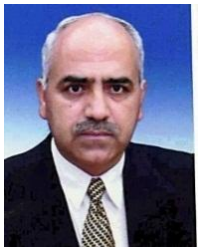
Simplified Methods for Crack Risk Analyses of Early Age Concrete Part 2: Restraint Factors for Typical Case Wall-on-Slab

Majid Al-Gburi
M.Sc. Ph.D. Student
Lulea University of Technology – Sweden
Dept. of Structural Engineering
Email: majid.al-gburi@ltu.se

Dr. Jan–Erik Jonasson
Professor
Lulea University of Technology – Sweden
Dept. of Structural Engineering
Email: jej@ltu.se



Dr. S. T. Yousif
Assistance professor
University of Mosul – Iraq
Email:styousif59112@yahoo.com

Dr. Martin Nilsson
Lulea University of Technology – Sweden
Email: martin.c.nilsson@ltu.se

ABSTRACT

Existing restraint curves have been applied to the method of artificial neural networks (ANN) to model restraint in the wall for the typical structure wall-on-slab. It has been proven that ANN is capable of modeling the restraint with good accuracy. The usage of the neural network has been demonstrated to give a clear picture of the relative importance of the input parameters. Further, it is shown that the results from the neural network can be represented by a series of basic weight and response functions. Thus, the results can easily be made available to any engineer without use of complicated software.

Key Words: Restraint curves, early age concrete, wall-on-slab, artificial neural network

1. INTRODUCTION

The main reason of stresses in young concrete is restraining of volume deformations caused by temperature and moisture changes at early ages [1]. Through cracking of the newly cast concrete is the most severe situation, as it occurs during the temperature contraction phase and as the crack remains open, see the position of the vertical dashed-dotted line in figure 1. To be able to realize estimations of through cracking the external restraint from adjacent structures needs to be known. Examples of calculation of restraint curves and how they are applied into estimations of risks for through cracking are shown in [2].

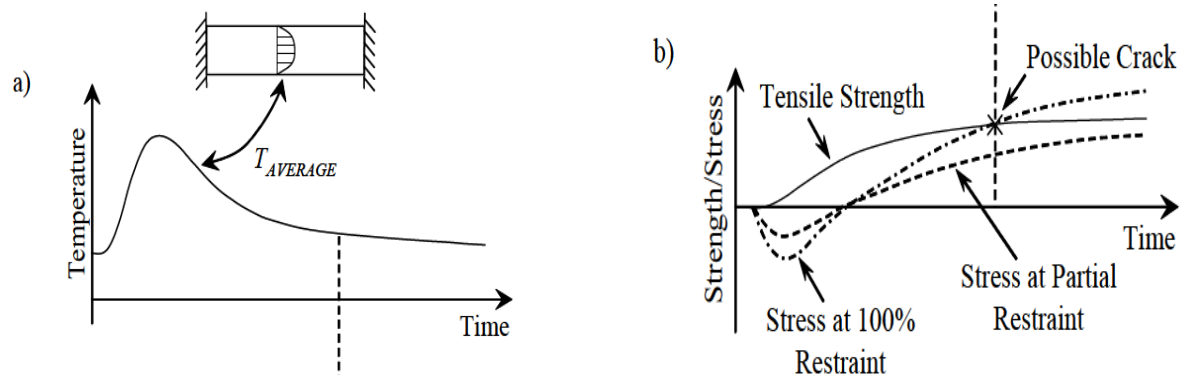


Figure 1 –Illustration of average temperature in young concrete and possible through cracking for different restraint conditions [1].

External restraint includes effects from adjacent structures like earlier casting sequences, foundations and subsoil. The degree of external restraint depends primarily on the relative dimensions and modulus of elasticity in the young concrete as well as in the surrounding restraining materials. The distribution of restraint varies at different positions of a concrete member [2].

The restraint in a section may be reduced in several ways, as for instance by favourable casting sequences or shortening the length of the section and suitable arrangements of construction joints. It is also possible to mitigate early age through cracking by the choice of a concrete mix with low temperature rise due to hydration or lower the casting temperature [3]. Most common measures on site is to cool the newly cast concrete or to heat the adjacent structure, [1] and [3].

When analyzing early age stresses in concrete based on restraint curves, we might use the local restraint method (LRM) or the equivalent restraint method (ERM) [2]. Without measures on site the application of LRM is obvious, and the measure cooling can be applied with both LRM and ERM, but heating can only be analyzed with ERM [2].

2. AIMS AND PURPOSES

The aims and purposes of this paper are to

- Apply and verify the use of artificial neural network for restraint curves concerning the typical structure wall-on-slab.
- Clarify the influences of geometrical dimensions on restraint in the wall.
- Develop a simplified method for practical application of the neural network for the typical case wall-on-slab.

3. ESTIMATION OF RESTRAINT IN EARLY AGE CONCRETE

In the literature there are many methods adopted to estimate and calculate the value of restraint in young concrete, see for example [4], [5], [6], [7] and [8]. Some of these methods need the use of a complex software, which usually is expensive and need experienced people.

In this study, an artificial neural network (ANN) is presented to calculate the amount of restraint in the wall for typical structure wall-on-slab. The analyzes are based on results from 2920 elastic finite element calculations of the restraint in the wall founded on a slab [1], where the geometrical dimensions of the wall and the slab are varied systematically within reasonable values. The resulting restraints are fed and verified by an ANN, and the outcome from ANN are transformed to an Excel spread sheet to make the estimation of restraints quick and easy to apply for any engineer. This saves both time and money at estimation of the restraint curve for walls founded on a slab.

3.1 Geometric effects on restraint for early age concrete

The degree of restraint depends on several factors, including geometry of structures, casting sequences, number and position of joints, and mechanical properties of materials. The effects from restraint are illustrated in the upper right part of figure 2 [3] as one essential part of a crack risk estimation for early age concrete.

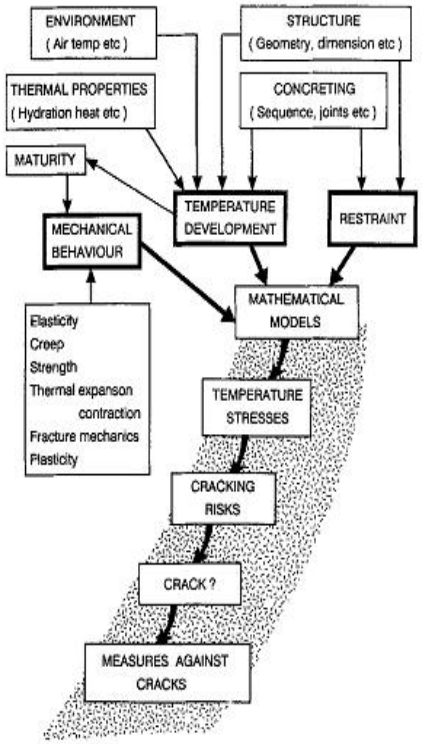


Figure 2 - Factors influencing stresses and cracking in early age concrete [3].

The restraint is reflected as a balance between the new concrete volume and the existing adjacent structure. In general, a larger volume of the new concrete results in a lower restraint while a small volume results in a high restraint, [9], [10] and [11].

The next chapter includes calculation of restraint in walls using the method of an artificial neural network based on geometric dimensions of the typical structure wall-on-slab.

4. APPLICATION OF THE ARTIFICIAL NEURAL NETWORK METHOD (ANN)

4.1 General overview

One form of artificial intelligence is the ANN, which attempts to mimic the function of the human brain and nerve system, but a simple unit of a neural network is much simpler than the biological cell [12].

A typical structure of ANN consists of a number of processing elements (PEs), or neurons, that usually are arranged in an input layer, an output layer and one or more hidden layers between, see figure 3 [13]. Each PE in a specific layer is fully or partially joined to many other PEs via weighted connections. The input from each PE in the previous layer (x_i) is multiplied by an adjustable connection weight (w_{ji}).

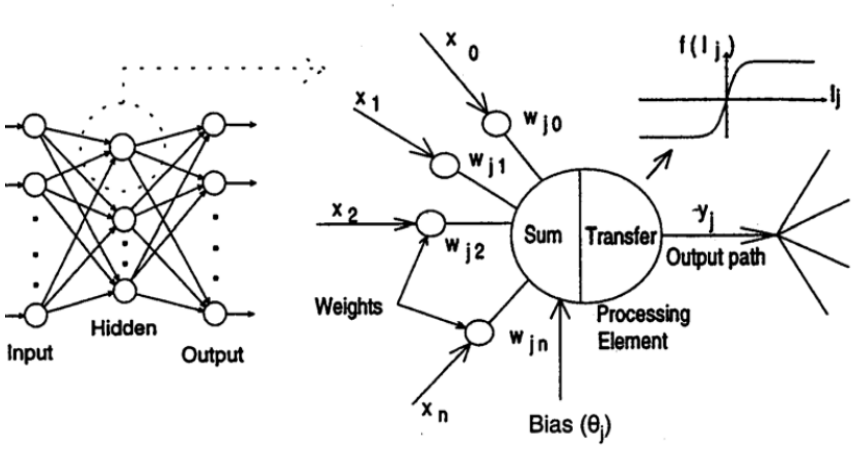


Figure 3 - Structure and operation of an ANN [13].

At each PE, the weighted input signals are summed and a threshold value or bias (θ_j) is added. This combined input (I_j) is then passed through a nonlinear transfer function, e.g. a sigmoid transfer function, to produce the output of the PEs (y_j). The output of one PE provides the input to the PEs in the next layer. This process is illustrated in figure 3, and explains in the next paragraph.

To determine the number of hidden neurons a network should have to perform its best, and one is often left out to the method of trial and error [14]. If the numbers of neurons are increased too much, over fit will occur, i.e. the net will have a problem to generalize. Each connection has a strength or weight that is used to modify the output of the neurons. The weights can be positive, which will tend to make the neuron go high, or negative, which will tend to make the neuron go low. The training process changes these weights to get the correct answers.

4.2 Learning an ANN

Artificial neural network models

Always we divide the data collected from field data or finite element programs in two groups. The first group is used in the training of the neural network (NN), and the other data group is used to test the obtained networks, Perceptron Multilayer (PML) networks, with a back-propagation algorithm used for the training. The multi-layer feed forward back-propagation technique is implemented to develop and train the neural network of current research, where the sigmoid transform function is adopted.

The Levenberg-Marquardt (LM) technique's built in MATLAB proved to be efficient training functions, and therefore, it is used to construct the NN model, [15] and [16]. This training function is one of the conjugate gradient algorithms that started the training by searching in the steepest descent direction (negative of the gradient) on the first iteration. The LM algorithm is known to be significantly faster than the more traditional gradient descent type algorithms for training neural networks. It is, in fact, mentioned as the fastest method for training moderately sized feed-forward neural network [14]. While each iteration of the LM algorithm tends to take longer time than each repetition of the other gradient descent algorithms, the LM algorithm yields far better results using little iteration, leading to a net saving in computer processor time. One concern, however, is that it may over fit the data. The network should be trained to recognize general characteristics rather than variations specific to the data set used for training.

Network data preparation

Pre-processing of data by scaling was carried out to improve the training of the neural network. To avoid the slow rate of learning near end points specifically of the output range due to the property of the sigmoid function, which is asymptotic to values 0 and 1, the input and output data were scaled between the interval 0.1 and 0.9. The linear scaling equation is expressed by:

$$y = \left(\frac{0.8}{\Delta}\right) X + \left(0.9 - \frac{0.8\, X_{max}}{\Delta}\right) \quad (1)$$

Eq. 1 was used in this study for a variable limited to minimum (X_{min}) and maximum (X_{max}) values given in table 1, with:

$$\Delta = X_{max} - X_{min} \quad (2)$$

It should be noted that any new input data should be scaled before being presented to the network, and the corresponding predicted values should be un-scaled before use, [12] and [14].

Back propagation algorithm

The back propagation algorithm is used to train the BPNN (Back Propagation Neural Network). This algorithm looks for the minimum error function in weight space using the method of gradient descent. The combination of weights that minimizes the error function is considered to be a solution to the learning problem. The algorithm can be described in the following steps, [15] and [16]:

1. Once the input vector is presented to the input layer it calculates the input to the hidden layer, h_j^H, as:

$$h_j^H = \theta_j + \sum_{i=1}^{NI} w_{ji} x_i \quad (3)$$

where x_i represents the input parameter; θ_j represents the bias function of hidden layer; NI represent the number of neuron in the input layer; and w_{ji} represents the weight factor between input and hidden layer.

Each neuron of the hidden layer takes its input, h_j^H, and uses it as the argument for a function and produces an output, Y_j^H, given by:

$$Y_j^H = f(h_j^H) \quad (4)$$

2. Now the input to the neurons of the output layer, h_k^0, is calculated as:

$$h_k^o = \theta_k + \sum_{j=1}^{NH} w_{kj} Y_j^H \quad (5)$$

where θ_k represents the bias function of output layer; w_{kj} represents the weight factor between hidden and output layer; and NH represents the number of neuron in the hidden layer.

3. The network output, y_k, is then given by:

$$y_k = f(h_k^o) \quad (6)$$

where f represents the activation function.

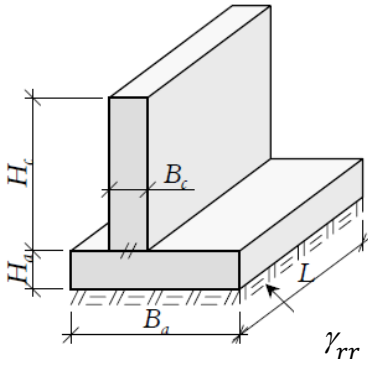
Training and testing of the neural networks

In [1] the geometry of 2920 wall-on-slab cases has been varied as shown in table 1. 2803 of them were used in the training of the neural network, as shown in figure 5a, and 117 were used for tests with the obtained network, as shown in figure 5b. Perception Multilayer (PML) networks, with a back-propagation algorithm, were used for the training. The multi-layer feed forward back-propagation technique is implemented to develop and train the neural network of current research, where the sigmoid transform function is adopted.

The training and testing results are given in figure 5 at the position $y/H_c = 0.1$, where y is the vertical co-ordinate above the upper surface of the slab. This height position is usually near the critical point at design with respect to the risk of through cracking in walls for typical structure wall-on-slab, [1] and [3]. As can be seen in the figure the coefficient of correlation, R, is 0.989 at training and 0.992 at verification, which indicates that the resulting model is very good.

Table 1 - List of parameters and their values used in the finite element method calculations of the elastic restraint variations in the walls of wall-on-slab structures [1].

Parameter	Sample	Maximum	Minimum	Unit
Slab width	B_a	8	2	m
Wall width	B_c	1.4	0.3	m
Slab thickness	H_a	1.8	0.4	m
Wall height	H_c	8	0.5	m
Length of the structure	L	18	3	m
External rotational restraint	γ_{rr}	1	0	-
Relative location* of the wall on slab	ω	1	0	-



*) ω = 0 means a wall placed in the middle of the slab;
ω = 1 means a wall placed along the edge of the slab.

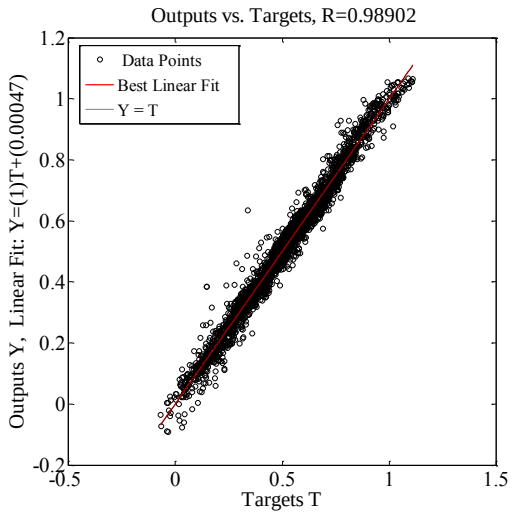


Figure 5a - Training results of ANN model at 0.1 wall height.

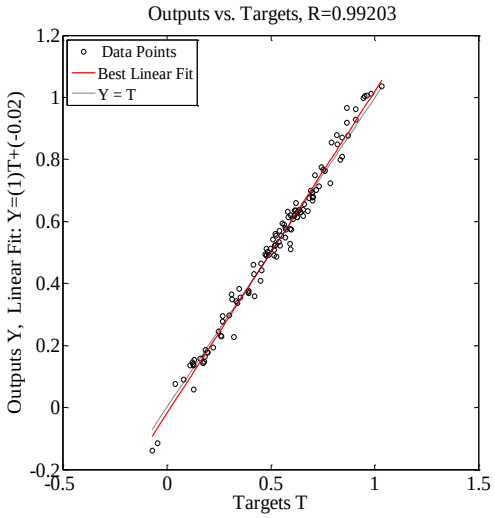


Figure 5b - Comparison between FEM-calculation and ANN model at 0.1 wall height.

5. STUDY OF IMPORTANCE GEOMETRY FACTORS MODEL

The method of the partitioning weights, proposed by Garson [17] and adopted by Goh [18], was used within this study in order to determine the relative importance of the various input parameters, see table 2. The major important parameter influencing the restraint is the wall height (H_c) at all levels of the wall (y/H_c) following by the external rotational restraint (γ_{rr}). The same indication is shown in [1]. Thereafter follow the wall thickness (B_c), and the length (L) of the structure. The relative location of the wall on the slab (ω) has a high impact in the lower part of the wall, and the effect decreases upward the wall. The thickness of the slab (H_a) has a little effect, and smallest influence has the width of the slab (B_a).

Table 2 - Relative importance on restraint of input parameters for wall-on-slab.

y/H_c	B_a	H_a	B_c	H_c	L	γ_{rr}	ω
0.0	4.65	12.61	19.62	18.63	5.85	17.85	20.76
0.1	6.97	11.84	17.7	20.11	11.24	20.44	11.66
0.2	7.3	13.6	14.14	21	11.68	15.65	16.63
0.3	6.8	11.0	11.6	26.2	12.4	16.8	15.2
0.4	8.89	12.87	12.47	21.47	12	16.6	15.68
0.5	7.68	10.15	12.9	25.57	15.36	13.37	14.56
0.6	6.66	8.04	11.36	29.84	16.46	21.15	6.47
0.7	5.99	9.75	10.9	30.97	16.41	19.12	6.82
0.8	6.73	7.44	9.61	32.1	14.95	23.16	5.98
0.9	6.36	5.36	8.01	42.23	13.67	18.9	5.43
1.0	4.13	4.42	8.01	41.6	23.15	14.67	3.97

6. STUDY OF PARAMETERS INFLUENCING THE RESTRAINT

In this chapter the restraints effects from different parameters are presented for the height y = 0.1H_c. Weight and threshold values for all heights, from y =0 to y = H_c, are shown in appendix A. In figures 6-11, the following symbols are used: Ba = B_a, Ha = H_a, Bc = B_c, Hc = H_c, Grr = γ_{rr} and w = ω.

6.1 Effect of wall height (H_c)

The wall height is the most important factor affecting the degree of restraint in the case wall-on-slab, as shown in table 2. Generally, the degree of restraint decreases with an increase in wall height, which is compatible with the results shown in [1], [19], [20] and [21]. On the other hand, the restraint became bigger with increased wall length, as shown in figure 6, up to about 10m. Thereafter the restraint is no longer increasing with increased wall length.

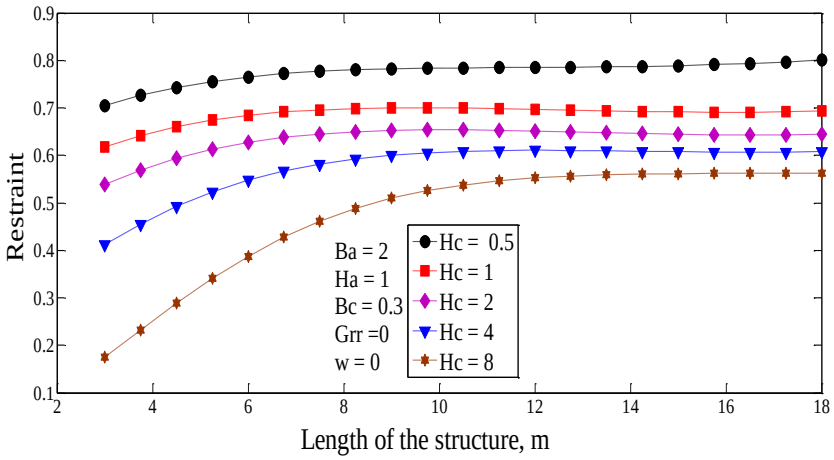


Figure 6 - Variation of restraint with length and wall height as predicted by ANN model at height 0.1 H_c.

6.2 Effect of external rotational restraint (γ_{rr})

As shown in table 2, the second parameter of influence on restraint is the external rotational restraint. The bending moment during a contraction in a wall rotates the ends of the structure upward and the center downward. If the material under the foundation is stiff, the resistance on the structure is high, which at total rotational stiff ground reflects by γ_{rr} equal to 1. If the material under the foundations is very soft, the value of γ_{rr} is zero. The results of the ANN with γ_{rr} =1 showed high restraint, which is in line with results in [22]. The restraint is about 40% lower when γ_{rr} is equal to zero. For both γ_{rr} = 1 and γ_{rr} = 0 the restraint increases with length of the structures up to about 10m (for $L/H_c \leq 5$), see figure 7.

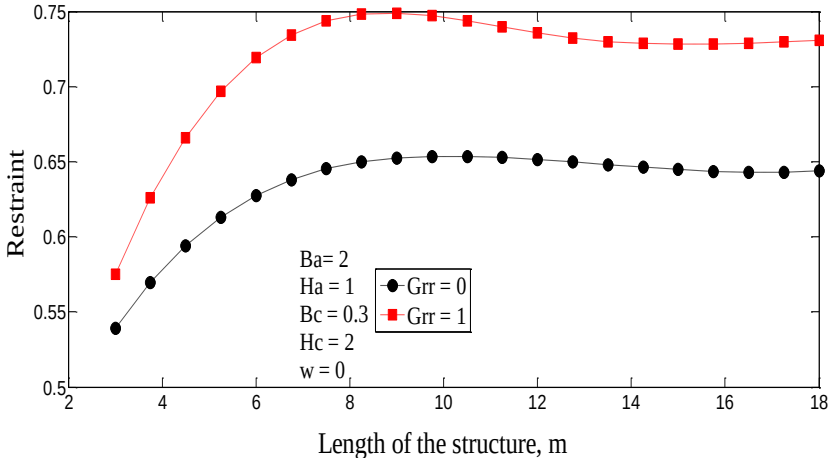


Figure 7 - Variation of the restraint with structural length and external rotational restraint as predicted by ANN model at height 0.1 H_c.

6.3 Effect of wall thickness (B_c)

Increase of the size of the new concrete means higher possibility of counteracting the external constraints (from old concrete, i.e the slab in this case), which is reflected in figure 8 as the restraint will decrease with increased wall thickness. Besides, up to a structural length of 10m (for $L/H_c \leq 5$) the restraint increases with increased structure length, which is in agreement with results in [4], [23] and [24].

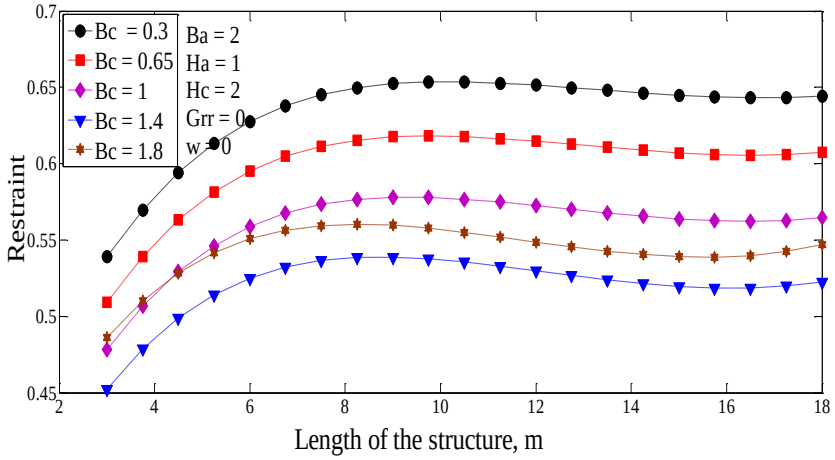


Figure 8 - Variation of the restraint with structural length and wall width as predicted by ANN model at height 0.1 H_c.

6.4 Effect of the relative position of the wall on the slab (ω)

When a wall is placed in the middle of the slab ($\omega = 0$), it has the highest restraint from the slab, and other more eccentric positions become successively less and less restraint as shown in figure 9. An increase of the length of the structures results in increased restraint up to the length of about 10m (for $L/H_c \leq 5$).

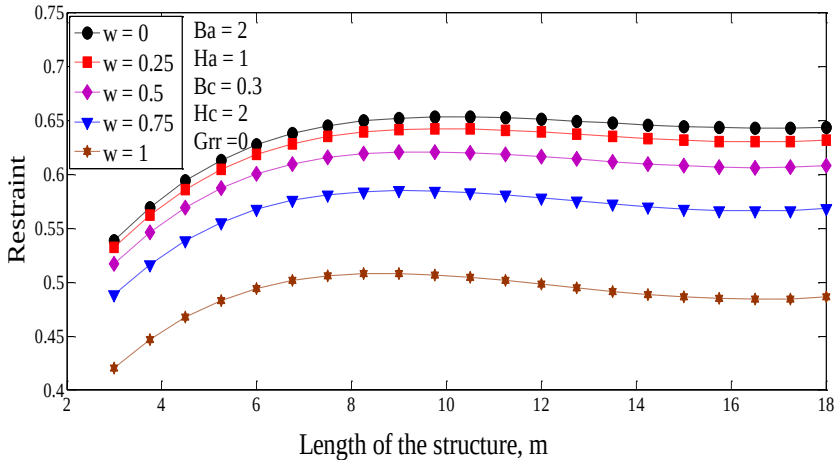


Figure 9 - Variation of the restraint with structural length and wall position on slab as predicted by ANN model at height 0.1 H_c.

6.5 Effect of slab thickness (H_a)

An increase in slab thickness (H_a) results in an increased value of restraint. The effect of increasing the length of the wall is also increasing the value of restraint up to a length of about 10m (for $L/H_c \leq 5$), see figure 10. This is in agreement with findings in [4], [20], and [23].

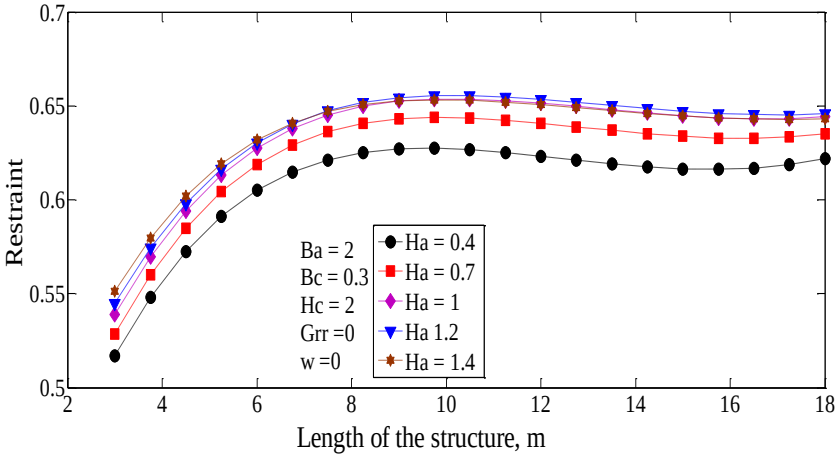


Figure 10 - Variation of the restraint with length and slab thickness as predicted by ANN model at height 0.1 H_c.

6.6 Effect of slab width (B_a)

Generally, the value of restraint increases with the increase of the slab width for all levels of the wall height. A smaller increase in the value of restraint is observed with the increase in

structural length beyond 10m (for $L/H_c > 5$), see figure 11. The same indication is found in [4], [21], [25], and [26].

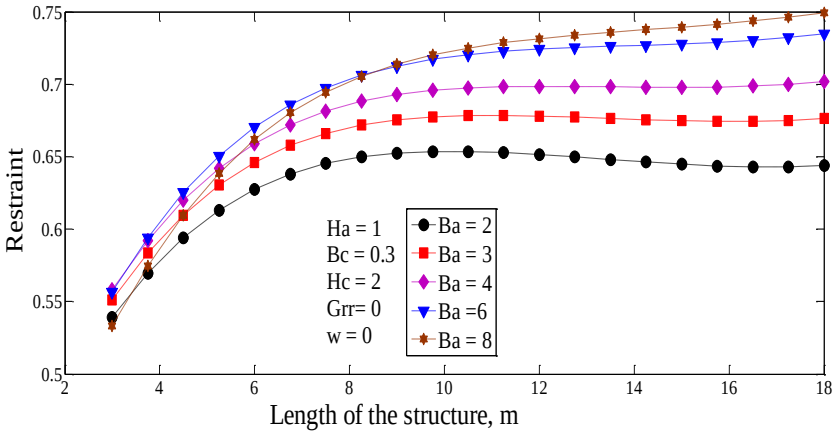


Figure 11 - Variation of the restraint with length and slab width as predicted by ANN model at height 0.1 H_c.

7. ANN MODEL DEVELOPMENTS FOR RESTRAINT PREDICTION

The ANN model is used to derive a design formula to calculate the restraint by using multi-layer perceptions (MLP) for training the model with the back-propagation training algorithm. The model has seven inputs representing the width of the slab (B_a), the height of a slab (H_a), the width of the wall (B_c), the height of the wall (H_c) , the length of the structure (L), the rotational boundary restraint (γ_{rr}), and the relative location of the wall on the slab (ω). All the parameters and their values are listed in table 1.

The structure of the optimal ANN model is shown in figure 12, while its connection weights and threshold levels are summarized in Appendix A, tables A1-A11.

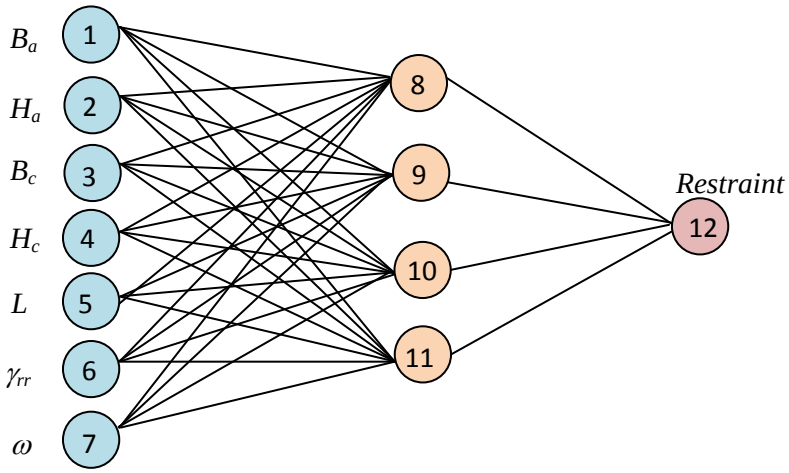


Figure 12 - Structure of the optimal ANN.

7.1 The design formula

The equation length depends on the number of nodes in the hidden layer. To shorten the length of the equation, an adoption of the number of nodes by four is introduced with a correctness of 95%. An adoption of 17 nodes gives an accuracy of 99%. The small number of connection weights of the neural network enables the ANN model to be translated into a relatively simple formula, in which the predicted restraint can be expressed as follows:

$$\gamma_R = \frac{1}{1+e^{-\left(\theta_{12}+\left(w_{8:12}\frac{1}{1+e^{-(x_1)}}\right)+\left(w_{9:12}\frac{1}{1+e^{-(x_2)}}\right)+\left(w_{10:12}\frac{1}{1+e^{-(x_3)}}\right)+\left(w_{11:12}\frac{1}{1+e^{-(x_4)}}\right)\right)}}$$

where (7)

$$X_1 = \theta_8 + (w_{8:1})\cdot(B_a) + (w_{8:2})\cdot(H_a) + (w_{8:3})\cdot(B_c) + (w_{8:4})\cdot(H_a) + (w_{8:5})\cdot(L) + (w_{8:6})\cdot(\gamma_{rr}) + (w_{8:7})\cdot(\omega) \quad (8)$$

$$X_2 = \theta_9 + (w_{9:1})\cdot(B_a) + (w_{9:2})\cdot(H_a) + (w_{9:3})\cdot(B_c) + (w_{9:4})\cdot(H_a) + (w_{9:5})\cdot(L) + (w_{9:6})\cdot(\gamma_{rr}) + (w_{9:7})\cdot(\omega) \quad (9)$$

$$X_3 = \theta_{10} + (w_{10:1})\cdot(B_a) + (w_{10:2})\cdot(H_a) + (w_{10:3})\cdot(B_c) + (w_{10:4})\cdot(H_a) + (w_{10:5})\cdot(L) + (w_{10:6})\cdot(\gamma_{rr}) + (w_{10:7})\cdot(\omega) \quad (10)$$

$$X_4 = \theta_{11} + (w_{11:1})\cdot(B_a) + (w_{11:2})\cdot(H_a) + (w_{11:3})\cdot(B_c) + (w_{11:4})\cdot(H_a) + (w_{11:5})\cdot(L) + (w_{11:6})\cdot(\gamma_{rr}) + (w_{11:7})\cdot(\omega) \quad (11)$$

It should be noted that, before using Eqs. 7, 8, 9, 10, and 11, that all input variables need to be scaled between 0.1 and 0.9 using Eq. 1 for the data ranges in table 1. It should also be noted that predicted restraint obtained from Eq. 7 is scaled between 0.1 and 0.9 and in order to obtain the actual value, this restraint has to be re-un-scaled using Eq. 1. ANN should be used only for interpolation and not extrapolation [13].

7.2 Numerical example

A numerical example is provided to present the implementation of the restraint formula. Input parameters are: B_a =2m, H_a = 0.4m, B_c = 0.3m, H_c = 4m, L= 18m, γ_{rr} = 1, and ω = 0. As shown in figure 13, the convergence in results from finite-element (FE) calculations [1] and results using the Excel spread sheet is very good. Therefore, the Excel spread sheet can be used as a substitute for fast and accurate calculation of restraints in the wall.

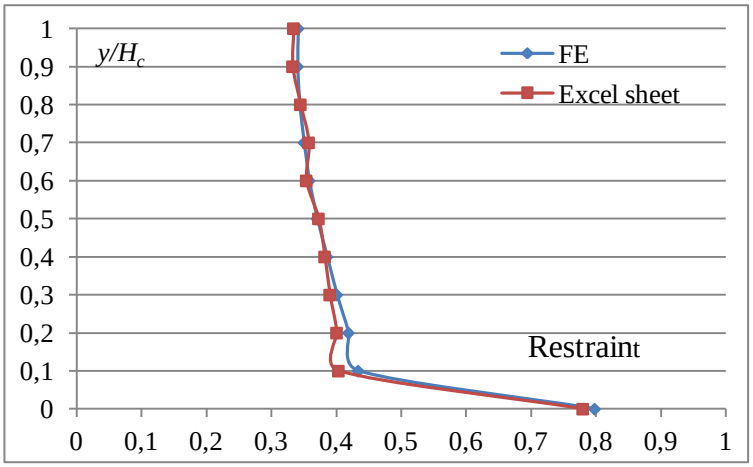


Figure 13- Comparison between finite-element restraints and results using the Excel spread sheet.

8. CONCLUSIONS

Existing research concerning restraint curves has been applied to the method of artificial neural networks to model restraint in the wall for the typical structure wall-on-slab. Seven input parameters have been used, and it has been proven that the neural network is capable of modelling the restraint with good accuracy.

The usage of the neural network has been demonstrated to give a clear picture of the relative importance of the input parameters. The dimension of the wall (height and width) as well as the external rotational restraint turned out to give the highest importance on restraint in the wall. On the opposite, the width of the slab was found to be of least significance in this respect.

Further, it is shown that the results from the neural network can be represented by a series of basic weight and response functions. Resulting functions can easily be implemented to simple computer tools. Thus, the results can easily be made available to any engineer without use of complicated software.

REFERENCES

1. Nilsson, M., "Restraint Factors and Partial Coefficients for Crack Risk Analyses of Early Age Concrete Structures", Lulea, Sweden, Division of Structural Engineering, Lulea University of Technology, Doctoral Thesis 2003:19, 170 pp.
2. Al-Gburi, M., Jonasson, J.E., Nilsson, M., Hedlund, H., Hösthagen, A., "Simplified Methods for Crack Risk Analyses of Early Age Concrete Part 1: Development of Equivalent Restraint Method", aim to be published in the *Nordic Concrete Research* publication as part 1 in connection to the present paper.
3. Emborg, M., & Bernander, S., "Assessment of the Risk of Thermal Cracking in Hardening Concrete", *Journal of Structural Engineering (ASCE*), Vol.120, No 10, October 1994, pp. 2893-2912.
4. ACI Committee 207, "Effect of Restraint, Volume Change, and Reinforcement on Cracking of Massive Concrete", ACI Committee 207.ACI207.2R-1995. Reapproved 2002, 26 pp.

5. JSCE, "English Version of Standard Specification for Concrete Structures 2007", *Japan Society of Civil Engineer*, JSCE, December 2010, 503 pp.
6. Cusson, D., & Hoogeveen, T., "An Experimental Approach for the Analysis of Early Age Behavior of High Performance Concrete Structures Under Restrained Shrinkage", *Cement and Concrete Research*, 2007, 37: 2, pp. 200-209.
7. Bamforth, P. B., "Early-age Thermal Crack Control in Concrete", CIRIA Report C660, Construction Industry Research and Information Association, London, 2007.
8. Olofsson, J., Bosnjak, D., Kanstad, T., "Crack Control of Hardening Concrete Structures Verification of Three Steps Engineering Methods", 2000, 13th Nordic Seminar on Computational Mechanics, Oslo.
9. Weiss, W. J., Yang, W., Shah, S. P., "Influence of Specimen Size/Geometry on Shrinkage Cracking of Rings", *Journal of Engineering Mechanics*, Vol. 126, No. 1, January, 2000, pp. 93-101.
10. Moon, J.H., Rajabipour, F., Pease, B., Weiss, J., "Quantifying the Influence of Specimen Geometry on the Results of the Restrained Ring Test", *Journal of ASTM International*, Vol. 3, No. 8, 2006, pp. 1-14.
11. Hossain, A.B., & Weiss, J., "The Role of Specimen Geometry and Boundary Conditions on Stress Development and Cracking in the Restrained Ring Test", *Cement and Concrete Research*, 36, 2006, pp. 189– 199.
12. Yousif, S. T., & Al-Jurmaa, M. A., "Modeling of Ultimate Load for R.C. Beams Strengthened with Carbon FRP using Artificial Neural Networks", *Al-Rafidain Engineering*, Vol.18, No.6, December 2010, pp. 28-41.
13. Shahin, M.A., Jaksa, M.B, Maier, H.R., "Artificial Neural Network−Based Settlement Prediction Formula for Shallow Foundations on Granular Soils", *Australian Geomechanics* September 2002, pp. 45-52.
14. Yousif, S. T., "Artificial Neural Network Modeling of Elasto-Plastic Plates", Ph.D. thesis, College of Engineering, Mosul University, Iraq, 2007, 198 pp.
15. Hudson B., Hagan, M., Demuth, H., "Neural Network Toolbox for Use with MATLAB", User's Guide, the Math works, 2012.
16. Hagan, M.T., Demuth, H.B., Beale, M.H., "Neural Network Design", Boston, MA: PWS Publishing, 1996.
17. Garson, G.D., "Interpreting Neural Network Connection Weights", *Artificial Intelligence,* Vol. 6, 1991, pp. 47-51.
18. Goh, A.T.C., "Back-Propagation Neural Networks for Modeling Complex Systems", *Artificial Intelligence in Engineering*, Vol.9, No.3, 1995, pp. 143-151.
19. Emborg M., "Thermal Stresses in Concrete Structures at Early Ages", Div. of Structural Engineering, Lulea University of Technology, Doctoral Thesis, 73D, 1989, 280 pp.
20. Kheder, G. F, Al-Rawi, R. S., Al-Dhahi, J. K., "A Study of the Behavior of Volume Change Cracking in Base Restrained Concrete Walls", *Materials and Structures*, 27, 1994, pp. 383-392.
21. Kheder, G.F., "A New Look at the Control of Volume Change Cracking of Base Restrained Concrete Walls", *ACI Structural. Journal*, 94 (3), 1997, pp. 262-271.
22. Lin, F., Song, X., Gu, X., Peng, B., Yang, L., "Cracking Analysis of Massive Concrete Walls with Cracking Control Techniques", *Construction and Building Materials*, 31, 2012, pp. 12–21.
23. Kim S.C., "Effects of a Lift Height on the Thermal Cracking in Wall Structures", *KCI Concrete Journal* (Vo1.12 No.1), 2000, pp. 47-56.
24. Larson M., "Evaluation of Restraint from Adjoining Structures", IPACS-Rep, Lulea University of Technology, Lulea, Sweden, 1999.

25. Nagy A. “Parameter Study of Thermal Cracking in HPC and NPC Structures”, *Nordic Concrete Research*, No.26, 2001/1.
26. Kwak, H.G., & Ha, S.J., “Non-Structural Cracking in RC Walls: Part II. Quantitative Prediction Model”, *Cement and Concrete Research* 36, 2006, pp. 761–775.

Appendix A

Weights and threshold levels for the ANN-model

Table A1: Weights and threshold levels for the ANN- model at 0.0 H_c

Hidden layer nodes	w_{ji} (weight from node at hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	-2.44	-0.133	1.92	1.56	1.75	-0.33	0.962	3.968
I=9	0.0448	0.4451	-3.90	0.805	0.386	-1.33	-8.569	-2.429
I=10	0.1464	4.28	-2.27	-1.08	-1.099	-2.75	-2.29	2.505
I=11	1.742	0.099	-1.800	-1.172	-0.813	0.4078	-0.738	5.5439
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output Threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	0.455	10.73	6.24	-0.4758				-4.88

Table A2: Weights and threshold levels for the ANN- model at 0.1 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.0228	0.21462	0.3629	-18.48	0.9613	1.122	0.212	-2.271
I=9	0.422025	-0.06176	1.98924	-0.6882	0.30437	-0.26	-0.417	-1.2636
I=10	0.456249	0.983905	0.64158	0.13169	0.3093	-0.68	-0.168	-0.3123
I=11	0.238852	0.70202	-0.7221	0.60450	0.18516	-0.41	0.0307	-0.334
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output Threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	9.014351	-12.8866	15.6198	-21.768				5.60494

Table A3: Weights and threshold levels for the ANN - model at 0.2 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.023445	0.21469	0.361232	-18.5183	0.96787	1.12751	0.21163	-2.24292
I=9	-0.42725	-0.2689	0.31187	-0.70018	2.02201	-0.06425	0.431046	-1.28285
I=10	-0.1732	-0.7011	0.317028	0.129956	0.66889	1.008782	0.469116	-0.32013
I=11	0.03241	-0.4146	0.18831	0.61946	-0.7446	0.71167	0.24299	-0.33843
output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	8.7721	-12.329	14.5275	-20.611				5.428395

Table A4: Weights and threshold levels for the ANN- model at 0.3 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.02282	0.2146	0.36293	-18.4809	0.96137	1.122909	0.212814	-2.27106
I=9	-0.41754	-0.2642	0.30437	-0.68824	1.98924	-0.06176	0.42202	-1.26363
I=10	-0.16823	-0.6808	0.30936	0.13169	0.64158	0.98390	0.45624	-0.31232
I=11	0.030785	-0.4095	0.185169	0.604506	-0.7221	0.707202	0.23885	-0.33428
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	9.014351	-12.886	15.61978	-21.7685				5.60494

Table A5: Weights and threshold levels for the ANN - model at 0.4 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	2.08677	1.89426	-13.088	-1.1256	-0.8407	41.17545	-32.701	-6.50761
I=9	0.377159	0.839513	-0.18151	-20.125	-0.3396	1.031952	-0.6816	1.04641
I=10	0.562324	0.737864	-0.90535	-1.2830	-0.6673	0.518905	-0.4199	-1.09078
I=11	0.579971	0.730745	-0.7001	-0.1999	-2.6483	0.734031	-0.4312	-2.35647
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	0.419494	2.097828	9.638326	-18.157				-1.666

Table A6: Weights and threshold levels for the ANN - model at 0.5 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	-1.45824	-2.32771	0.962872	-8.541	17.20482	22.46012	0.418163	-19.6627
I=9	0.756076	1.319794	-1.30934	-15.31	1.05319	0.74600	-0.42114	0.208142
I=10	-0.76824	0.25679	-0.64332	-8.505	9.60673	-4.05398	-0.69526	1.556513
I=11	1.10672	1.1844	-1.2012	0.3616	-2.586	-1.3277	-1.460	4.105541
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	1.38328	2.76767	0.91604	2.4217				-4.05196

Table A7: Weights and threshold levels for the ANN -model at 0.6 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.62026	0.47351	0.08976	5.18154	-3.78167	0.3101	-0.35249	1.50859
I=9	0.164556	1.471751	0.243077	23.6184	-24.3285	-32.39	1.23612	27.6750
I=10	-0.3054	-0.06287	-0.51847	-5.7015	3.504167	-0.065	0.054154	-1.11537
I=11	0.3387	1.058383	-0.44985	-27.486	0.912928	-0.366	-0.07223	1.450579
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	10.75089	-0.92821	11.19475	2.29642				-11.3610

Table A8: Weights and threshold levels for the ANN -model at 0.7 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.42400	0.46559	-0.1034	5.27938	-4.29611	0.05693	-0.223	1.94446
I=9	0.20791	0.18420	0.20273	5.76087	-4.2182	-0.5358	-0.031	1.80288
I=10	0.65041	2.65318	-1.61066	-38.5765	1.52342	-7.6225	0.2018	3.22180
I=11	0.111222	0.213763	-0.64653	-38.1758	2.497382	9.53468	-0.973	-4.43226
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	14.2932	-14.2761	1.782392	1.422751				-1.32971

Table A9: Weights and threshold levels for the ANN -model at 0.8 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.482716	1.208346	-0.4909	5.799786	-4.74126	2.507117	-0.529	0.186043
I=9	0.715349	0.563623	0.37424	38.31441	-19.5478	-1.05955	-0.063	-1.96305
I=10	0.757585	1.53234	-1.3743	-17.3411	2.203233	-3.26635	-0.051	-0.26653
I=11	0.579219	0.284736	-1.1736	-11.4707	4.159462	7.178042	-0.853	-6.87323
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	2.574084	-1.41622	4.10681	2.846912				-2.155

Table A10: Weights and threshold levels for the ANN- model at 0.9 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden Threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.23413	0.39783	-0.1258	4.48423	-2.73151	0.31317	-0.11	0.838376
I=9	-0.72798	-2.68067	1.4830	30.1924	-0.56191	4.97414	-0.445	-2.30485
I=10	0.117852	0.00236	-0.2931	-7.23425	1.47341	0.96875	-0.165	0.709421
I=11	-0.01498	-0.12181	-0.1528	-5.9498	2.101039	0.556965	-0.072	-0.16348
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							
	I=8	I=9	I=10	I=11	-	-	-	Output Threshold (θ_j)
I=12	19.49644	-1.96336	-16.522	36.35604				-18.3594

Table A11: Weights and threshold levels for the ANN- model at 1 H_c

Hidden layer nodes	w_{ji} (weight from node hidden layer i in the input layer to node j in the hidden layer)							Hidden layer threshold (θ_j)
	I=1	I=2	I=3	I=4	I=5	I=6	I=7	
I=8	0.10182	0.66933	-9.45E-	4.53729	-3.77808	3.23152	-0.0659	0.35271
I=9	0.206039	0.070818	0.26018	21.02211	-6.83813	-0.35579	0.14037	-2.78455
I=10	-0.10211	0.271457	-0.1774	-26.613	4.154443	0.310213	-0.0372	5.312146
I=11	0.422038	0.611063	-0.7748	-8.54489	1.897913	-0.74841	-0.1359	-1.07541
Output layer nodes	w_{ji} (weight from node i in the hidden layer to node j in the output layer)							Output threshold (θ_j)
	I=8	I=9	I=10	I=11	-	-	-	
I=12	4.860535	-4.40516	-3.2183	8.698041				-1.1289