Modified Intelligent Energy Management system in a smart house[1]

Dariush Shahgoshtasbi
Electrical Engineering Department
University of Texas at San Antonio
San Antonio, TX, USA
isjd@wacong.org

Mo Jamshidi
Electrical Engineering Department
University of Texas at San Antonio
San Antonio, TX, USA
moj@wacong.org

***Abstract*— Demand response has an important role in improving energy efficiency. By using it, we are able to shift electrical load from peak demand time to other periods which is usually in response to price signal. In residential level and in a dynamic pricing system which modification of energy consumption is unrecognized by a consumer, using an automated Energy Management System (EMS) should be considered. In this paper, which is the modified version of our last work, an intelligent EMS in a smart house is presented[1]. It has two components, fuzzy component and intelligent lookup table. Fuzzy component is in the EMS and makes the proper output for intelligent lookup table based on its fuzzy rules and inputs. The second component which its core is an associative neural network is moved to smart appliances. So each appliance has a separate intelligent lookup table. They are able to map inputs to desired outputs. They take three types of inputs which come from fuzzy component, outside sensors and other smart appliances. So, all the appliances should send their control bits to each other. Whatever is trained in these lookup tables are different scenarios in different conditions. This system is able to find the best energy efficiency scenario in different situations.**

Keywords- Energy Efficiency, Demand Response, Smart Grid, Fuzzy logic, Neural Networks.

I. INTRODUCTION

Smart Grid is a novel initiative which its aim is to deliver energy to the users and also to achieve consumption efficiency by means of bidirectional communication [2]. Combination of different hardware devices and software along with an ICT infrastructure for a bidirectional communication make the smart grid architecture. ICT has a vital rule in the smart grid architecture as it gives sustainability, creativity and intelligence to it. This electricity network is able to intelligently integrate the actions of all users which are connected to it in order to return them back to users. Users can use this information to optimize their energy consumption. Thus, one of the main objectives of smart grid is encouraging end users to participate in making decision about energy consumption in an efficient way. But in order to reach energy efficiency, such architecture is not enough. We need to add intelligence to it in different levels. In home level, the approach is to add intelligence and then encourage customers to save energy by changing their energy consumption behavior. So, demand response can be in response to an economical signal which is mostly a pricing signal. By using demand response, we are able to shift electrical load from peak demand time to other periods which reduces the ratio of peak to average load. This can be resulted in improving efficiency, reducing costs and risk of outages. Demand response can be done on different levels like generation, transmission or end user level. A lot of works has been done on generation and transmission levels [2][3][5][6][7][8]. In the end user level, the smart grid is not only able to provide information about electricity consumption for both users and network operators, but also is able to connect renewable energy resources to the grid. In the residential or end user level demand response, challenges include real time pricing information to consumer, networking home devices, security and implementing automated Energy Management Systems (EMS) should be considered. In this paper, which is the modified version of our last work [1], an intelligent EMS system is presented. It takes inputs from the grid and by using an intelligent algorithm, tries to find effective and efficient energy consumption. It also can be matched by users' preferences and behaviors and then find optimal energy scheduling according to the dynamic notion of price. This approach is useful especially in a dynamic pricing system

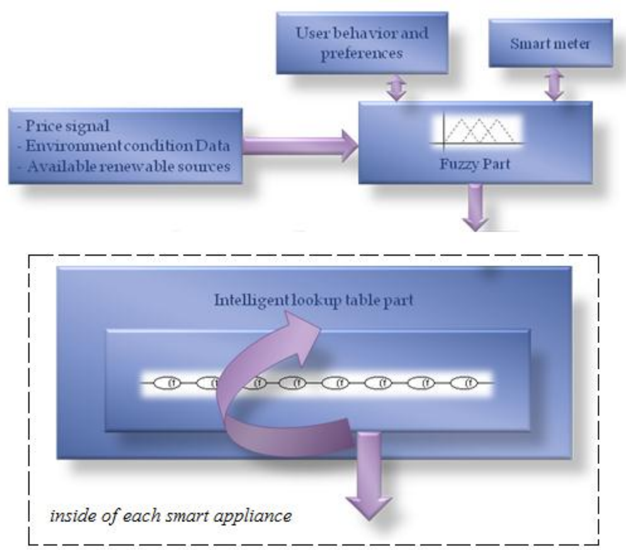

Figure 1. Modified automated EMS

[1] This work was supported, in part, by a grant from CPS Energy through Texas Sustainable Energy Research Institute and Lutcher Brown Chair, the University of Texas, San Antonio, TX, USA.

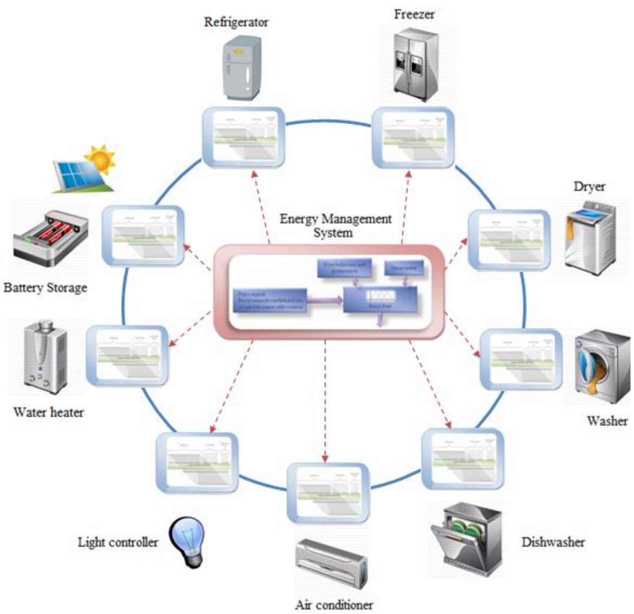

Figure 2. Smart appliances connection.

which modification of energy consumption is unrecognized by and renewable resources or can be human behavior and a consumer. The suggested system has two components as it is shown in Figure 1. The first component is a fuzzy system which its inputs can be external events like price signal, environment condition data preferences. This component which is inside the EMS has some fuzzy rules along with their membership functions which make appropriate output for the second component which is intelligent lookup table. The reason of using Fuzzy component is to get real time changeable external information and find the best system situation based on the fuzzy rules and then by using centralized defuzzification get a unique number in the output range. The second and the main component is intelligent lookup table which is moved to smart appliances. So each appliance has a separate intelligent lookup table. The reason of doing that is to have a very simple structure of neural network which will be resulted in fewer scenarios in each appliance. In this situation all of the appliances should send some control bits to each other. Therefore, they should be connected to each other as it is shown in Figure 2. The Intelligent lookup table is made by using a new topology of neural network [4]. The output part of this topology is modified. By using this modification which will be explained in section III, we are able to connect more inputs to the neural network. The first reason of using this type of neural network is that its structure is crystal type and can be easily extended for more inputs and the second reason is that it is able to act as an associative memory to map inputs to desired outputs. This paper is organized as follows. In Section II, the fuzzy component of the system is explained. In section III, the structure of the intelligent lookup table, its training and functionality are described and in section IV, the simulation of the suggested approach is shown.

II. Fuzzy Logic Component

This component has 5 types of inputs. The first one is price signal which comes from the smart meter, the second one is battery storage situation, and the rest are environment condition data like humidity, temperature and solar as their membership functions are depicted in Figure 3. These inputs are divided into two categories. Price signal, battery storage and solar inputs are in the first category which makes a unique and appropriate input for the intelligent lookup table. The output of this category has 7 membership functions which gives us the energy consumption situation in the gird. These 7 fuzzy sets are Neutral-, Neutral+, Normal, High-, High+, Peak- and Peak+ which are selected based on the fuzzy rules and mapping of input values to their appropriate membership functions. Finally the resulting set is defuzzified by Centriod defuzzification technique and gets a single number as (1).

$$x^* = \frac{\int \mu_i(x)\, x\, dx}{\int \mu_i(x)\, dx} \tag{1}$$

Where x^* is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and x is the output variable. This number is rounded and its binary value goes to the intelligent lookup table. The second category of this component which contains environment condition data makes some input control signals to the second component which will be explained in the next section. The fuzzy rules and membership function values are set based on our knowledge from the grid and the environment. These values can be changed by human behavior and preferences. The behavior and preferences are logged and if there are any changes in them (like A/C setting), the system just modifies the initial membership values.

III. Intelligent Lookup Table Component

A. Structure

The main part of this system has a crystal structure with cells as shown in Figure 4-a. As it can be seen, each cell is constructed of two middle neurons and four side neurons such that the two upper ones have fixed weights and the two lower ones have trainable weights. The two upper neurons and also the middle neurons have linear function and the two lower ones have a sigmoid function as (2).

$$y = \frac{1 + e^{-\lambda x}}{1 - e^{-\lambda x}} \tag{2}$$

The neurons of this layer receive their binary inputs from the previous layer. Figure 4-a can be summarized as Figure 4-b. For each cell we have:

$$f(n) = f(n-1) \times \left(f(input_n \times W_p) + f(\overline{input}_n \times W_q) \right), \;\; f(0) = c \tag{3}$$

Where n is the input cell number. In Figure 4-a, if $input_n$, which is the complement of $\overline{input}_n$, gets value 1, the first and the forth neuron will fire whereas if it gets 0, the second and third neurons do so. In the training step, only active neurons are trained. The characteristic of the lower neurons is that the increasing and decreasing rate of the weight changes are not equivalent during the training step as will be explain in section III.B. In each cell, there are only two active neurons at a time and the output of that cell enters as input to the succeeding cell. If each input consists of n bits, cells would be connected as a chain up to the $(n-1)th$ input. The only remaining input is n where the output of the chained string is connected to $\frac{2^n}{2}$ of this

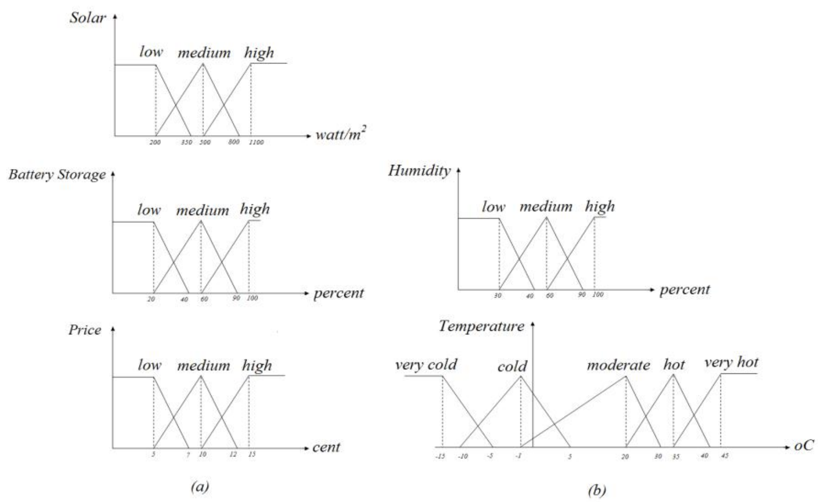

Figure 3. Fuzzy membership functions for the fuzzy component. (a): Category 1. (b): Category 2.

input cell. We call this layer the parallel layer as its cells are parallel to each other. In this layer, only one cell of input n is selected. With this selection, we reach our desired memory cell. For networks with more than one output, $\frac{2^n}{2}$ of input cell n are added to the parallel layer for each output. These added cells are only used for adjustment of all outputs except the first one. They have no influence on the adjustment of other weights. On the other word, the weights of the network are adjusted only by the first output and the other outputs are adjusted by the weights of added input cells in the parallel layer [4]. Figure 5 shows this structure for an example network with 3 inputs and 2 outputs. The problem of using this type of network is when we have a lot of inputs, the number of cells in parallel layer increase dramatically. On the other hand, we increase the capacity of associative memory, but most of the cells in parallel layer will be never used. In order to solve the problem, instead of having $\frac{2^n}{2}$ cell in parallel layer, we consider m cells and by using a mapping table, each used output is mapped to one of these m considered cells. In order to make an intelligent lookup table by the suggested neural network, we create feedbacks from outputs of the network to its inputs as it is depicted in Figure 6. This makes a sequential system with memory. In this situation, next outputs will be made based on the present inputs and feedback outputs. As it was mentioned earlier, the system has three types of inputs; the first type comes directly from the fuzzy component and is a unique value which shows energy consumption situation in the grid. This input is connected to n first inputs of neural networks and implies the network which scenario should be started. The second type of inputs is some control bits which come either from fuzzy component or some outside sensors and the last type are feedback outputs. All of

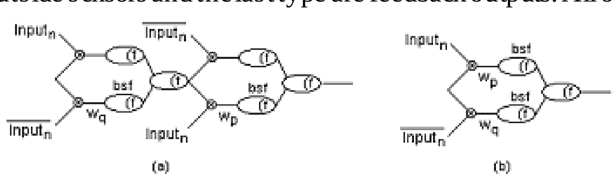

Figure 4. One cell of the network in the associative memory layer (a): Original Shape , (b): The summary of part a.

the inputs together based on their conditions make different scenarios for the intelligent lookup table. Control bits can show for example water heater, air conditioner, refrigerator and other appliances situation in the smart house. Also they can be battery storage and solar situation which come from the fuzzy component. This intelligent lookup table should be trained based on the different inputs and control bits.

B. Learning

The algorithm of network training is a way to decode the problem to be solved [9]. The learning of this network is supervised. The single layer perceptron learning rule is used to train this network [10]:

$$W_j(t+1) = W_j(t) + \eta \times (Train_{out} - out) \times In_p \tag{4}$$

Where W is the weight, j is the input cell number, In_p is the input of that cell, out is the output of the network and $Train_{out}$ is the desired output. The important point is that using the learning rule of single layer perceptron for this multi-layer network makes learning easy. The way of using this rule is presented here. As mentioned earlier, only active neurons are trained and in each cell there is only one lower active neuron at a time. So the Eq. (4) can be rewritten as:

$$W_j(t+1) = W_j(t) + SelectNeuron \times \eta \times (Train_{out} - out) \times In_p \tag{5}$$

In each layer, the η values which have been used for the stabilization of weights are smaller when weights increase than when they decrease. This always makes weights decreasing values larger than weights increasing values. These increasing and decreasing values are reduced in subsequent layers. The reason is that in binary numbers, the bit toggle frequency is halved when we move from LSB bits toward MSB bits. Since the inputs of the suggested network are binary, considering η as mentioned, will cause the changes of weights to become slower in subsequent layers. On the other words, weights change with lower frequency for 2^n sequential

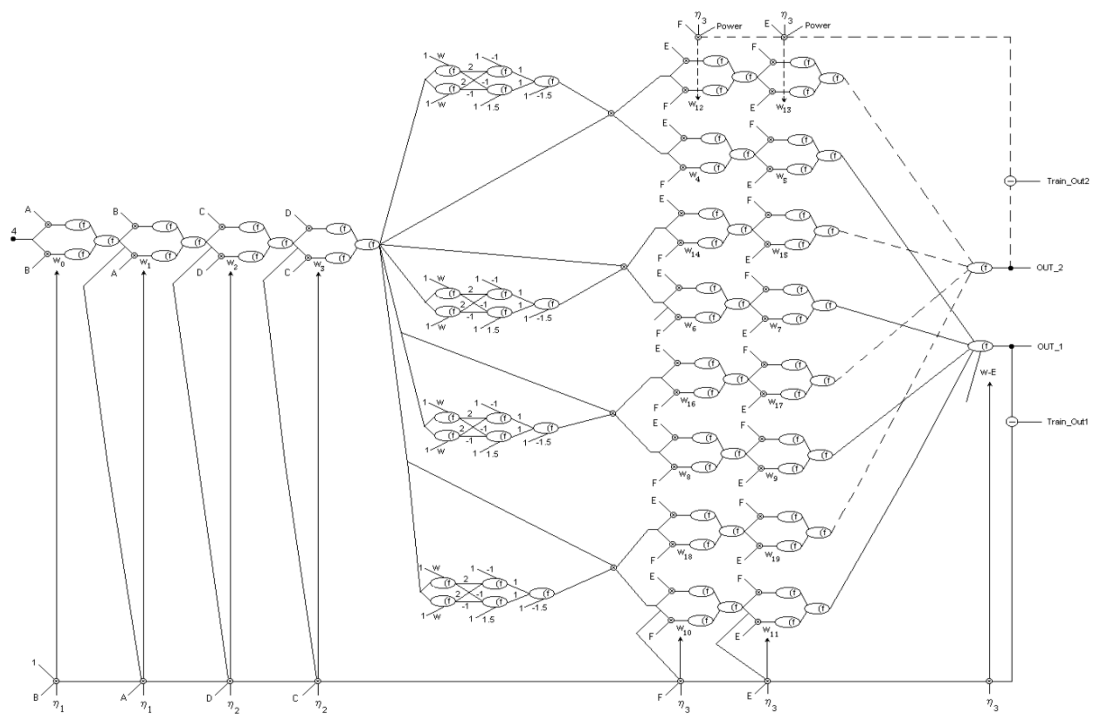

Figure 5. The structure of the associative memory layer with 3 inputs and 2 outputs. A,C,E are inputs and B,D,F are their complements.

binary inputs. Considering the first layer as MSB and moving toward parallel layer as moving to LSB, the weight toggles would be less for the neurons of input cell *n* which are in the last layer and frequency of changes in their inputs is more than other cells. Due to the fact that the output error creates feedbacks to all layers, the weights of all active neurons will decrease or increase if this error becomes negative or positive respectively. But since η values are not equal, their increasing and decreasing rate would be different in each layer. By this method the network can be stabilized and get adjusted to 2^n input forms which result in the convergence of weights [4]. Experiments show that in case of having more than one output, it is better to consider a coefficient *Power* to adjust the weights of all outputs except the first one to expedite the convergence of weights.

C. Functionality

This network acts as an associative memory and is able to map a set of inputs to a desired output. Whatever will be trained in the network are different scenarios (based on different conditions) and connection between them. It can be trained in a certain time and mostly maps outputs based on the previous learning. The system will be trained when we have a new scenario or we intend to change one. This characteristic of the suggested lookup table gives the system speed and reliability which are two important items in the field of energy in a smart house. Scenarios in the mentioned intelligent lookup table are connected to each other in a way which suggest the best energy efficiency for appliances in the smart house. In fact, this system predicts outputs based on control bits and fuzzy inputs. For example, if a smart house has both battery storage and solar system and weather forecast information shows that we would have a cloudy weather in the next hours (which solar would not work), if the control bit is set and imply the battery storage needs to be charged, the system charges it in order to use its energy in the future peak hours. As it was mentioned before, each smart appliance has a separate intelligent lookup table. The first step is that all the appliances

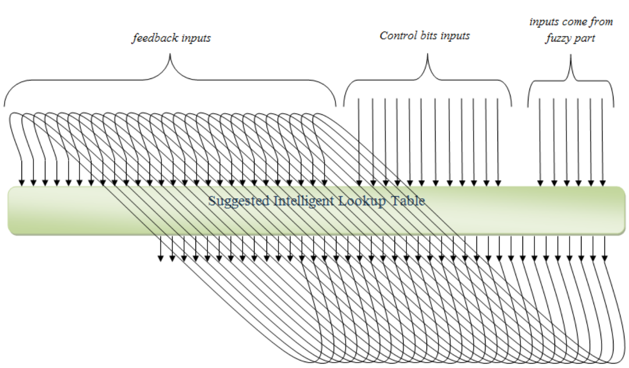

Figure 6. The intelligent lookup table structure.

Algorithm for Smart Appliances

```
The first highest priority node sends control bit information to the second one
The second one adds its control bit information to it and sends them to the third one and so on
The last one adds its control bit information and sends all the control bit information to the first one
do {
    fuzzy_output_change ← checkfuzzyoutput()
}while (fuzzy_output_change != True)
do {
    if (fuzzy_output_change == True) then
        findbestscenariobasedoninputs()
    fuzzy_output_change ← checkfuzzyoutput()
    a_node_control_bit_information_changes ← controlbitchanges()
    if (a_node_control_bit_information_changes == True) then
        sendcontrolbittoothers ()
}while (True)
```

Figure 7. Algorithm for Smart Appliances.

should send their control bits to each other. We consider priority for our appliances. The priority can be in the plug which the appliances is connected to. The highest priority appliance sends its control bit information to the second one. It is clear that if the second one is turned off or is not plugged in, the third one will be considered as the second one. The second one adds its control bit information to it and sends them to the third one. This process continues to reach the last one. The last one adds its control bit information and sends all the control bit information to the first one. Therefore, all the appliances have all the control bits. After that all the intelligent lookup tables are waiting for inputs from Fuzzy component. When they receive it, each intelligent lookup table tries to find the best scenario for its appliances based on the inputs and whatever they have been trained. This algorithm is shown in Figure 7. We just have *n* Scenarios which the connections between them in different situations are trained in the network.

IV. Simulation

For simulation of the mentioned system, we consider a smart house with water heater, air conditioner, light, solar panel, battery storage, refrigerator, freezer, dishwasher, washer and dryer. The fuzzy component of the simulation has 5 types of inputs along with membership functions as they are shown in Figure 3. In the first category, we have 15 fuzzy rules which make proper inputs for the intelligent lookup table. These fuzzy rules are shown in Figure 8. An associative memory with 38 inputs and 18 outputs is designed. Totally 4096 cells are considered for its parallel layer. The outputs of the fuzzy component enter the first six inputs of the network. We consider 12 control bits which enter the network as next inputs. These control bits are for solar, battery storage and home appliances situations. The rest inputs come from feedback outputs. Seventeen different scenarios are defined and based on different conditions; proper scenarios are trained to each neural network. These scenarios are shown in Table 1. The weight changes are shown in Figure 9. As shown in this figure, the weights stabilize after 25 samples which shows that this kind of connection between neurons results in weight convergence to the desired output. The figure also shows that the weight changes in cells toward parallel layer are reduced which indicates the stabilization of the network. Figure 10 and 11 show the weight changes for two lower trainable weights in parallel layer for all of the training situations. They show that weight changes are limited and also for all of the outputs are very close to each other. For the simulation result, the mentioned house is simulated in GridLAB-D software and the results of real power with and without using the intelligent EMS during one day (24 hours) are compared. As it is shown in Figure 12, by applying the intelligent EMS in the house, the energy consumption is moved from peak hour to non-peak hour by the system.

TABLE I. Seventeen defined scenarios for energy efficiency in a smart house

No.	Scenario
1	Charge battery by grid
2	Charge battery by solar
3	Turn on water heater
4	Turn on air conditioner
5	Turn on dishwasher if programmed on
6	Turn on washer if programmed on
7	Turn on dryer if programmed on
8	Give portion of energy to dishwasher if programmed on
9	Give portion of energy to washer if programmed on
10	Give portion of energy to dyer if programmed on
11	Give portion of energy to refrigerator if needed (on)
12	Give portion of energy to freezer if needed (on)
13	Turn off dishwasher
14	Turn off washer
15	Turn off dryer
16	Get energy from battery storage
17	Turn on light controller

Algorithm for Energy Management System

1: **If** (cost is low) then output ← neutral_minus
2: **If** (cost is meduim and battery is high and solar is high) **then** output ← neutral_plus
3: **If**(cost is meduim and battery is high and solar is meduim) **then** output ← neutral_plus
4: **If**(cost is meduim and battery is high and solar is low) **then** output ← normal
5: **If**(cost is meduim and battery is meduim and solar is high) **then** output ← normal
6: **If**(cost is meduim and battery is meduim and solar is meduim) **then** output ← normal
7: **If**(cost is meduim and battery is meduim and solar is low) **then** output ← high_minus
8: **If**(cost is meduim and battery is low and solar is high) **then** output ← high_minus
9: **If**(cost is meduim and battery is low and solar is meduim) **then** output ← high_plus
10: **If**(cost is meduim and battery is low and solar is low) **then** output ← high_plus
11: **If**(cost is high and battery is high) **then** output ← peak_minus
12: **If**(cost is high and battery is meduim and solar is high) **then** output ← peak_minus
13: **If**(cost is high and battery is meduim and solar is meduim) **then** output ← peak_minus
14: **If**(cost is high and battery is meduim and solar is low) **then** output ← peak_plus
15: **If**(cost is high and battery is low) **then** output ← peak_plus

Figure 8. Fuzzy rules for the intelligent Energy Management System

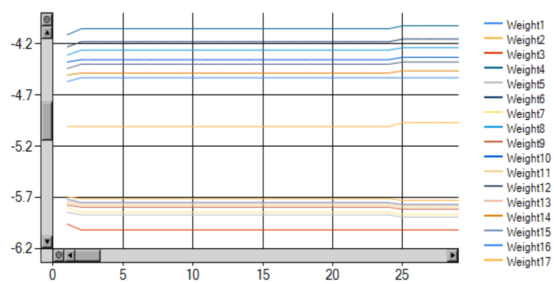

Figure 9. The value of weight changes for simulation of system with 38 inputs and 18 outputs.

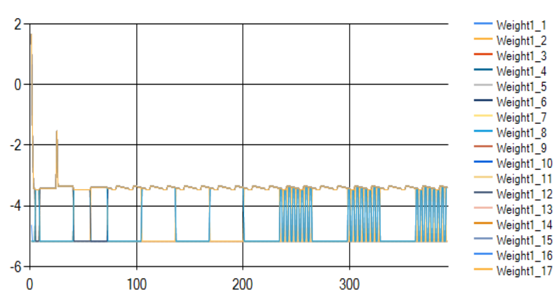

Figure 10. Weight changes for the first lower trainable weight in parallel layer for all of the training situations.

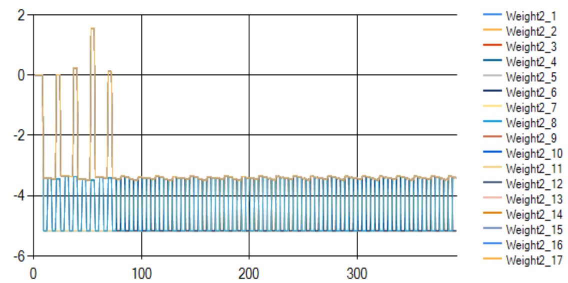

Figure 11. Weight changes for the second lower trainable weight in parallel layer for all of the training situations.

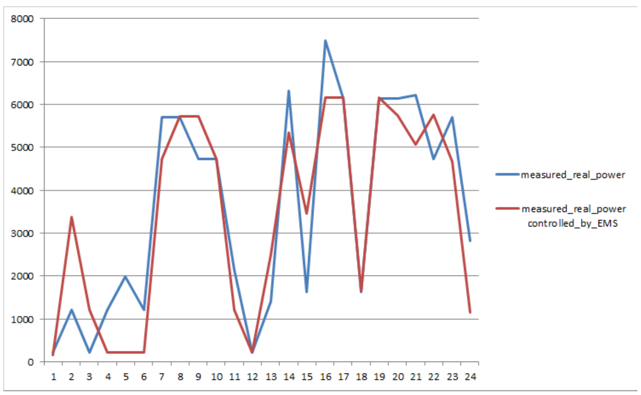

Figure 12. Comparing measured power with and without applying the intelligent EMS to the system during one day (24 hours).

V. Conclusions

The modified version of our previous work as an automated Energy management system (EMS) for residential level is presented. The suggested system has two components, fuzzy component and intelligent lookup table. The fuzzy component which is in the EMS has 15 fuzzy rules along with membership functions which makes appropriate outputs for the intelligent lookup table component. The intelligent lookup table is an associative neural network which is moved to smart appliances. Therefore, each appliance has a separate intelligent lookup table. They map inputs to desired outputs. They can be trained based on the different scenarios and connections between them. This method makes the structure of neural networks simpler which is resulted in faster convergence of weights. The intelligent lookup table has three types of inputs. The first type comes directly from the fuzzy component. The second type is control bits which come either from the fuzzy component or some outside sensors and the last type of inputs is feedbacks from its outputs. This automated energy management system is able to find the best energy efficiency scenario in different situations.

References

[1] D. Shahgoshtasbi, Mo Jamshidi, "Energy efficiency in a smart house with an intelligent Neuro-Fuzzy lookup table", The 6th IEEE international conference on System of Systems Engineering (SOSE 2011), Albuquerque, NM, USA, 2011.

[2] Omid Siakani, O. Hussian, T. Dillon, A. Tabesh, "Intelligent Decision Support System for Including Consumers' Preferences in Residential Energy Consumption in Smart Grid", Second International Conference on Computational Intelligence, Modelling and Simulation, 2010.

[3] T. R. Kuhn, "Energizing Efficiency's Potential," The Electricity Journal, vol. 19, pp. 83-87, 2006.

[4] D. Shahgoshtasbi, " A New Topology of Neural Network for Memory Simulation", World Automation Congress, Seville, Spain, 2004.

[5] R. Earle, E. P. Kahn, and E. Macan, "Measuring the Capacity Impacts of Demand Response," The Electricity Journal, vol. 22, pp. 47-58, 2009.

[6] O. Sezgen, C. A. Goldman, and P. Krishnarao, "Option value of electricity demand response," Energy, vol. 32, pp. 108-119, 2007.

[7] J. Sancho, J. Sánchez-Soriano, J. A. Chazarra, and J. Aparicio, "Design and implementation of a decision support system for competitive electricity markets," Decision Support Systems, vol. 44, pp. 765-784, 2008.

[8] D. Coll-Mayor, M. Paget, and E. Lightner, "Future intelligent power grids: Analysis of the vision in the European Union and the United States," Energy Policy, vol. 35, pp. 2453-2465, 2007.

[9] Cornelius T. Leondes, "Algorithms and architectures", Academic Press, 1998.

[10] R. J. Schalkoff, "Artificial Neural Network", McGraw-Hill, International, 1997.