergio Pellegrino ,Shape optimization of cover plates for retractable roof structures. Computers and structures 82, 1227-1236'(2004).
- [13]. Tang jiafu Wang Dingwei, Richard Y K Fung, Kai-Leung Yung ,Understanding of fuzzy optimization: theories and methods. Journal of system science and complexity vol. 17, 117-136(2004).
- [14]. JianguoCai, Jian Feng, Chao Jiang ,Development and analysis of a long-span retractable roof structure. Journal of constructional steel research 92, 175-182(2014).
- [15]. Kassabian PE, You Z. Pellegrino S. retractable roof structures, proceedings of institution of civil engineering, structures and buildings, ,45-56,(1999).

Appendix -A

Script for fuzzy logic in MATLAB:

```
clc;
clear;
close all;

%% Create Time-Series Data

data = load('my_data');

% About Engine Dataset
% Inputs:
% 1. Fuel Rate
% 2. Speed
% Targets:
% 1. Torque
% 2. Nitrous Oxide Emissions

Inputs = data.Inputs';
Targets = data.Targets';

nData = size(Inputs,1);

Targets = Targets(:,2); % Select 1st Output to Model

%% Shuffling Data

PERM = randperm(nData); % Permutation to Shuffle Data

pTrain=0.85;
nTrainData=round(pTrain*nData);
TrainInd=PERM(1:nTrainData);
TrainInputs=Inputs(TrainInd,:);
TrainTargets=Targets(TrainInd,:);

pTest=1-pTrain;
nTestData=nData-nTrainData;
TestInd=PERM(nTrainData+1:end);
TestInputs=Inputs(TestInd,:);
TestTargets=Targets(TestInd,:);

%% Selection of FIS Generation Method

Option{1}='Grid Partitioning (genfis1)';
Option{2}='Subtractive Clustering (genfis2)';
Option{3}='FCM (genfis3)';

ANSWER=questdlg('Select FIS Generation Approach:',...
    'Select GENFIS',...
    Option{1},Option{2},Option{3},...
    Option{3});
pause(0.01);

%% Setting the Parameters of FIS Generation Methods

switch ANSWER
case Option{1}
    Prompt={'Number of MFs','Input MF Type:','Output MF Type:'};
    Title='Enter genfis1 parameters';
```

```
DefaultValues={'5', 'gaussmf', 'linear'};

PARAMS=inputdlg(Prompt,Title,1,DefaultValues);
pause(0.01);

nMFs=str2num(PARAMS{1}); %#ok
InputMF=PARAMS{2};
OutputMF=PARAMS{3};

fis=genfis1([TrainInputsTrainTargets],nMFs,InputMF,OutputMF);

case Option{2}
    Prompt={'Influence Radius:'};
    Title='Enter genfis2 parameters';
    DefaultValues={'0.3'};

    PARAMS=inputdlg(Prompt,Title,1,DefaultValues);
    pause(0.01);

    Radius=str2num(PARAMS{1}); %#ok

fis=genfis2(TrainInputs,TrainTargets,Radius);

case Option{3}
    Prompt={'Number fo Clusters:',...
            'Partition Matrix Exponent:',...
            'Maximum Number of Iterations:',...
            'Minimum Improvemnet:'};
    Title='Enter genfis3 parameters';
    DefaultValues={'15', '2', '200', '1e-5'};

    PARAMS=inputdlg(Prompt,Title,1,DefaultValues);
    pause(0.01);

nCluster=str2num(PARAMS{1});    %#ok
Exponent=str2num(PARAMS{2});    %#ok
MaxIt=str2num(PARAMS{3});    %#ok
MinImprovment=str2num(PARAMS{4});    %#ok
DisplayInfo=1;
FCMOptions=[Exponent MaxItMinImprovmentDisplayInfo];

    fis=genfis3(TrainInputs,TrainTargets,'sugeno',nCluster,FCMOptions);
end

%% Training ANFIS Structure

Prompt={'Maximum Number of Epochs:',...
        'Error Goal:',...
        'Initial Step Size:',...
        'Step Size Decrease Rate:',...
        'Step Size Increase Rate:'};
Title='Enter genfis3 parameters';
DefaultValues={'200', '0', '0.01', '0.9', '1.1'};

PARAMS=inputdlg(Prompt,Title,1,DefaultValues);
pause(0.01);

MaxEpoch=str2num(PARAMS{1});    %#ok
ErrorGoal=str2num(PARAMS{2});    %#ok
```

```
InitialStepSize=str2num(PARAMS{3});    %#ok
StepSizeDecreaseRate=str2num(PARAMS{4});    %#ok
StepSizeIncreaseRate=str2num(PARAMS{5});    %#ok
TrainOptions=[MaxEpoch ...
ErrorGoal ...
InitialStepSize ...
StepSizeDecreaseRate ...
StepSizeIncreaseRate];

DisplayInfo=true;
DisplayError=true;
DisplayStepSize=true;
DisplayFinalResult=true;
DisplayOptions=[DisplayInfo ...
DisplayError ...
DisplayStepSize ...
DisplayFinalResult];

OptimizationMethod=1;
% 0: Backpropagation
% 1: Hybrid

fis=anfis([TrainInputs TrainTargets],fis,TrainOptions,DisplayOptions,[],OptimizationMethod);

%% Apply ANFIS to Data

Outputs=evalfis(Inputs,fis);
TrainOutputs=Outputs(TrainInd,:);
TestOutputs=Outputs(TestInd,:);

%% Error Calculation

TrainErrors=TrainTargets-TrainOutputs;
TrainMSE=mean(TrainErrors.^2);
TrainRMSE=sqrt(TrainMSE);
TrainErrorMean=mean(TrainErrors);
TrainErrorSTD=std(TrainErrors);

TestErrors=TestTargets-TestOutputs;
TestMSE=mean(TestErrors.^2);
TestRMSE=sqrt(TestMSE);
TestErrorMean=mean(TestErrors);
TestErrorSTD=std(TestErrors);

%% Plot Results

figure;
PlotResults(TrainTargets,TrainOutputs,'Train Data');

figure;
PlotResults(TestTargets,TestOutputs,'Test Data');

figure;
PlotResults(Targets,Outputs,'All Data');

if ~isempty(which('plotregression'))
    figure;
    plotregression(TrainTargets, TrainOutputs, 'Train Data', ...
```

```
TestTargets, TestOutputs, 'Test Data', ...  
    Targets, Outputs, 'All Data');  
    set(gcf, 'Toolbar', 'figure');  
end
```

```
figure;  
gensurf(fis, [1 2], 1, [30 30]);  
xlim([min(Inputs(:,1)) max(Inputs(:,1))]);  
ylim([min(Inputs(:,2)) max(Inputs(:,2))]);
```

APPENDIX - B

```
function PlotResults(targets, outputs, Name)
```

```
    errors=targets-outputs;  
  
    MSE=mean(errors.^2);  
    RMSE=sqrt(MSE);  
  
    error_mean=mean(errors);  
    error_std=std(errors);  
  
    subplot(2,2,[1 2]);  
    plot(targets,'k');  
    hold on;  
    plot(outputs,'r');  
    legend('Target','Output');  
    title(Name);  
xlabel('Sample Index');  
grid on;  
  
    subplot(2,2,3);  
    plot(errors);  
    legend('Error');  
    title(['MSE = ' num2str(MSE) ', RMSE = ' num2str(RMSE)]);  
    grid on;  
  
    subplot(2,2,4);  
    histfit(errors, 50);  
    title(['Error Mean = ' num2str(error_mean) ', Error St.D. = ' num2str(error_std)]);  
  
end
```

TabishIzhar" Use of Artificial Neural Network and Fuzzy Logic in Optimization of Collapsible Roof" International Journal of Computational Engineering Research (IJCER), vol. 09, no. 4, 2019, pp 38-50