A new intelligent Neuro-Fuzzy paradigm for Energy Efficient Homes

Dariush Shahgoshtasbi, *Student Member*, Mo Jamshidi, *Fellow, IEEE*

***Abstract*— Demand response, which is the action voluntarily taken by a consumer to adjust amount or timing of its energy consumption, has an important role in improving energy efficiency. With demand response, we can shift electrical load from peak demand time to other periods based on changes in price signal. At residential level, automated Energy Management System (EMS) have been developed to assist users in responding to price changes in dynamic pricing systems. In this paper, a new intelligent EMS (*i*EMS) in a smart house is presented. It consists of two parts: fuzzy subsystem and intelligent lookup table. Fuzzy subsystem is based on its fuzzy rules and inputs which produces the proper output for intelligent lookup table. The second part, whose core is a new model of an associative neural network, is able to map inputs to desired outputs. The structure of the associative neural network is presented and discussed. The intelligent lookup table takes three types of inputs which come from fuzzy subsystem, outside sensors and feedback outputs. Whatever is trained in this lookup table are different scenarios in different conditions. This system is able to find the best energy efficiency scenario in different situations.**

***Index Terms*— Energy Efficiency, Fuzzy logic, Demand Response, Neural Networks, Smart Grid**

I. Introduction

SMART grid is a novel initiative whose aim is to deliver energy to the users and also to achieve consumption efficiency by means of bidirectional communication [1]. Combination of different hardware devices and software along with an Information and Communication Technology (ICT) infrastructure for a bidirectional communication constitutes 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 and interoperability is not enough. We need to add intelligence to it at different levels. At home level, the approach is to add intelligence and then encourage customers to save energy by changing their energy consumption behavior.

The electrical grid has two main sides: supply which contains generation, transmission and distribution and demand which consumes the power. The balance between supply and demand sides is necessary at all times, otherwise some blackouts will occur. Throughout the course of a day, when demand increases, the utility companies have to turn on some reserve generation capacity and send the power to the grid to respond to the additional demand. Because these generators usually use gas or diesel to run, they are very expensive. They also emit more CO_2 compared to nuclear and hydro power plants, but less CO_2 compared to coal-fired power plants used for base load supply. If demand increases and there is not enough capacity, the utilities pay customers to shed load, usually during times of peak load or an emergency situation. Blackouts are the worst case scenario, occurring when demand exceedingly increases and the load cannot be handled. Peak demand traditionally has been a problem in supply-side management, solved by the construction of new power plants. Focusing on management of demand-side can be an alternative way to balance energy use at peak times. 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 [2] and risk of outages. Demand response can be done at different levels like generation, transmission or end user level. A lot of work has been done at generation and transmission levels [1,3-7]. At 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 can dispatch renewable energy resources to the grid. At the residential or end user level demand response, challenges that should be considered include real time pricing information to consumer, networking home devices, security and implementing automated EMS. In this paper, a new intelligent EMS system, called *i*EMS is presented. It takes inputs from the grid and by using an intelligent algorithm, tries to find effective and efficient energy consumption. It can also match users' preferences and behaviors and then find optimal energy scheduling according to the dynamic price. This approach is useful especially in a dynamic pricing system which modification of energy consumption is unrecognized by a

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.

Dariush Shahgoshtasbi is a research assistant in ACE lab at University of Texas at San Antonio, TX 78249 USA (e-mail: isjd@wacong.org).

Mo Jamshidi is with the Electrical Engineering Department, University of Texas at San Antonio, TX 78249 USA (e-mail: moj@wacong.org).

A preliminary version of this paper appeared at the 6th IEEE international conference on System of Systems Engineering (SOSE), Albuquerque, NM 2011

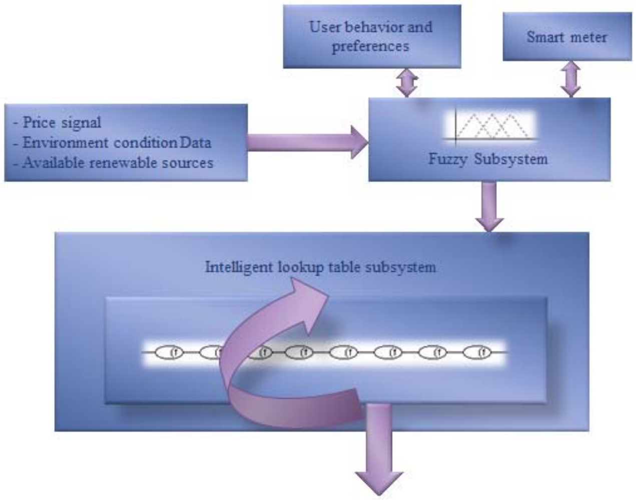

Fig. 1. The proposed iEMS

consumer. The suggested system has two subsystems as it is shown in Figure 1.

The first subsystem, is a fuzzy system in which the inputs can be external variable like price signal, environment condition data, renewable resources or even human behavior and preferences. This subsystem has some fuzzy rules along with membership functions that make it an appropriate output for the second subsystem, which is an intelligent lookup table. The function of this fuzzy subsystem is to obtain the best system state based on external real time data. The output variable is obtained through centroid defuzzification which is one of the most common techniques of converting fuzzy output into a numerical output. The user behaviors and preferences are stored in log files and applied in the fuzzy subsystem. Clearly, different areas of the country have different climates and some regions may use some appliances, such as air conditioner, more than other regions. Also, user preferences vary in different areas and by considering all of them, the system should perform efficiently in all situations. So, user behaviors and preferences, weather information and price data in each area, shown in separate blocks, are logged and, after a while, the values of membership functions in this subsystem are set to proper values in order to obtain the maximum efficiency. By using this method, if the home owner changes, the system is able to adjust itself to new owner behaviors after a while.

The second, and most important subsystem, is an intelligent lookup table that consists of using a new topology of neural network [8]. The first reason for using this type of neural network is that its structure is crystal type which can be easily extended for more inputs. Crystal type means this topology of neural networks is made by the same shape cells. Another reason to use this type of neural network, is that it is capable of acting as an associative memory that maps inputs to desired outputs.

This paper is organized as follows. In section II, the related works are reviewed. In section III, the fuzzy subsystem of the system is explained. In section IV, the structure of the intelligent lookup table, its training and functionality are described and in section V, the simulation of the suggested approach is shown. Acknowledgements and Conclusions are made in sections VI and VII, respectively.

II. RELATED WORK

Reliability and affordability of electricity supply have historically been the main objectives of improving the electrical grid. Governments have invested in the generation and transmission infrastructure of the grid in certain nations such as USA, Canada and European countries. They were built in anticipation of customer demand growth. In the early days of infrastructure, the cost of utility declined as plants became more efficient and larger. However, in 1960's power generation cost steadily increased due to increases in fuel cost, environmental protection, decreases in technological advances, and problems with nuclear power projects [9]. A demand increase and price increases for electricity caused a global energy crisis in the 1970's. At the time, defenders of conservation argued that working on reduction of demand-side would be better and cheaper than to increase on the supply-side.

Cost reduction, supply reliability increase, market efficiency, customer service improvement, environment sustainability improvement and market power mitigation are benefits of using demand response [10]. Peak demand has been a problem on the supply-side as it requires construction of new power plants to supply the demand. Demand-side management is an alternative way to balance peak time loads. A lot of work has been done in residential level in order to make the energy consumption efficient.

In the field of energy management, Rausch and Palensky [11] use an intelligent global energy management program called PROFESY. PROFESY is used to manage energy consumption at peak hours by loading and monitoring each customer profile. The consumer profiles are controlled locally by a device called the Maximum Demand Monitor (MDM). PROFESY which is a Java written program combines and optimizes the local MDMs. PROFESY also implements load prediction algorithm through the use of neural networks and also relying on a database to globally coordinate the customer's local MDMs.

In [12] LeMay, *et al.* have combined Building Automation System (BAS) and Advanced Meter Infrastructure (AMI) technologies to make the Meter Gateway Architecture (MGA) implement Demand Side Management (DSM) functionalities. MGA serves as a foundation for integrated control loads by energy aggregators, unified hubs and smart appliances. The main control component is the unified hub which, by using ZigBee protocol, receives utility price signal in real time by the smart meter. The signals can be sent to smart appliances to control themselves or for the hub to control the demand response.

Molderink, *et al.* [13] present an overview of current research to improve energy efficiency of electricity supply. They discussed Domestic technologies (Distributed Generation, Energy Storage, and Demand Side Load management) with different control methodologies (Local scope, Microgrid and Virtual Power Plant). Their presented residential DSM has three steps: a local offline part which predicts energy demand and production in each single house by using neural network; a global offline controller which gathers the individual load profiles and generates a global planning; a local real time controller which is an online scheduler and decides which appliances are switched on/off, etc.

Sianaki, *et al.* [1] present an approach which decreases the energy consumption in a smart house by using an intelligent decision-making model. Dynamic pricing, user behavior and renewable sources of energy are important factors for this system to work. They present an intelligent approach in which demand response can be achieved on a continuous basis at home level.

In [14] Lui, *et al.* describe smart grid Demand Response (DR) system architecture developed by Whirlpool Smart Device Network (WSDN), which includes both the Home Area Network and the Smart Grid domains that has three levels of DR operation. In the lowest level, a smart appliance individually responds to a smart grid control or pricing signal. In the second level, a home energy management system coordinates the responses from all smart devices (solar panel, electrical vehicle supply equipment, micro generation, appliances). The third level coordinates the responses from hundreds to millions of houses through the internet (as the networks build the smart meter domain do not have sufficient bandwidth yet). Consumers should control how appliances respond to the DR signals. The architecture of the network and the home energy management system functional blocks and its security are discussed. A centralized Smart Device Controller which also operates as gateway towards the Smart Grid manages appliances.

Conejo, *et al.* in [15] by using bidirectional communication with the electricity supplier, describe an optimization model which can be easily integrated in the energy management system of a house. This model adjusts the hourly load level in response to hourly electricity price and minimizes the energy cost. They suppose the price and decision for the initial t-1 hours are clear. Also the price for the current hour, 10 min prior to the current hour and also consumer demand at the beginning of this hour are known. The prices for the following 24-t hours are unknown and are considered via robust optimization. The basic energy consumption levels for the following 24-t are variables to be determined.

In [16] Kishore, *et al.* propose mechanisms to optimize power consumption both in a home with smart appliances and across multiple such homes in neighborhood. It is supposed that appliances communicate with each other and also with EMS over HAN network. At first, a simple optimization model is used to determine the optimal timing of appliance operation in a single home by considering price and user behavior. Then they show that if multiple homes each optimize their appliances, the problem of demand peak is not solved and just shifted simply to the off-peak period. Then they propose a distributed scheduling mechanism to reduce peak demand within neighborhood homes. At the end, they propose an EMS optimization model based on dynamic programming which is more realistic than the first one.

In [17] Han, *et al.* present an energy management system based on Zigbee protocol. The various sensor networks and device controllers in order to save energy consumption in a smart house is used. A new smart house control system based on sensor networks in order to make home networks more intelligent is presented.

O'Neill, *et al.* [18] present a new algorithm named CAES for energy management system in the residential level. It controls residential device usage by using an online application called Q-learning which estimates the impact of future energy prices and consumer decisions based on current energy decisions. It uses Markov chain in order to model both residential device usage and energy prices but with unknown transition probability distribution. Related work to this paper can also be seen in [1, 11, 14]. Like these works, user's behavior, dynamic pricing and renewable resources are also considered in this work. The difference between this work and the state of the art is using the intelligent paradigm and the new topology of neural networks which makes this system act as a lookup table.

III. Fuzzy Logic Subsystem

This subsystem consists of five inputs. The first is price signal which comes from the smart meter, the second is battery storage state, and the rest are environment condition data like humidity, temperature and solar. Their membership functions are depicted in Figure 2 and are separated into two categories (a) and (b). Price, battery storage and solar radiation inputs (a) provide unique and appropriate input for the intelligent lookup table. Seven membership functions have been considered for the output of this engine which gives us the energy consumption situation in the gird. These seven fuzzy sets are Neutral-, Neutral+, Normal, High-, High+, Peak- and Peak+. Finally, the resulting set is defuzzified by centroid defuzzification technique to obtain a numerical value in (1).

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

Where x^* is the defuzzified (numerical) output, $\mu_i(x)$ is the i^{th} output membership function and x is the output variable. This number is rounded to a whole number whose binary value goes to the intelligent lookup table. The second category creates some input control signals to the second subsystem 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 desired values can be modified by human preferences which can be monitored for any changes, such as changes in A/C setting. These preferences are used to initialize membership function values.

IV. Intelligent Lookup Table Subsystem

A. *Structure*

The main element of this system has a crystal structure with cells as shown in Figure 3-a. As seen, each cell is constructed from two middle neurons and four side neurons. The two upper neurons have fixed weights and along with the middle neurons, they all have linear functions. The lower two neurons are different in the sense that they have trainable weights and are part of the sigmoid function as seen on (2).

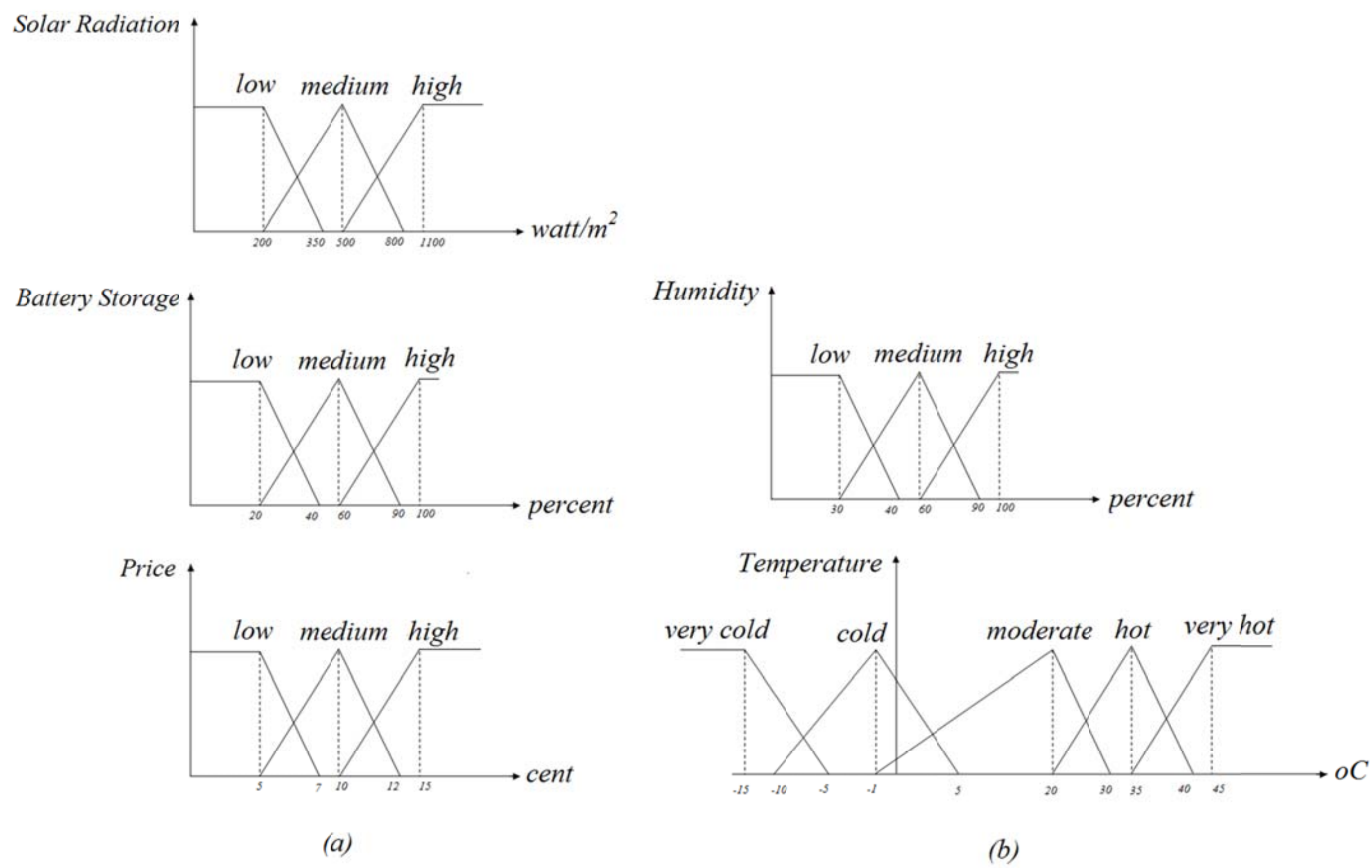

Fig. 2. Fuzzy membership functions for the fuzzy subsystem. (a): Category 1. (b): Category 2.

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

The neurons of this layer receive their binary inputs from the previous layer. Figure 3-a can be summarized as Figure 3-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 3-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 value 0, the second and third neurons fire. During 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 at the training step, this will be further explained in section IV-*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,

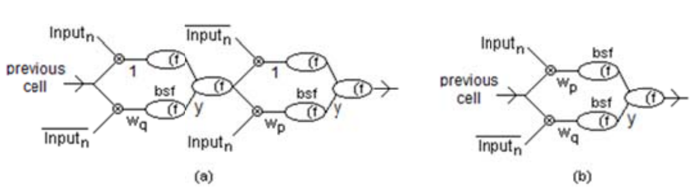

Fig. 3. One cell of the network in the associative memory layer (a): Original Shape , (b): The summary of part a

cells would be connected as a chain up to the $(n-1)th$ input. The only remaining input is n^{th}, where the output of the chained string is connected to $\frac{2^n}{2}$ of this 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 and we then 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 not an influence on the adjustment of other weights. In the other words, 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 [8]. Figure 4 shows this structure for an example network with 3 inputs and 2 outputs. A shortcoming of using this type of network is that the number of cells in parallel layer increase dramatically as the number of inputs increase. On the other hand, as we increase the capacity of an associative memory, most of the cells in parallel layer will never be used. 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 into one of these m cells [19]. Feedback from the proposed neural network is used to make intelligent lookup table, as depicted in Figure 5. A sequential system with memory is created in this configuration in which next outputs depends on present inputs and feedback.

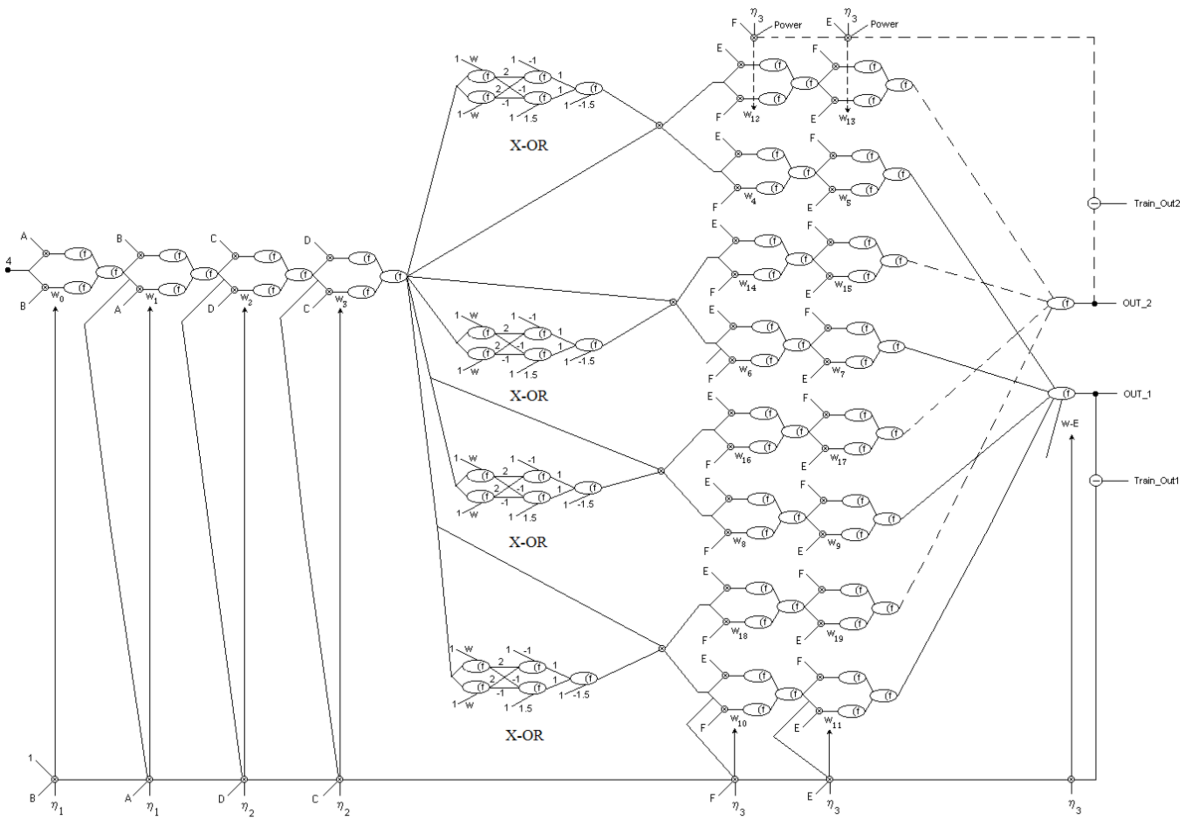

Fig. 4. The structure of the associative memory layer with 3 inputs and 2 outputs. A,C,E are inputs and B,D,F are their complements.

As it was mentioned earlier, the system has three types of inputs. The first type of input is output of the fuzzy subsystem which represents energy consumption in the grid as a continuous variable. This input is connected to *n* first inputs of neural networks and implies the network which scenario should be started. The second input type is that of discrete-valued control bits from fuzzy subsystem and external sensors and the last input type are feedback outputs. All inputs, based on their conditions, make different scenarios for the intelligent lookup table [20]. Control bits can show for example, water heater, air conditioner, refrigerator and other appliances state in the smart house, or battery storage and solar radiation from the fuzzy subsystem. This intelligent lookup table should be trained based on different inputs and control bits.

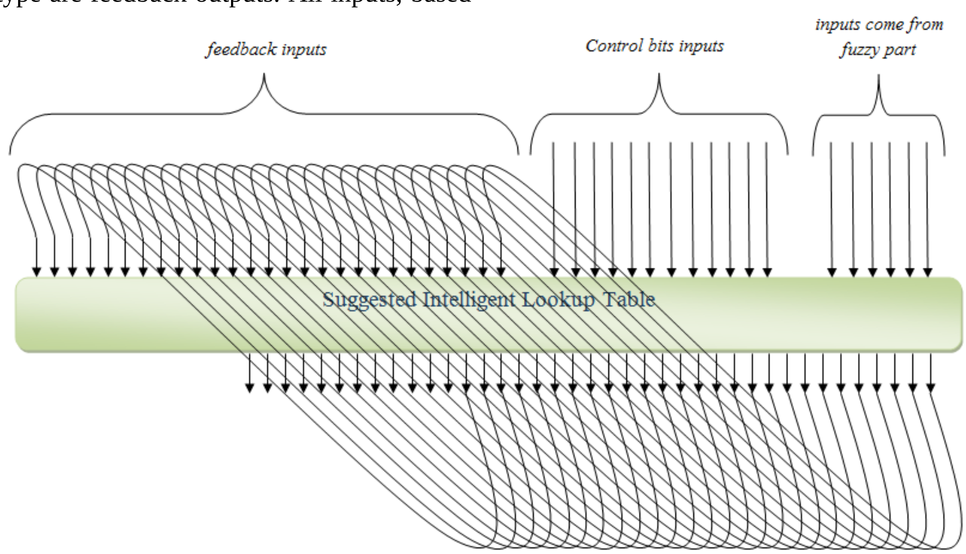

Fig. 5. The intelligent lookup table structure.

B. Learning

The algorithm of network training is to decode the problem to be solved [21]. The learning of this network is supervised. The single layer perceptron learning rule is used to train this network [22]:

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

Where W_j is the *jth* cell's weight, *j* is the input cell number, μ is the learning rate, 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 method 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 Eq. (4) can be rewritten as:

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

In each layer the μ values, which have been used for the stabilization of weights, are smaller when weights increase than when they decrease. This always makes weights decreasing increments larger than weights increasing increments. These increasing and decreasing increments are reduced in subsequent layers. The reason is that in binary numbers, the bit toggle frequency is halved when we move from Least Significant Bit (LSB) bits toward Most Significant Bit (MSB) bits. Since the inputs of the suggested network are binary, considering μ as mentioned, the changes of weights will become slower in subsequent layers. In other words, weights change with lower frequency for 2^n sequential binary inputs. With the first layer as MSB and the 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. Frequency of changes in their inputs is more than other cells. With output error feedback to all layers, weights of all active neurons will increase or decrease if the error is positive or negative, respectively. As μ values are not equal, their increasing and decreasing rate will be different in each layer. By this method, the network can be stabilized and get adjusted to 2^n input forms which result in convergence of weights [8]. 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. Inputs

Input networks topologies can vary based on their desired usage. These networks only send the signature of inputs as *n* binary bits to the memory network. For example, with classification problems we can use some simple neural networks as shown in Figure 6. In this figure, to set the output values of input network, a coordinate plane is divided to several areas according to the needed number of output bits. In Figure 6 an example is shown for 4 outputs which consists of a coordinate plane and two sets of parallel lines. The first two inputs vary in value across the X and Y axis of the coordinate plane. The second two inputs vary over the sets of parallel lines (one set per input). Orientation of the parallel lines can either be perpendicular or oblique to the coordinate frame. Neuron outputs are trained such that their output is 1 if the input is between parallel lines, otherwise it is zero. By this method we can find a unique code for each area on the coordinate plane and guess the approximate coordinate for the input points. By using this protocol, it is obvious that the memory network for a 3 dimensional coordinate plane and a 2 dimensional coordinate plane with eight areas, as is shown in Figure 7, are the same because both of them have 3 inputs. Also we can use this method for nonlinear patterns as it is shown in Figure 8.

Input patterns to neural networks may contain noise due to distortion which is important to consider. Associative memory networks, however, have fault tolerance which helps the

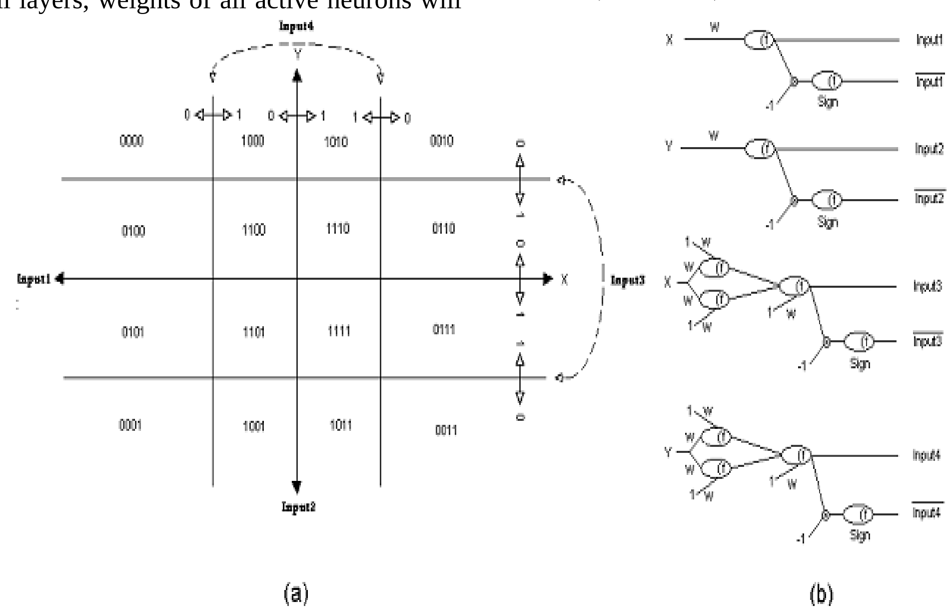

Fig. 6. (a): Dividing the coordinate plane for 4 outputs (4 inputs for memory network). (b): The suggested input neural networks to make binary inputs for memory network according to part (a) and defined protocol.

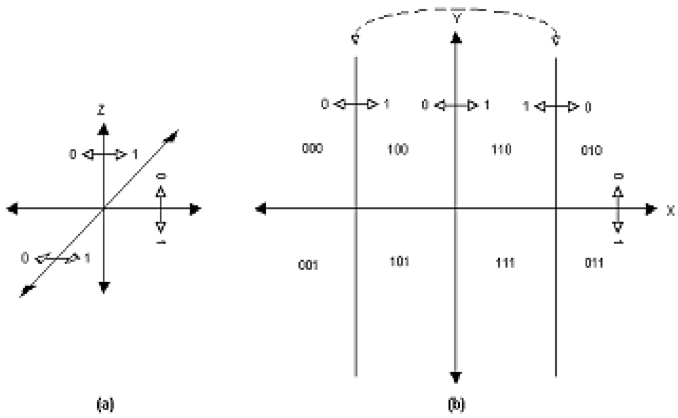

Fig. 7. (a): Dividing the 3 dimensional coordinate plane (b): Dividing the 2 dimensional coordinate plane with 3 outputs (from input network)

network to reach a desired output in spite of some input distortion. We declare the suggested associative memory only adjusts outputs by binary inputs which are made by input neural networks. So the input neural networks must prepare suitable outputs for the associative memory network even for noisy inputs. It should be noted that in our previous example, of a classification problems, our system uses unique codes to cover a suitable portion of the coordinate plane. Any input that falls into a region is assigned to a unique code. Therefore, we handle a particular level of noise. It is obvious that if there are more inputs in the network, the capacity of the memory network increases. But because it causes more dividing of coordinate plane, each area becomes smaller, so noise sensitivity of the input networks also increases. If input networks require more fault tolerance, other associative memories such as Hopfield, Bidirectional Associative Memory (BAM) and etc. can be used. As it was mentioned before, the intelligent lookup table takes three types of inputs which come from fuzzy subsystem, outside sensors and feedback outputs.

A. Proof of Convergence

As mentioned earlier, the learning rate μ values in different layers, can be written as:

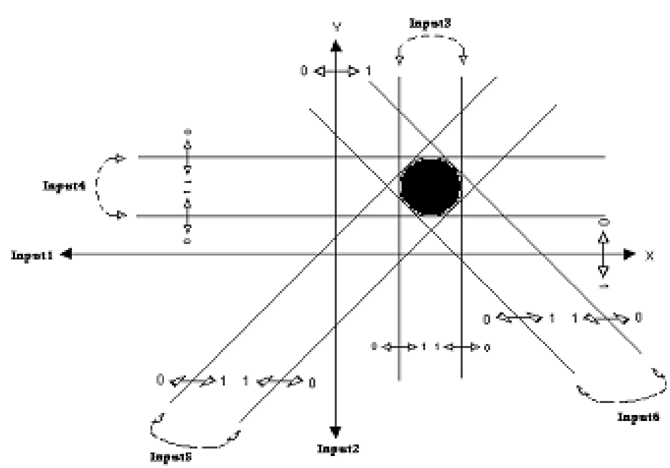

Fig. 8. Dividing the coordinate plane for nonlinear pattern

$$\mu_{\text{first\_layer}} > \mu_{\text{second\_layer}} > \mu_{\text{third\_layer}} > \cdots \qquad (6)$$

In the meanwhile the adjustment of the weights can be written as:

$$W_{\text{new}} = W_{\text{old}} + \text{SelectNeuron} \times \mu \times \text{input} \times (\text{Train}_{\text{out}} - \text{Out}) \qquad (7)$$

where:

$$\Delta W = \text{SelectNeuron} \times \mu \times \text{input} \times (\text{Train}_{\text{out}} - \text{Out}) \qquad (8)$$

and *SelectNeuron* shows the active neuron and *input* is the input from previous layer to that active neuron. To simplify, each cell in each layer is considered as Figure 3-b and by using Eq. (2) for output rate in each layer we may write:

$$\begin{array}{c} f_1(\text{input}_1 \times W_1) \\ f_2(\text{input}_2 \times W_2) \\ f_3(\text{input}_3 \times W_3) \\ \vdots \end{array} \qquad (9)$$

Where *fn(Inputn × Wn)* is output of cell *n*, and *Inputn* is input of cell n and *Wn* is the weight of the active neuron in layer *n* for n=1,2,…

As the middle neuron of each cell has linear function with slope 1, so it may be written:

$$\begin{array}{l} \text{Input}_1 = \text{Const.} = \text{In} \\ \text{Input}_2 = f_1(\text{In} \times W_1) \\ \text{Input}_3 = f_2(f_1(\text{In} \times W_1) \times W_2) \\ \vdots \end{array} \qquad (10)$$

Also the initial weights are considered as below:

$$\text{Initial } W_1 > \text{Initial } W_2 > \text{Initial } W_3 > \cdots \qquad (11)$$

Now, according to the constant value of *In* which is greater than 1 and the values of functions *f'* which are always between 1 and -1 and also μ values that are considered in a way that changes of *W* are always small and less than changes of *f* , and with equations (6), (10) and (11) we can write:

$$|\text{Input}_1| > |\text{Input}_2| > |\text{Input}_3| > \cdots \qquad (12)$$

According to equations (6), (8) and (12) and considering that error feedbacks to all layers we get:

$$\Delta W_1 > \Delta W_2 > \Delta W_3 > \cdots \qquad (13)$$

And using equations (9), (12) and (13) we get:

$$\Delta f_1 > \Delta f_2 > \Delta f_3 > \cdots \qquad (14)$$

Equation (14) shows when in the suggested network, we move from the first layer toward parallel layer the input changes to

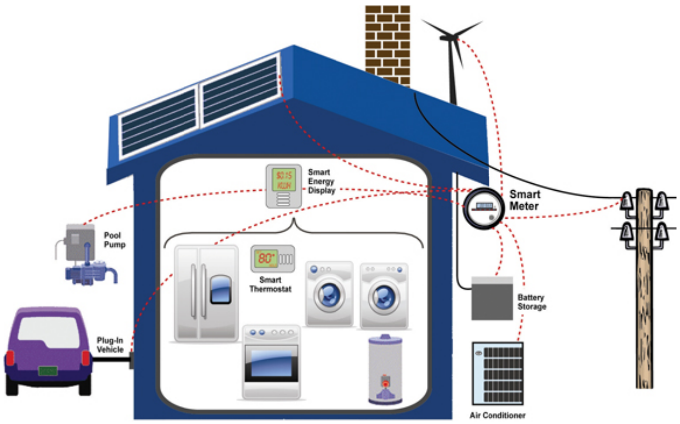

Fig. 9. Typical configuration of a smart home

next layers are diminished. Meanwhile according to network topology, in each layer including parallel layer, there is only one active neuron at a time. Since the error of parallel layer output affects all layers, we found that the input changes in the parallel layer is always the least among all layers. Also inputs of parallel layer are limited between 1 and -1. Given the characteristics of physical layer, and the convergence rule of Single Layer Perceptron [19], active neurons in the layer converge to their desired outputs, i.e. the error goes to zero and the network stabilizes.

B. Functionality

This network acts as an associative memory and is able to map a set of inputs to a desired output. The network is trained with different scenarios and operating conditions. It can be trained in a certain time and mostly maps outputs based on previous learning. The system is trained when we have a new scenario or we intend to change one. The suggested lookup table gives the system speed and reliability, which is important 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 energy generation reduces), 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. We just have *n* Scenarios which the connections between them in different situations are trained in the network.

V. 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. Figure 9 shows a typical configuration of a smart home. The fuzzy subsystem of the simulation has 5 types of inputs along with membership functions as they are shown in Figure 2. In the first category, we have 15 fuzzy rules which make proper inputs for the intelligent lookup table. In this category, there are three inputs, but the number of fuzzy rules was reduced from 27 to 15 as shown in Figure 10. For instance, when the cost is low, without considering the battery and solar radiation situations in the system, the fuzzy rule is set to Neutral-, meaning that 9 fuzzy rules are set to Neutral-. These fuzzy rules based on solar radiation, battery storage and price, produce a proper fuzzy domain output which is one of the seven energy consumption situations, mentioned in section III, and send it to the intelligent lookup table. It means that the fuzzy subsystem based on its inputs produces the first scenario and the rest of the scenarios are produced based on the control bits and feedback outputs by the intelligent lookup table. An associative memory with 38 inputs and 18 outputs is designed. Totally 4096 cells are

Fuzzy rules for the first category

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

Fig. 10. Fuzzy rules for the intelligent Energy Management System

TABLE I
SEVENTEEN DEFINED SCENARIOS FOR ENERGY EFFICIENCY IN A SMART HOME

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

considered for its parallel layer. The outputs of the fuzzy subsystem 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 functionalities. The rest of the 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. As was mentioned earlier, just the connections between scenarios are trained in the system as offline training. When there are changes or new scenarios in the system, the neural network can be trained again during off-peak hours.

The weight changes are shown in Figure 11. 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

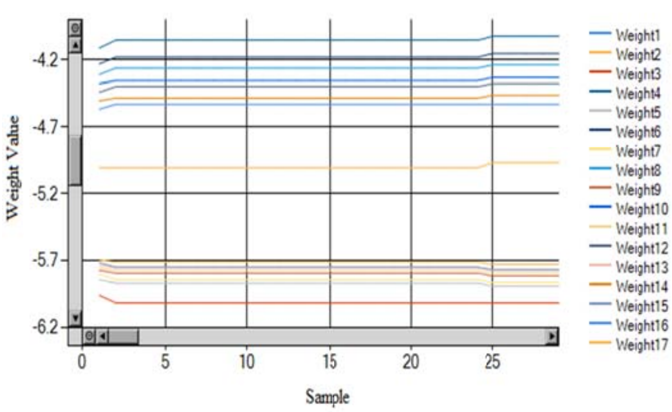

Fig. 11. The value of weight changes for simulation of system with 38 inputs and 18 outputs.

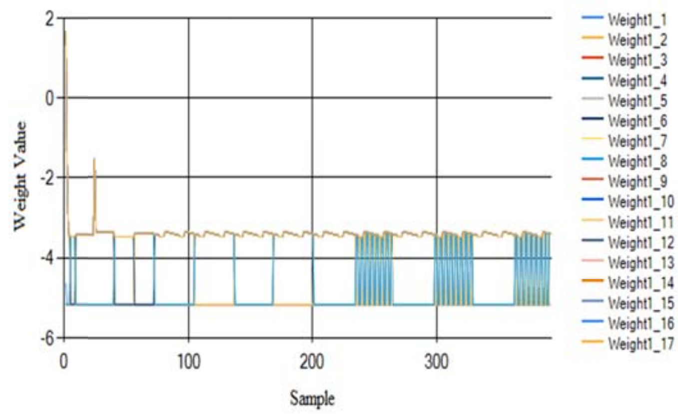

Fig. 12. Weight changes for the first lower trainable weight in parallel layer for all of the training situations.

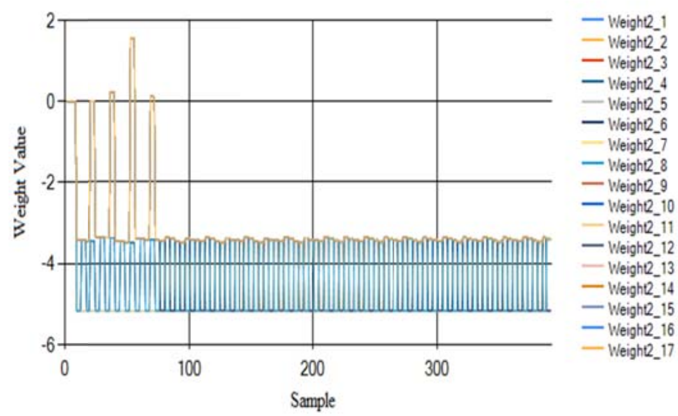

Fig. 13. Weight changes for the second lower trainable weight in parallel layer for all of the training situations.

which indicates the stabilization of the network. Figure 12 and 13 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 results, the mentioned house is simulated in GridLAB-D software [23]. The energy management system program which is written in C# language is able to connect to the GridLAB-D and change the appliances' glt files in order to get the best energy consumption in the smart home based on the previous offline training. The results for real power with and without using the intelligent EMS (*i*EMS) during one day (24 hours) are compared. As it is shown in Figure 14, by applying the *i*EMS in the house, the energy consumption is moved from peak hour to non-peak hour by the system. This system also was tested with real data for one day (10/4/2011) in New York City [24] and the result for energy consumption and cost with and without applying the *i*EMS to the system during 24 hours are compared. Figure 15 compares energy consumption and figure 16 compares the cost of energy during 24 hours for the mentioned day. As the figures show, based on the weather information and real data price, there is a 20% improvement in energy consumption and 21% saving in the cost of energy

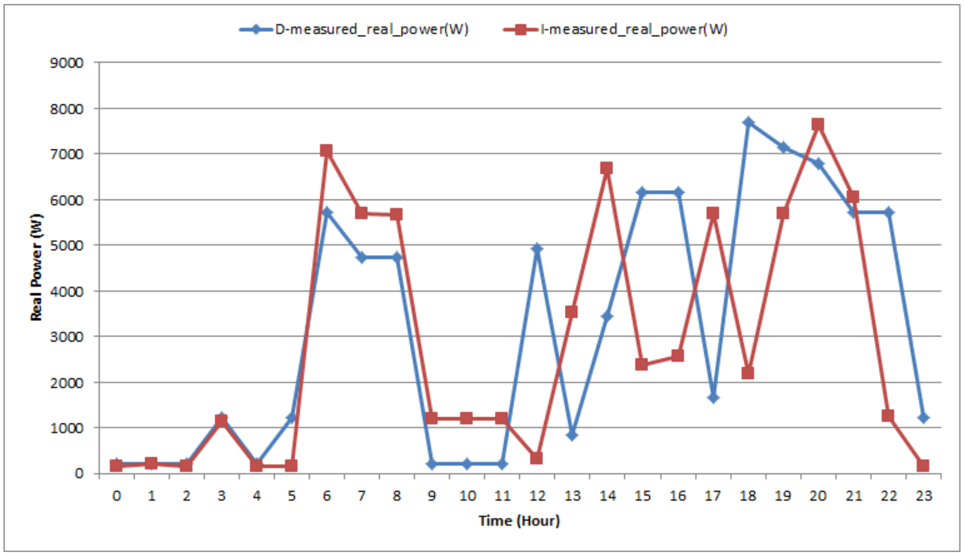

Fig. 14. Comparing measured power with and without applying the intelligent EMS to the system during one day (24 hours)

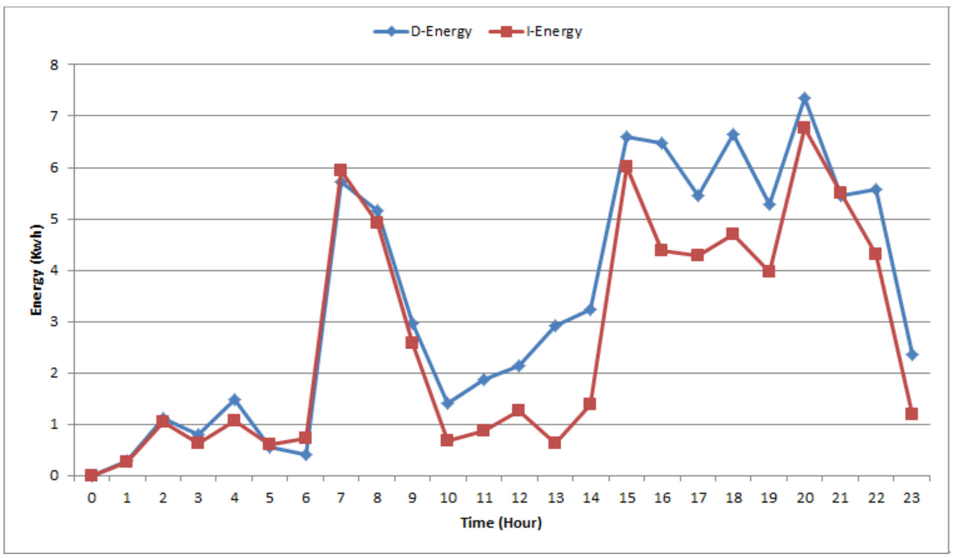

Fig. 15. Comparing energy consumption with and without applying the *i*EMS to the system during 24 hours on 10/4/2011 in New York City

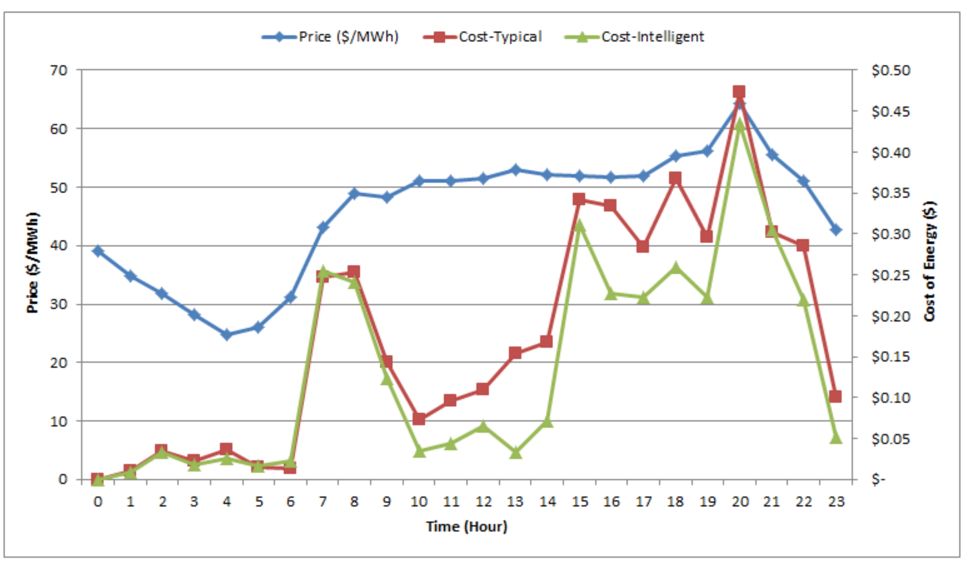

Fig. 16. Comparing cost of energy with and without applying the *i*EMS to the system during 24 hours on 10/4/2011 in New York City

which proves the positive functionality of the designed *i*EMS especially in peak hours.

VI. Acknowledgement

The authors wish to thank peers at the ACE Laboratory for reviewing this work. The assistant of Mr. Patrick Benavidez is particularly appreciated for suggestion made on improving the paper.

VII. Conclusion

An automated intelligent energy management system (*i*EMS) for residential level is presented. The suggested EMS system has two subsystems, fuzzy sub and intelligent lookup table. The fuzzy subsystem has 15 fuzzy rules along with membership functions which makes appropriate outputs for the intelligent lookup table subsystem. The intelligent lookup table is a new associative neural network which maps inputs to desired outputs. It can be trained based on the different scenarios and connections between them. The intelligent lookup table has three types of inputs. The first type of input is output of the fuzzy subsystem which represents energy consumption in the grid as a continuous variable. The second input type is that of discrete-valued control bits from fuzzy subsystem and external sensors and the last input type are feedback outputs. This automated energy management system is able to find the best energy efficiency scenario for different situations. After training the system, it is able to predict the situation in a smart house and find the best energy consumption scenario.

References

[1] O.A. Sianaki, O. Hussain, T. Dillon, A.R. 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, Bali, Indonesia, 2010.

[2] M. H. Albadi and E. F. El-Saadany, "A Summary of Demand Response in Electricity Markets," Electric Power Systems Research, vol. 78, pp. 1989–1996, November 2008.

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

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

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

[6] 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.

[7] 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.

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

[9] J. Eto, "The Past, Present, and Future of U.S. Utility Demand-Side Management Programs", Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, LBNL-39931, 1996.

[10] Peak Load Management Alliance, "Demand Response: Principles for Regulatory Guidance," February 2002.

[11] T. Rausch, P. Palensky, "PROFESY: intelligent global energy management", IEEE International Conference on Intelligent Engineering Systems (INES 05), pp. 59-64, September 2005.

[12] M. LeMay, R. Nelli, G. Gross, and C. A. Gunter, "An integrated architecture for demand response communications and control", The 41st Annual IEEE Hawaii International Conference on System Sciences (HICSS '08), Waikola, Hawaii, January 2008.

[13] A. Molderink, V. Bakker, M.G.C. Bosman, J. L. Hurink, G.J.M. Smit, "A three-step methodology to improve domestic energy efficiency", Proceedings of Innovative Smart Grid Technologies (ISGT), January 2010.

[14] T. J. Lui, W. Stirling, H. O. Marcy, "Get Smart", IEEE Power and Energy Magazine, Volume 8, Issue 3, pp. 66-78, May 2010.

[15] J. Antonio, Conejo, M. Juan, Morales, Luis Baringo, "Real-Time Demand Response Model", IEEE Transaction on Smart Grid, Vol. 1, No. 3, 2010, pp. 236–242.

[16] S. Kishore and L. V. Snyder, "Control mechanisms for residential electricity demand in smartgrids," Proc. The 1st IEEE International Conference on Smart Grid Communications (SmartGridComm '10), pp. 443–448, 2010.

[17] D. Han and J. Lim, "Smart home energy management system using IEEE 802.15.4 and ZigBee," IEEE Transactions on Consumer Electronics, Vol. 56, No. 3, pp. 1403-1410, Aug 2010.

[18] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning", Proc. of Smart Grid Communications (SmartGridComm) Conference, 2010.

[19] D. Shahgoshtasbi and M. Jamshidi, "Modified Intelligent Energy Management system in a smart house", Proc. World Automation Congress, Puerto Vallarta, Mexico, 2012.

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

[21] C. T. Leondes, "Algorithms and architectures", Academic Press, 1998.

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

[23] http://www.gridlabd.org/

[24] http://www.ferc.gov/market-oversight/mkt-electric/new-york/2011/10-2011-elec-nyiso-dly.pdf