ACCURACY IMPROVEMENT OF SOC ESTIMATION IN LITHIUM-ION BATTERIES BY ANFIS VS ANN MODELING OF NONLINEAR CELL CHARACTERISTICS

by

Mohammad Hassan Amir Jamlouie

Bachelor of Science in Electrical Engineering, University of Mazandaran, 2012

A project presented to Ryerson University

in partial fulfillment of the requirements for the degree of

Master of Engineering

in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2018

Author's Declaration

I hereby declare that I am the sole author of this project. This is a true copy of the project, including any required final revisions.

I authorize Ryerson University to lend this project to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my project may be made electronically available to the public.

Abstract

ACCURACY IMPROVEMENT OF SOC ESTIMATION IN LITHIUM-ION BATTERIES BY ANFIS VS ANN MODELING OF NONLINEAR CELL CHARACTERISTICS

2018

Mohammad Hassan Amir Jamlouie
Master of Engineering
Ryerson University

Over the last century, the energy storage industry has continued to evolve and adapt to changing energy requirements. To run an efficient energy storage system two points must be considered. Firstly, precise load forecasting to determine energy consumption pattern. Secondly, is the correct estimation of state of charge (SOC).

In this project there is a model introduced to predict the load consumption based on ANN implemented by MATLAB. The Designed intelligent system introduced for load prediction according to the hypothetical training data related to two years daily based load consumption of a residential area.

For another obstacle which is accurate estimation of SOC, two separate models are provided based on ANN and ANFIS for Lithium-ion batteries as an energy storage system. There are several researches in this regard but in this project the author makes an effort to introduce the most efficient based on the MSE of each performance and as a result the method by ANN is found more accurate.

Acknowledgements

I’d like to acknowledge my professor Dr. Kaamran Raahemifar, who always believed in my abilities and never hesitated with inspirational support.

Dedication

Dedicated to my parents who’ve supported me to achieve this success and inspired me to progress, so I can dedicate myself to this work.

Table of Contents

List of Figures

List of Tables

Introduction

Energy storage systems provide a wide array of technological approaches to managing our power supply in order to create a more reliable energy infrastructure and bring cost effectiveness to both producers and consumers. Energy storage system is categorized to different methods including Mechanical, Electrochemical, Chemical and Electrical techniques to store the excess electrical energy produced by all resources. In spite of the fact that each technic has its own merit and disadvantage, all of them have benefits for electric utilities, end user, equipment vendors, energy service providers, regulators and independent system operators and environment.

Although energy storage systems may face us some primary costs, feasibility studies show that their applications will provide more benefits than the costs are paid at first. Each system can provide some of this applications or a combination of them which includes Grid Angular Stability (GAS), Grid Voltage Stability (GVS), Grid Frequency Excursion Suppression (GFS), Regulation Control (RC), Spinning Reserve (SR), Short Duration Power Quality (SPQ), Long Duration Power Quality (LPQ), 3-hr Load Shifting (LS3), 10-hr Load Shifting (LS10).

Optimal scheduling of energy storage (ES) systems requires two pieces of information to be known. First is an accurate prediction of the load profile over a time period where the ES unit will operate and the second available energy in the system at the time of scheduling that is determined through state-of-charge (SOC) estimation.

To get more precise, there is some clarification provided regarding aforementioned prevalent energy storage systems, their benefits, applications, utilities, then we draw an analogy between them. Afterwards, Artificial Neural Network(ANN) and Adaptive Neuro Fuzzy Interface System (ANFIS) are briefly introduced. In the next step implemented model based on Artificial Neural Networks(ANN) as the target of load forecasting for optimal scheduling of ES Systems is provided.

Subsequently, there is another implementation for state of charge estimation for lithium ion batteries. In order to achieve an accurate model for SOC estimation, we consider the lithium ion batteries in cell scale and provide an estimation method based on both Adaptive Neuro Fuzzy Interface System (ANFIS) and Artificial Neural Network (ANN). The system is trained and tested according to an original data prepared for this project. Also an analogy is drawn between two methods to nominate more efficient and accurate techniques for this purpose.

The implementations of this project are performed by Artificial Neural Network (ANN) and Adaptive Neuro Fuzzy Inference System (ANFIS). The software provides aforementioned tools is MATLAB R2016a Academic for students and the device is Laptop "LENOVO ideapad Z510".

Chapter 1: Energy storage technologies

1.Energy Storage System (ES)

Power systems always face serious challenges with depletion of fossil fuels, environmental obligation, integration of renewables, electric vehicle, demand response, electricity market and smart grids. Along the same line, energy storage system is a necessity to maintain supply and demand balance and also provide ancillary services for power systems. By the way of illustration, when electricity exceeds by the other sources such as solar cells or nuclear power plants, it can be stored for a long period of time by the help of sustainable technologies such as pump hydro energy storage systems. Such energy storage systems are considered as local sources to feed the local loads which are connected together to work as microgrids and subsequently they are synchronized with the central grid called macro grids. This cooperation can meet the supply and demand satisfaction to provide more efficient and reliable system. In this section we introduce different technology of ES, benefits and applications. Here is a categorized list for prevalent energy storage methods:

1.1. Mechanical Energy Storage Systems

Electrical energy is converted to mechanical mode for energy storage. Pumped hydro energy storage (PHES), compressed air energy storage (CAES) and flywheel energy storage are the most common technologies for Mechanical energy storage systems.

1.1.1. Pump hydro energy storage (PHES)

For load levelling (peak shaving) Pumped Hydro Energy Storage (PHES) is the most successful, cost effective and widely used ES techniques available to electrical ancillaries and auxiliaries. PHES can be useful for the storage of electrical energy which is produced by solar and wind energy. Excessive electrical energy is used to water supply (lake, river or any reservoir) at the lower level to a higher reservoir with smaller size which is shown in fig. 1.1. As demand exceeds the normal generation capacity, water will flow to the lower reservoir to a hydraulic turbine to drive the electric generator to produce electrical energy in order to supply additional demand.

In most PHES the generator is reversible and by gaining power from the grid acts as water pump as well with changing from turbine to motor mode. So, only in a few minutes The overall energy recovery efficiency of pumped storage, that is, the recovered electrical energy as a percentage of electrical energy used to pump water, is about 70 per cent. [1]

Relatively, there are few appropriate sites at which there is supply reservoir at lower level and another one at higher level. It is worth to be mentioned that by the help of natural or excavated underground caverns to be considered lower reservoirs, the number of possible PHES sites are increased. The Power and Energy of PHES can be gained from equations 1 and 2.

$$\boldsymbol{E = mgh} \tag{1}$$

where,

E= potential energy stored in PHES system, J, m = mass of water body, kg, g=gravitational acceleration, m/sec2, and h = hydraulic head height, m.

and,

$$\boldsymbol{P = \rho\, \eta\, Q\, h\, g} \tag{2}$$

P = generated output power, W, ρ = fluid density (1000 for water), kg/m3, η = system efficiency, Q = fluid flow rate, m3/sec, h = hydraulic head height, m, and g = gravitational acceleration, m/sec^{2}.

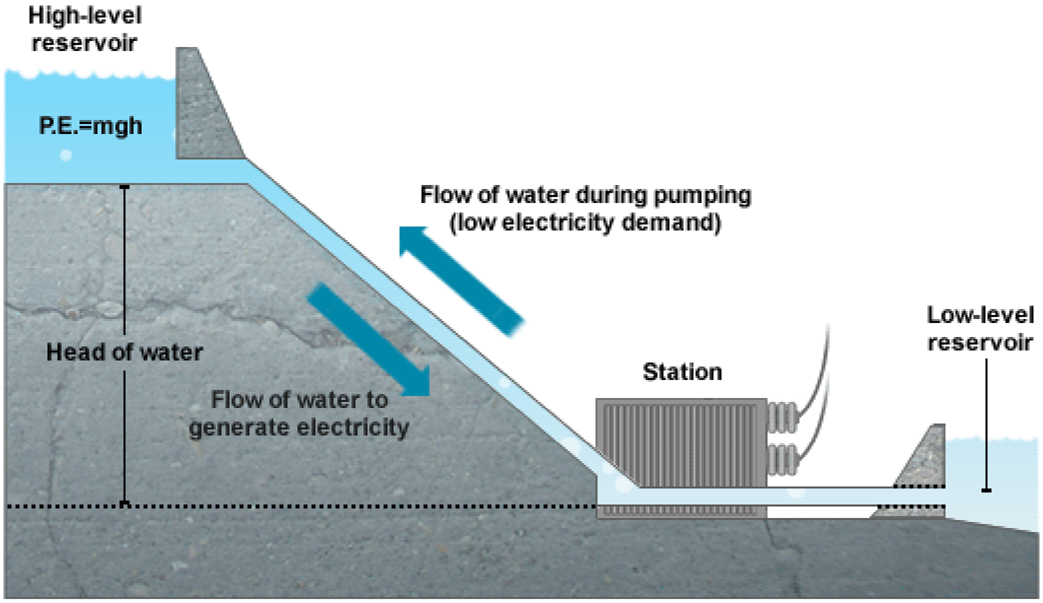

High-level reservoir
P.E.=mgh
Flow of water during pumping (low electricity demand)
Low-level reservoir
Head of water
Station
Flow of water to generate electricity

Fig. 1.1: Illustrating the Pumped Hydro principle [2]

1.1.2. Compressed air energy storage (CASH):

By Compressed air energy storage (CASH), during off-peak when the low cost electrical energy is available, excessive electrical energy is used to compress air. For power plants with energy storage in excess of approximately 100 MWh or 5 hours of storage [1] the CASH is the most cost effective method. The compressed air is most economically This technology stores compressed air underground in salt caverns, hard rock caverns or porous rock formation.

A CAES plant with underground storage must be built near a favorable geological formation. Aboveground compressed air storage in gas pipes or pressure vessels is practical and cost effective for storage plants with less than about 5 hours, however some above ground systems with up to about 10 hours of storage may be economically attractive depending on plant design and site conditions. [1]

During compassion the air is heated but before storage in order to prevent any damage to the reservoir inner wall, it is cooled. At time of additional demand, the compressed air is released and heated in a combustor and then flows to a gas turbine which is coupled to a motor/generator.

Since CASH reservoirs are huge and expensive for aboveground, underground reservoir is considered. Among the possibilities are natural caverns, deep aquifers, depleted gas or oil reservoirs mined-out rocks or salt caverns, and abandoned mines. [1] A compressed air energy storage is shown in fig. 1.2.

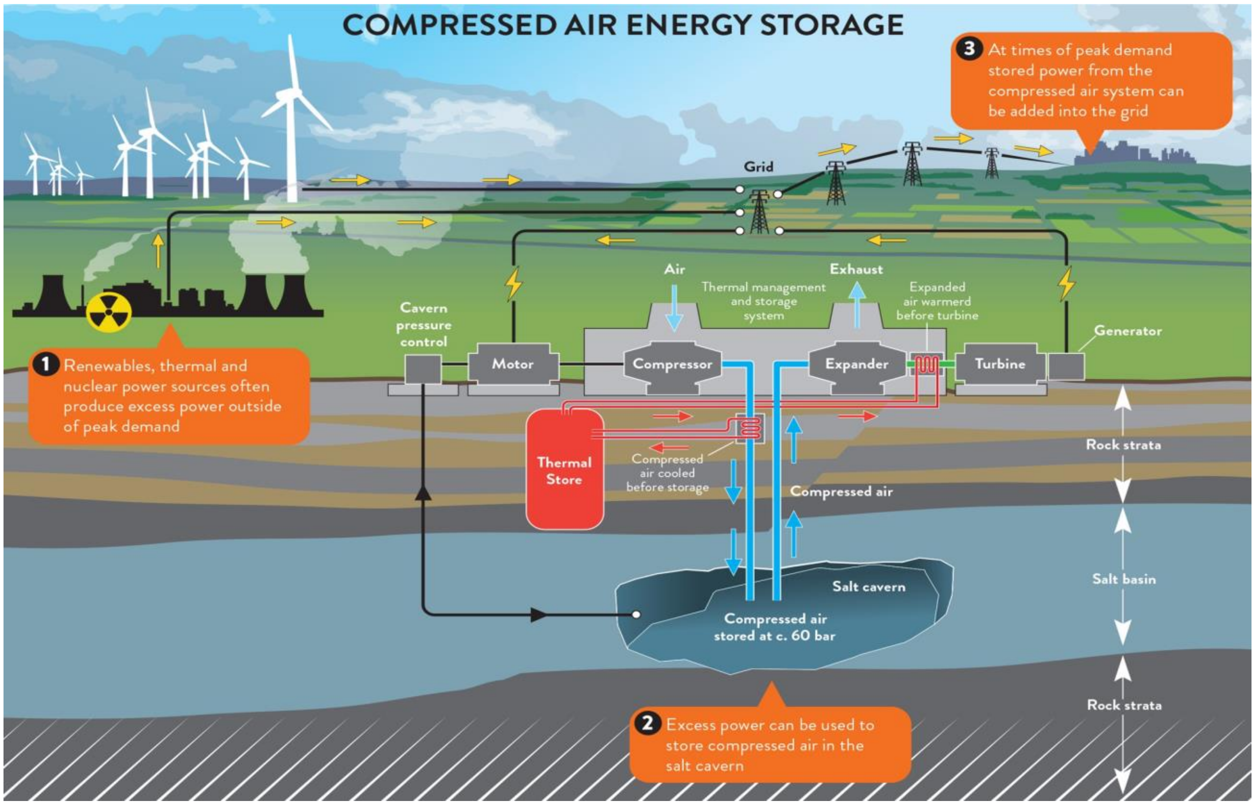

COMPRESSED AIR ENERGY STORAGE
3 At times of peak demand stored power from the compressed air system can be added into the grid
Grid
Air
Thermal management and storage system
Exhaust
Expanded air warmerd before turbine
Cavern pressure control
Motor
Compressor
Expander
Turbine
Generator
1 Renewables, thermal and nuclear power sources often produce excess power outside of peak demand
Thermal Store
Compressed air cooled before storage
Compressed air
Rock strata
Salt basin
Salt cavern
Compressed air stored at c. 60 bar
Rock strata
2 Excess power can be used to store compressed air in the salt cavern

Fig. 1.2: Illustrating the Compressed air energy storage [3]

1.1.3. Flywheel storage:

Traditionally, flywheels have been used for machines to have a smooth operation. The primary models consist of stone wheel coupled to an axel. Now, they are more complex and modern flywheels with the composition of steel rotor and magnetic bearings are considered as an energy storage to gain and provide mechanical energy by an integrated motor/generator system. The flywheels are used in supplementary UPS in different industries.

Currently, applications of this technology for the grid is in energy storage systems includes Grid Angular Stability (GAS) because of fast injection and absorption of real power for 1 – 2 sec periods and also Short Duration Power Quality (SPQ). For other applications electric vehicles and intermediate storage for renewable energy for power quality issued can be mentioned. Energy can be stored in flywheels can be calculated by eq.3.

$$\boldsymbol{E = 0.5\, J\, \omega^2\, \mathrm{m}} \qquad (3)$$

According to eq.4 J is the moment of inertia, kg.m2 (depends on mass and geometry of object) ω_m is the rotational speed, rad/sec

$$\boldsymbol{J = \int \quad rdm} \qquad (4)$$

r is the distance of each spinning mass element dm to the axis of rotation

Based in eq.5 for a circular disc flywheel of radius r and mass m:

$$\boldsymbol{J = m\, r^2} \qquad (5)$$

Circular High-Mass Rotating Body

By substituting J into E we have eq.6 as below:

$$\boldsymbol{E = 0.5\, m\, r^2\, \omega_m{}^2 \approx 0.5\, m\, v^2} \qquad (6)$$

In which ν is the linear velocity of the flywheel outer rim.

There are two different types of flywheel energy storage. High-power flywheels are more appropriate for fast discharge in order to make a prompt response to the grid. On the other hand, high energy flywheels are more used for long discharge duration. fig. 1.3 shows a schematic of flywheel storage system.

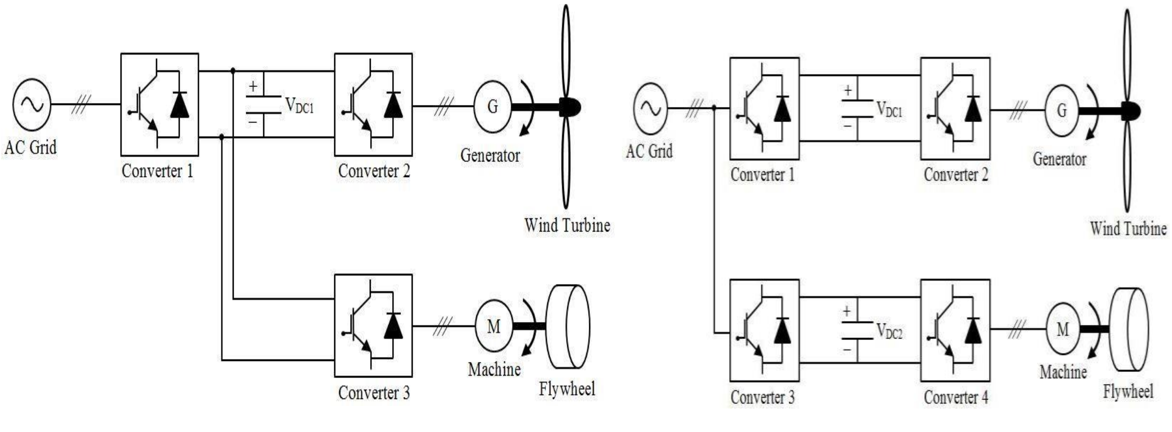


Fig. 1.3: Illustrating the Flywheel storage [1]

1.2. Electrochemical Energy Storage Systems

Electrochemical method stores energy in chemical reactants producing charges according to how batteries work. There are several types of batteries with different advantages which are clarified as below:

1.2.1. Flow batteries

They are recognized as a favorable technology for large systems, because they show flexibility regarding system design and are eminently scalable. Mechanism wise, both batteries and capacitors work based on electrochemistry but batteries have higher energy density Capacitor have higher power capability but capacitors store energy directly as charges and have higher power capability. Different types of widely used batteries and a brief analogy between them is shown in table 1.1.

Battery	Lead-acid	NaS	Li-ion	VR
Advantages	Low cost, and good charging rate	Potential low cost, high cycle life, high energy, good power density, and high efficiency	High specific energy and energy density, low self-discharge, and long cycle life	High energy, high efficiency, high charge rate, and low replacement cost
Disadvantages	Limited energy density, and hydrogen evolution	Thermal management, safety, and seal and freeze-thaw durabilities	Low charging rate, and safety (potential of fire)	Cross mixing of electrodes

Table 1.1: Different types of most used batteries and differences

Because of low cost and good charging rate, Lithium-ion batteries draw the attention of industry. For single-function applications of Lead-acid batteries Grid Angular Stability (GAS), Grid Voltage Stability (GVS), Grid Frequency Excursion Suppression (GFS), Short Duration Power Quality (SPQ) and Long Duration Power Quality(LPQ) can be mentioned.

1.3. Chemical Energy Storage Systems

Electricity can be stored during chemical reactance. One of the methods to stabilize energy during the excess of electricity generation is gasification. For instance, producing hydrogen by the help of water electrolysis and synthetic natural gas which is explained in 2.3.2. Another way to supply electrical energy during fluctuation is to have a reliable capacity for gasification of biomass and carbon containing waste in the energy sector. Gasification is the reaction of carbonaceous raw materials with steam at high temperature to produce syngas (mainly CO, CO2 and H2) [4].

When electrical energy is needed, syngas can be used for combustion to generate electricity but when electricity exceeds, syngas can be converted to different fuels. In [5], [6], [7], [8] and [9] the final processed material can be methane, diesel, methanol and also gasoline. In fig. 1.4 the concept of an energy infrastructure utilizing the storage of renewable energy in chemicals through water electrolysis and biomass gasification. Always, there is cooperation between electrolysis and gasification because the amount of hydrogen which is produced in gasification is low in comparison with the syngas conversion but it is compensated with water electrolysis [12].

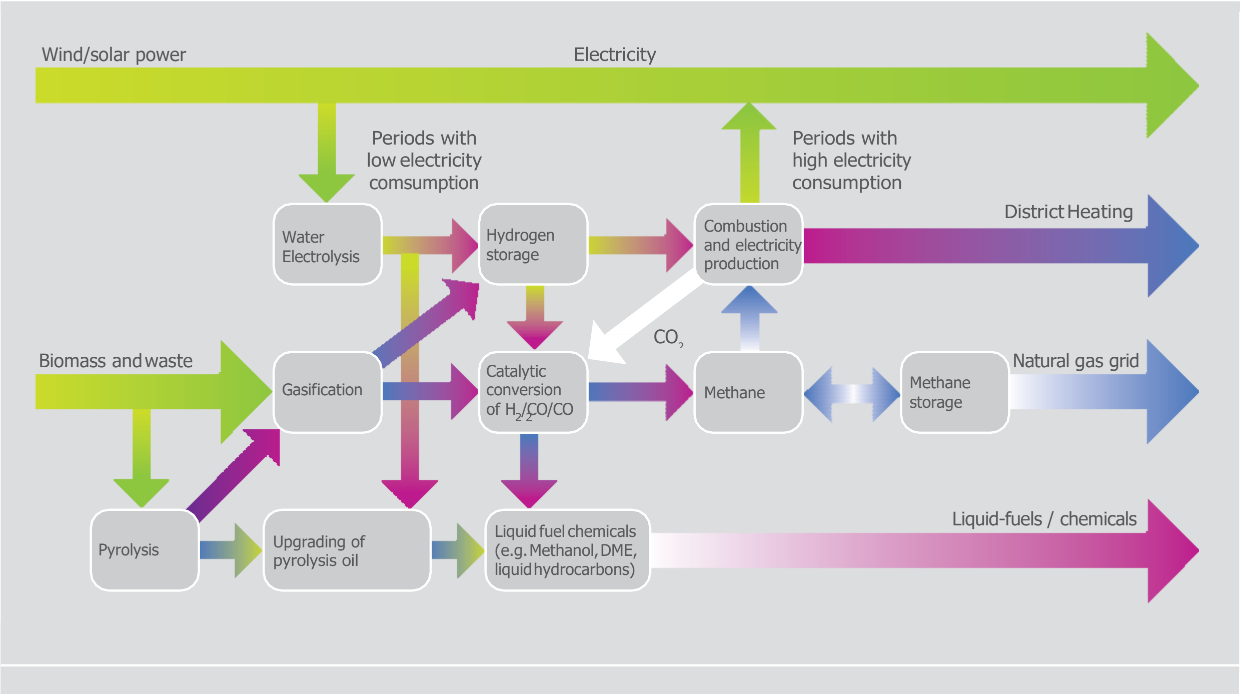

Wind/solar power
Electricity
Periods with low electricity comsumption
Periods with high electricity consumption
Water Electrolysis
Hydrogen storage
Combustion and electricity production
District Heating
CO_2
Biomass and waste
Gasification
Catalytic conversion of H_2/CO/CO_2
Methane
Methane storage
Natural gas grid
Pyrolysis
Upgrading of pyrolysis oil
Liquid fuel chemicals (e.g. Methanol, DME, liquid hydrocarbons)
Liquid-fuels / chemicals

Fig. 1.4: The concept of chemical energy storage system process [11]

1.3.1. Hydrogen

The hydrogen is a tool as energy storage molecule in itself or during of the synthesis of other storage molecules. During electrolysis process electrical energy can be converted to hydrogen, stored and re-electrified again. Hydrogen has less efficiency in comparison with batteries but it has much higher energy storage even more than CAES.

1.3.2. Synthetic Natural Gas (SNG)

Gasification is a process in which carbonaceous raw material in high temperature and steam produce a synthesis gas, which is a fuel gas mixture consisting primarily of hydrogen, carbon monoxide, and very often some carbon dioxide like CO, CO_2 and H_2. Synthetic Natural Gas (SNG) or Power-to-Gas (P2G) is one of the chemical energy storage system technologies. During electricity generation by renewable energy, when production exceeds the demand, electricity is used for water electrolysis to produce hydrogen (H_2) and oxygen (O_2). The oxygen is released to atmosphere or flows for industrial purposes. But as hydrogen is flammable, it should be inhibited. Another option is methane (CH_4) production. CH4 can then be obtained starting from CO2 and the earlier produced H2 by methanation. [10]

It is worth to mention that as the CO_2 produced in combustion process will be used in SNG, so the balance of the CO_2 is zero. With all the pieces in place, the demerit of this technology is the low efficiency (about 36%) which can be risen up to 55% by using heat recuperation. [11]

1.4. Electrical Energy Storage Systems

By this method, the energy is stored directly as charges. For the technologies of electrical energy storage supercapacitors, superconducting magnetic energy storage (SMES) and thermal energy storage system can be mentioned.

1.4.1. Supercapacitors

Supercapacitors have similar mechanism in comparison with batteries including application in energy storage, the usage of liquid electrolytes and cell configured to meet voltage, power, and energy requirements. But capacitors have higher power capability. Regarding projected applications higher energy and higher voltage applications, power quality and advanced transportation, fuel cell and micro-turbine load inrush support, leveling of fluctuating power flow from wind and solar generators can be indicated.

Supercapacitors categorized to electrostatic, electrolytic and electrochemical capacitor. fig. 1.5 shows a typical model of capacitor. The energy of supercapacitors like capacitors can be achieved according to eq.7 as below:

$$\boldsymbol{E=1/2.\ CV^2} \qquad (7)$$

$$C= \varepsilon\ A/d$$

$$\mathrm{Q} = C\ V$$

Where Q is positive or negative charge, V is the potential difference between the plates in volt, C is capacitance in Farads, ε the dielectric constant, A is the area of the plate in m^2 and d is the gap between the plates.

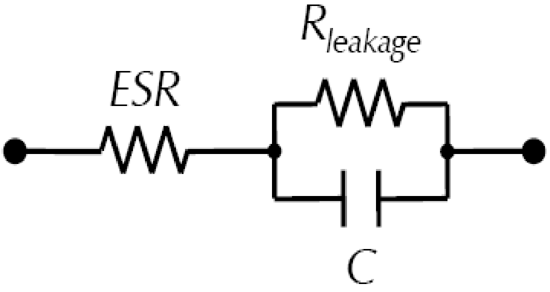


Fig. 1.5: Typical model of capacitor

RS (ESR): Internal resistance, Ω, Vr: Rated voltage, V and I_0: Fixed current, A

And by eq.8 resonant frequency, eq.9 maximum power and eq.10 discharge efficiency in load resistor and eq.11 Constant Current Charge/Discharge Efficiency in RLC circuits can be calculated as below:

$$f_0 = \frac{1}{2\pi \sqrt{LC}} \quad (8)$$

$$P_{max} = \frac{V^2}{4R_S} \quad (9)$$

$$\eta = \frac{R_L}{R_L + R_S} \quad (10)$$

$$\eta = \frac{(V_r - I_0 R_S)}{(Vr + I_0 R_S)} \quad (11)$$

1.4.2. Superconducting Magnetic Energy Storage (SMES)

Superconducting Magnetic Energy Storage (SMES) systems store energy in the magnetic field created by the flow of direct current in a superconducting coil which has been cryogenically cooled to a temperature below its superconducting critical temperature as is shown in fig. 1.6. SMES Achieves Novel Storage Technology Based on 3 Concepts:

-Superconductors carry current with no resistive loss
-Electric current induces magnetic field
-Magnetic field carries a form of energy that can be restored

In SMES since there is no conversion except AC to DC, unlike batteries there is no thermodynamic loss. Also, despite flywheel, compressed air and pump hydro energy storage systems, SMES has no any mechanical loss. Applications which are compatible by SMES are Grid Angular Stability (GAS), Grid Voltage Stability(GVS) and Short Duration Power Quality (SPQ). The energy of SMES is calculated in eq.12 as below:

$$\boldsymbol{E=0.5\ L\ I^2} \quad (12)$$

In which L is coil inductance and I is the related current.

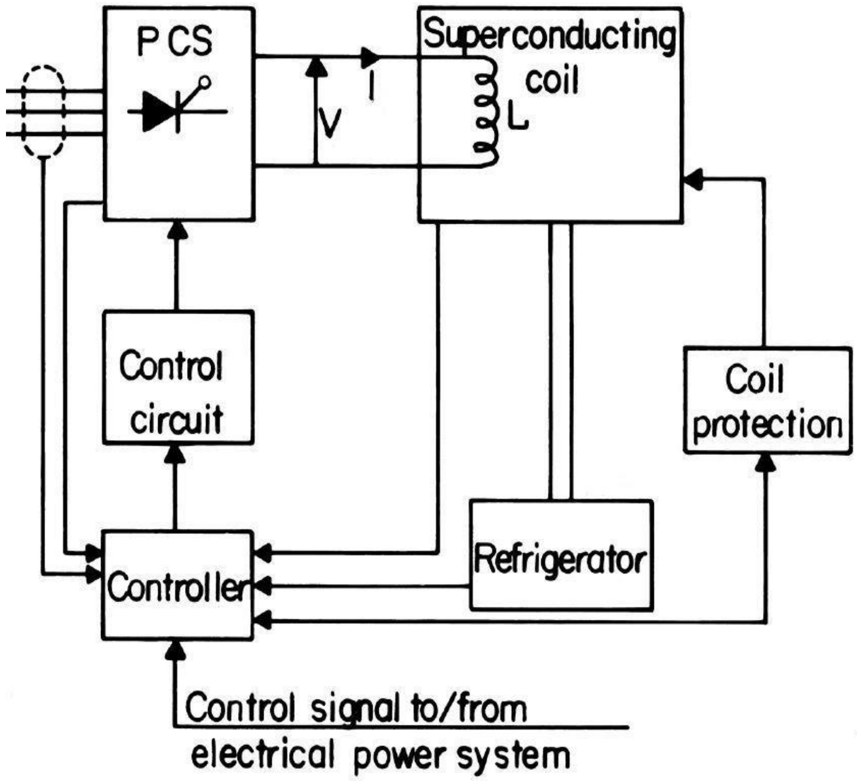


Fig. 1.6: Superconducting Magnetic Energy Storage (SMES) Structure [1]

As it is shown in fig. 1.6, the superconducting energy storage system includes a coil which is connected to the network by power converter system. Basically, there are separate controllers for coil and the PCS. There are two other units for cooling and coil protection can be seen in order to increase the cycle life of the superconducting energy storage.

1.4.3. Thermal Energy Storage Systems

There is a way to use electrical energy to store ice or freezing liquid water during the distribution of low cost electricity and use the coolness rather than air conditioning when the price is high. This section is more related to Mechanical Engineering and related fluid science, so it out to continue with other sections which is more related to the concept of this project.

After the clarification about energy storage system and different technologies, techniques which are used for implementations of this project are introduced. In this section, there is brief introduction regarding the tools have been used for this implementation. The Artificial Neural Network (ANN) and Adaptive Neuro Fuzzy Interface System (ANFIS), their application and mechanism is provided as below.

Chapter 2: Implementation techniques

2.1. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a modelling tool which is based on intelligent machine learning. This modelling method is utilized in different applications including system modeling [13], classification [14] and control [15]. The mechanism of ANN is based on biological neural networks in human brains.

The ANN learns continuously to improve its implementation by using a data as an example but without any task-specific programming. For instance, regarding image processing, ANN learn to recognize a flower image with analysis of the pictures labels “flower” and “not flower” and by using the analytic result can draw an analogy between the flower image with other images.

NN includes cells as artificial neurons which are called analogous and axons in biological brain. The neurons connect together by synapse which can act as a bridge for signal transition [16]. Neurons have state which is introduced by logic numbers 0 and 1. Neurons and synapses have different values and weight so they can change the signal strength they send to downstream. Meanwhile, they are able to be set in order to send specific signal level to downstream.

The NN structure is multi-layer in which different layers conduct various tasks. Signals from the first layer as input travel to the last layer as output after passing from other layers in between to achieve a main goal which is providing a solution based on the procedure performed in brain. Fig. 2.1 illustrates a simple scheme of neural network with three layers. The neural networks have been used in different purposes such as data prediction, image processing, cyber entertainment and games, data analysis and medical diagnosis.

An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the ANN because a neural network changes - or learns, in a sense -based on that input and output. ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found. ANN is also known as a neural network.

An ANN has several advantages but one of the most recognized of these is the fact that it can actually learn from observing data sets. In this way, ANN is used as a random function approximation tool. These types of tools help estimate the most cost-effective and ideal methods for arriving at solutions while defining computing functions or distributions. ANN takes data samples rather than entire data sets to arrive at solutions, which reduces costs. ANNs are considered fairly simple mathematical models to enhance existing data analysis technologies.

Mostly, ANNs have three layers that are interconnected including the input layer, hidden layer, and output layer. The first layer consists of input neurons. Those neurons send data on to the second layer, which in turn sends the output neurons to the third layer. Training an artificial neural network involves choosing from allowed models for which there are several associated algorithms.

In this project there two separate implementations are provided. Firstly, the ANN model for load prediction of residential complex is provided according to hypothetical daily dataset for 2 years. Secondly, there is an implementation by ANN for SOC estimation of lithium-ion batteries that is compared to another model for the same purpose which is implemented by ANFIS.

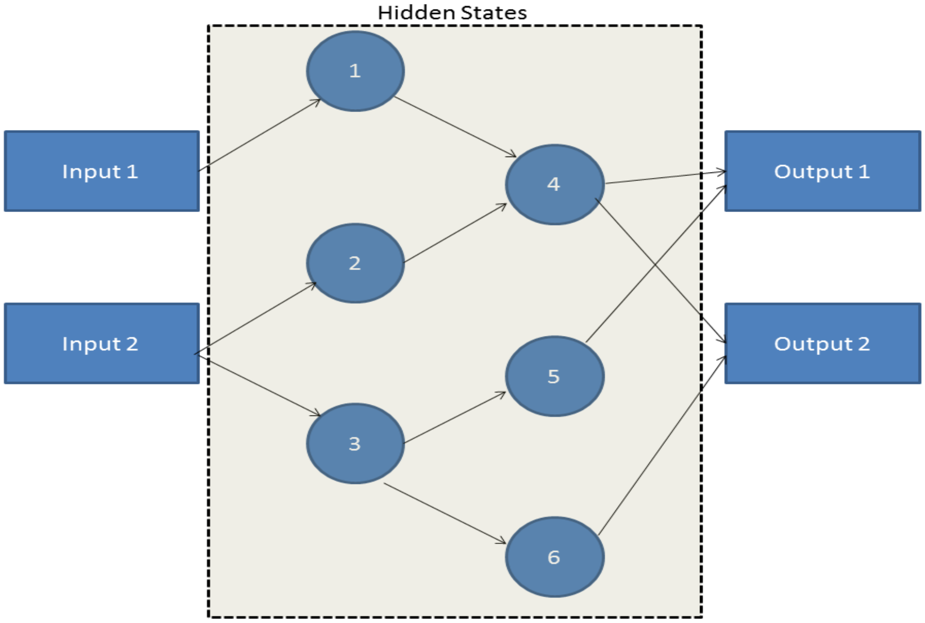


Fig. 2.1: Scheme of a simple 3-layer neural network

2.2. Adaptive Neuro Fuzzy Inference System

Generally, The ANFIS refers to an adaptive network which implements the function of a fuzzy inference system. System modeling in accordance with mathematical methods is not useful anymore on order to handle the area with ill-defined character. On the other hand, fuzzy modeling by taking advantage of if-then method and run qualitative aspects of human knowledge and behavior without any quantitative procedures. The fuzzy modelling has been explored by Takagi and Sugeno [17]. From then, several applications in prediction and estimation have been found [18] and [19]. Although there are various and numerous implementation have been done by the ANFIS, there are two aspects need more clarification and better percept:

1) No standard methods exist for transforming human knowledge or experience into the rule base and database of a fuzzy inference system.

2) There is a need for effective methods for tuning the membership functions (MF's) so as to minimize the output error measure or maximize performance index. [20]

Technically, fuzzy system includes five functional units which is shown in fig. 2.2.

-A rule base unit including a number of fuzzy if-then rules;

-A database unit which introduces the membership functions of the fuzzy sets in fuzzy rules

-A decision making unit implements the inference operations on the rules

-A fuzzification inference which transforms the crisp inputs into degrees of match with linguistic values

-A defuzzification inference unit to transform the fuzzy result of the inference into a crisp output. [21]

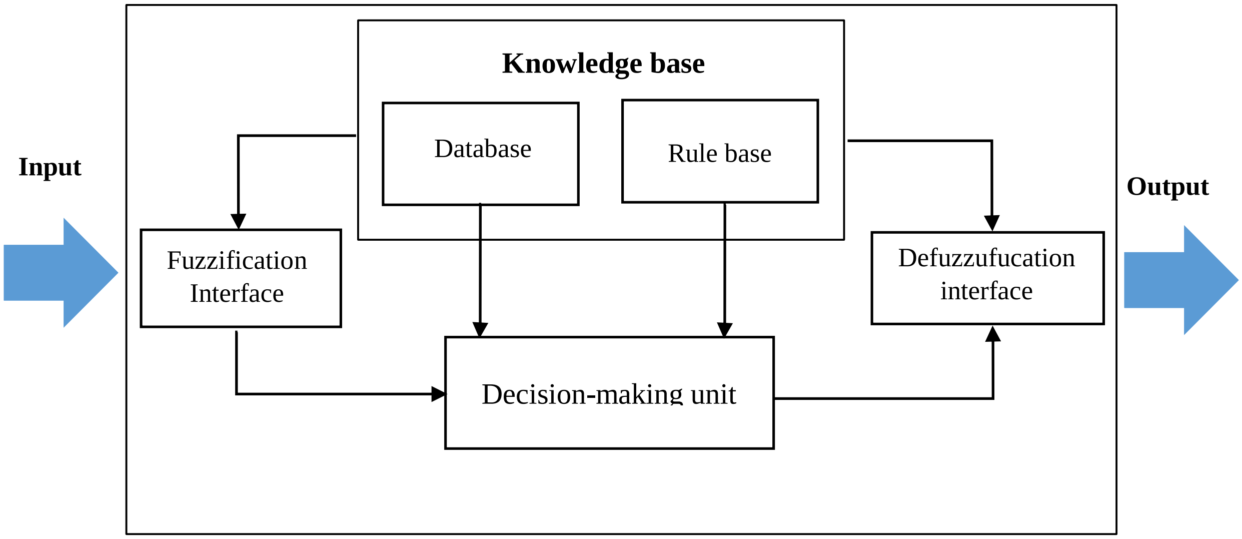
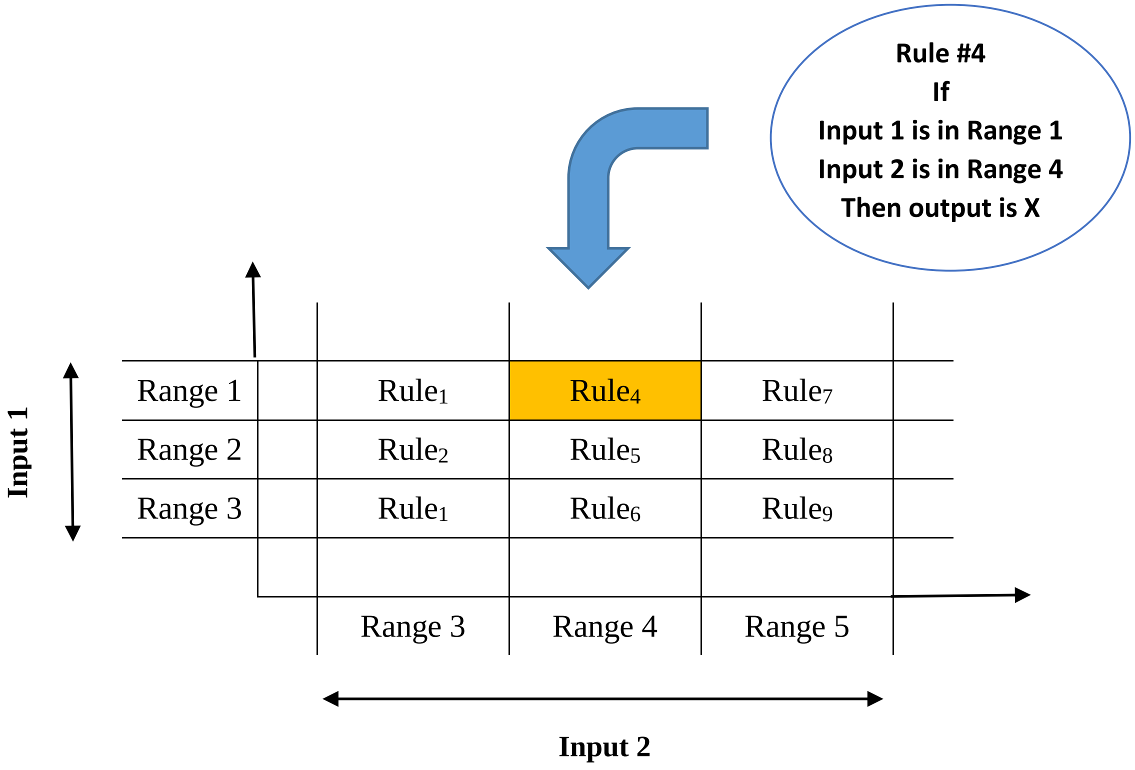


Fig. 2.2: Fuzzy inference system

There are two most famous Fuzzy rule-based Inference System as Mamdani fuzzy method and Tagaki-Sugeno (T-S) fuzzy method. Merits of the Mamdani draw considerable attentions to use this method in fuzzy projects. It’s intuitive and has widespread acceptance. Besides, it is well-suited to human cognition [21].

The T-S method is more compatible with linear techniques and guarantees continuity of the output surface. On the other hand, it has difficulties regarding multi-parameter synthetic evaluation. The problem is it cannot allocate different weight to each input and fuzzy rules. In contrast, the advantage of Mamdani method is ANFIS has the ability to learn because of differentiability during computation. However, because it is more transparent and has less componential exhaustive, the most commonly used fuzzy method is Sugeno.

The concept of the adaptive fuzzy network is shown in fig. 2.3. Based on the fuzzy partitioning of the input space, the weight of the inputs and number of membership functions the outputs are defined. According to the fig. 2.3, for instance, if ANFIS finds input 1 in range 1 and input 2 in range 4, the rule 4 will export the related output as X.



Rule #4
If
Input 1 is in Range 1
Input 2 is in Range 4
Then output is X

Range 1	Rule$_1$	Rule$_4$	Rule$_7$
Range 2	Rule$_2$	Rule$_5$	Rule$_8$
Range 3	Rule$_1$	Rule$_6$	Rule$_9$
	Range 3	Range 4	Range 5

Fig. 2.3: Fuzzy partitioning

The adjustment procedure of the fuzzy network is to set of parameters in order to reduce the error between the actual data and the output of the inference system. Along the same line, the most appropriate type of membership function shall be chosen. As shown in fig. 2.4 the most common use types of membership functions are Triangular and Gaussian Bell for 3 parameters, Trapezoidal for 4 parameters and Gaussian for 2 parameters which is used for the implementation of this project.

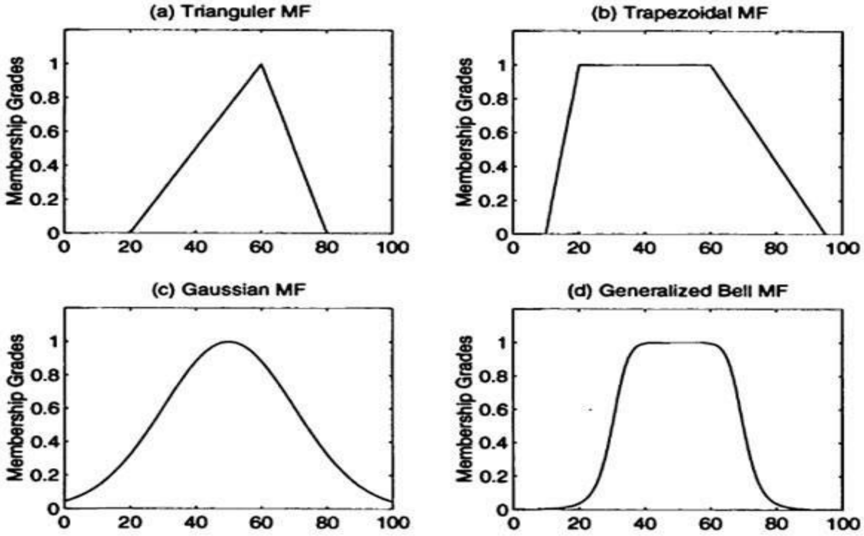


Fig. 2.4: Types of membership function
a) Triangular MF, b) Trapezoidal MF
c) Gaussian MF, d) Generalized Bell MF[21]

In this project the ANFIS technique is utilized to implement an estimation for SOC of lithium-ion batteries and drawing an analogy with the result achieved from ANN technique and going to be explained in section 5.3.1.

Chapter 3:

NN-Based Day-Ahead Residential Load Forecasting for Optimal Scheduling of Energy Storage Systems

In this section batteries are considered as a target. They are one of the most efficient and up to dated energy storage systems being used for peak shaving in order to discharge to support grid during peak time, reduce rating of grid components and asset upgrade deferral, load leveling, frequency regulation, spinning reserve to find an accurate schedule for the energy this storage system to take advantage from the benefits which can provide. Optimal scheduling of energy storage (ES) systems requires two pieces of information to be known. First is the available energy in the system at the time of scheduling that is determined through state-of-charge (SOC) estimation. Whereas, second is an accurate prediction of the load profile over a time period where the ES unit will operate. In residential load applications, the scheduled ES operating period is normally a day.

Therefore, day-ahead accurate load forecasting for residential customers becomes critical to the task of ES scheduling. Residential electricity customers have a certain pattern for consumption that could be learned through historical data. Software agents of artificial intelligence techniques have the capability to learn such pattern and complete the load forecasting job. Artificial neural networks (ANN) are among the candidates to carry out the load forecasting task, as stated many times in the literature. The development of ANN goes through multiple steps. The network size has to be first determined, and some network characteristics have to be set forth. In the training stage, an optimization algorithm is invoked to best select a number of network parameters. The training error is monitored during this phase. Finally, the ANN undergoes a testing step via a dataset whose input and output values are known. The network outputs are, therefore, compared to the outputs of the testing dataset. Testing data should not be part of the training process at all [22].

In this implementation, the target is to develop an ANN tool for 24-hour load forecasting of residential customers. A hypothetical spreadsheet file with load measurements over about two years is available. This data sets are based on daily load divided by 15 minutes steps during in 24 hours. Then, the daily load curve is represented by six constant values over six time zones as shown in fig. 3.1. Load steps of one day are known, and will be used as inputs to the ANN, which should yield the load curve for the following day. Due to different consumption patterns, seasons (summer, fall, winter, and spring) and day types (weekday and weekend) have to be differentiated. So, we assign two different indexes for season and day in this regard. Then we proceed to train the system and get the optimum result. Firstly, in following, we go to get more familiar with Artificial Neural Network (ANN) and its mechanism, then implement the procedure to train the system in optimum way to predict the load consumption for the dates required by the project manual.

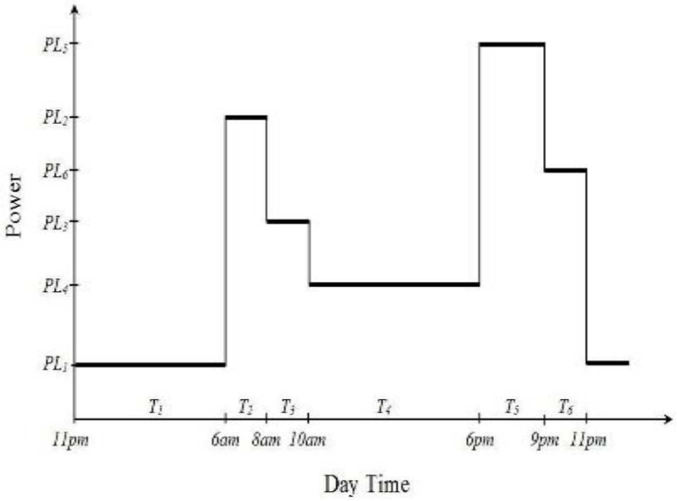


Fig. 3.1: Proposed daily load curve of residential customer

Many different neural network structures have been tried, some based on imitating what a biologist sees under the microscope, some based on a more mathematical analysis of the problem. The most commonly used structure is the networks which are typically organized in layers [23]. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in below fig. 3.2. It can be noted that the hidden layer is usually about 10% the size of the input layer. In the case of target detection, the output layer only needs a single node.

In this project the academic version of software, MathWorks Matlab®, is used. Matlab provides different types of neural networks in which various methods are applied including NN Fitting tools, NN Pattern Recognition tools, NN Clustering tools and NN Time series tools that the technique is used for this section is Fitting application. Fig. 3.2 illustrate the ANN network model with 4-layers utilized for load prediction.

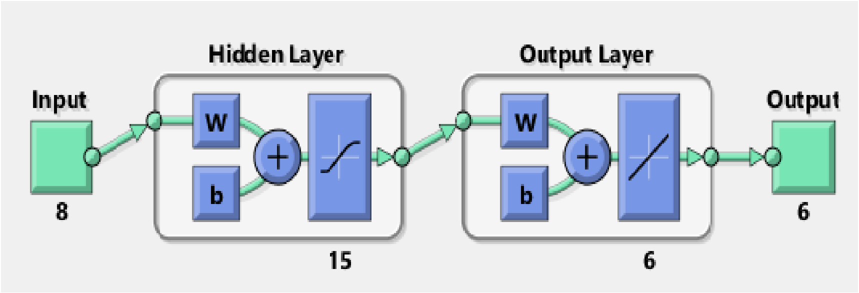
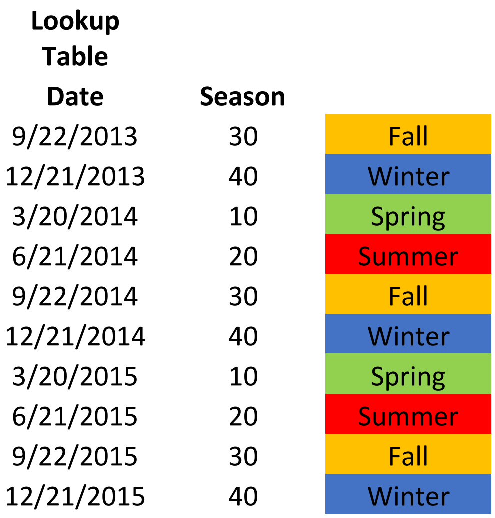


Fig. 3.2: ANN models for predicting 1-year load

3.1. Implementation

This section describes the step by step procedures for training the neural network to learn by the data from the Fall 2013 to Winter 2015 electricity load for residential customers in order to forecast load demand in arbitrary dates. The Matlab ANN toolbox was utilized in designing the network architecture. The main data list includes daily energy consumption in kW/h for every 15 minutes from the year 2013 to 2015. For each day, based on the pattern shown in fig. 3.1, data average for 6 different periods for each day is set and two separate indexes for season and day are allocated to illustrate differentiate energy consumption of different seasons which is shown in table 3.1. It should be mentioned that the index for weekdays and weekends is 1 and 0. For training the ANN, 4 separate data list are required as input training, output training, input testing and output testing which are prepared by the help of main data list. To be more clarified, each day in input data is a key for ANN to estimate the next date. For instance, for the date Feb 25th, 2014 in output, we need Feb 24th, 2014 in input.



Lookup Table

Date	Season	
9/22/2013	30	Fall
12/21/2013	40	Winter
3/20/2014	10	Spring
6/21/2014	20	Summer
9/22/2014	30	Fall
12/21/2014	40	Winter
3/20/2015	10	Spring
6/21/2015	20	Summer
9/22/2015	30	Fall
12/21/2015	40	Winter

Table 3.1: Different index allocated to various seasons

After the preparation of data sets by using the command “>>nnstart” in MATLAB we proceed to load the training input and output data in to the ANN application. In this simulation we take advantage from the load consumption for 680 days. In the next step, the number of validation and test data and also the number of hidden neurons are 3 critical parameters which shall be changed and tested several times to evaluate the result. Related MATLAB code is illustrated in appendix as command no.4.

After training the ANN the result for regression and MSE is monitored. According to table 3.2 and 3.3 the dataset for testing input testing output are used to evaluate the performance of the model.

In table 3.2 there are 10 dates have been chosen randomly from the data set as testing input and related output based on the date after testing data is illustrated in table 3.3.

It should be mentioned that the quantitative method which is used to make a comparison between the simulated data and actual data is mean square error (MSE). The MSE is computed based on eq. 13v as follows:

$$MSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \qquad (13)$$

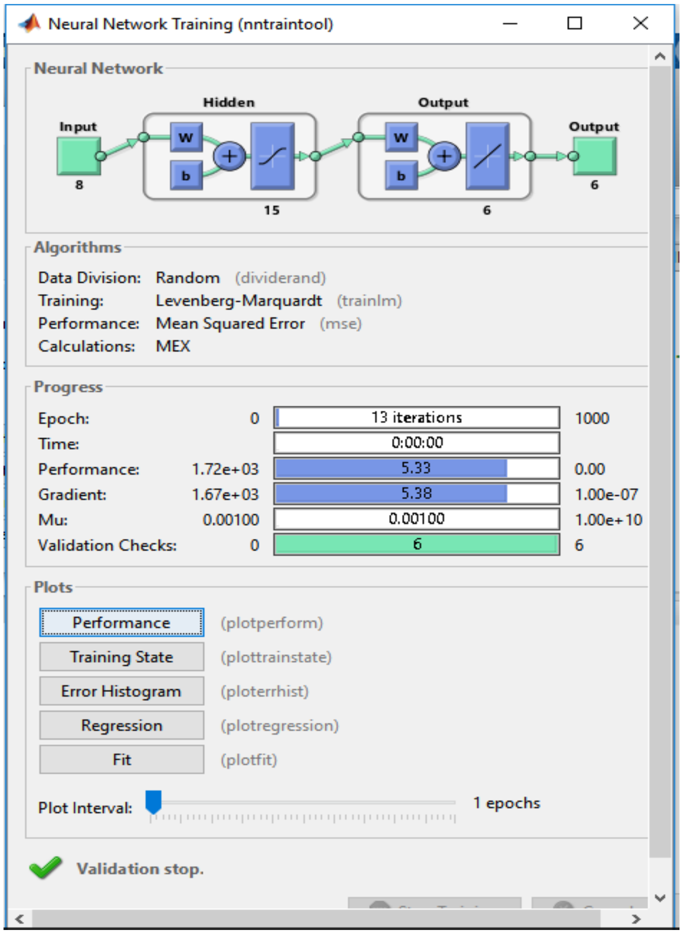
	2/24/2014	5/16/2014	7/22/2014	11/13/2014	12/27/2014	1/17/2015	4/5/2015	7/10/2015	8/25/2015	10/28/2015
T1	25.935	12.743	11.813	21.3214	18.2785	23.044	19.257	11.710	8.367	12.452
T2	27.262	13.38	13.29	22.05	19.02	21.69	19.462	11.16	9.9225	14.4225
T3	27.21	14.355	15.3	20.677	20.865	24.33	20.407	12.57	7.6425	15.81
T4	28.04	21.071	13.867	24.221	28.010	30.93	19.638	16.126	10.792	17.1018
T5	34.27	19.92	13.55	27.705	26.795	30.305	22.03	21	15.03	20.445
T6	31.665	16.5675	12.607	23.925	26.3625	29.587	18.292	23.175	12.562	18.6525
S.I	40	10	20	30	40	40	10	20	20	30
D.I	1	1	1	1	0	0	0	1	1	1

Table 3.2: Testing Input

	2/25/2014	5/17/2014	7/23/2014	11/14/2014	12/28/2014	1/18/2015	4/6/2015	7/11/2015	8/26/2015	10/29/2015
T1	25.8471	12.8528	7.56857	22.5021	21.2207	23.4878	18.282	12.0642	8.297	14.565
T2	27.96	11.52	8.705	20.28	20.2725	25.065	19.3425	12.0675	9.5925	17.715
T3	28.32	11.49	10.2075	22.4775	21.6	25.65	17.415	18.495	8.76	17.7675
T4	29.145	14.6625	12.5618	30.9075	26.6156	27.2156	20.2218	21.5718	10.1981	17.3868
T5	34.065	13.8	14.1783	34.75	32.335	30.795	22.905	20.345	15.365	21.805
T6	32.3625	17.85	14.3825	29.6775	29.0775	30.8025	19.2975	19.995	13.095	18.7575

Table 3.3: Actual data for 10 desired dates (Testing Output)

The first training has been performed by the Neural Network Fitting Tool (NNFT) with Validation Sample:15, Testing Sample: 5 and Hidden neurons:10. In each trial the ANN will be evaluated by the testing input as it is shown in table 3.2. The result will be assessed by the MSE. The target is to achieve the lowest mean square error in order to get the result closest to the actual data. The MSE is computed by the MATLAB based on the command no.1 which can be seen in appendix. Fig. 3.3 shows the ANN training process. Also fig. 3.4 and fig.3.5 illustrate the performance and the regression analysis of the ANN with aforementioned parameters. All other performances and regressions are similar to this part but the values are different which is not significant, because the MSE is the comparative element of result evaluation. By several training processes in NNFT the best value for parameters achieved as Validation Sample:15, Testing Sample: 15 and Hidden neurons:15.



Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Epoch:	0	13 iterations	1000
Time:		0:00:00	
Performance:	1.72e+03	5.33	0.00
Gradient:	1.67e+03	5.38	1.00e-07
Mu:	0.00100	0.00100	1.00e+10
Validation Checks:	0	6	6

(plotperform)
(plottrainstate)
(ploterrhist)
(plotregression)
(plotfit)

Plot Interval: 1 epochs

Validation stop.

Fig. 3.3: ANN training process Validation Sample:15, Testing Sample: 5 and Hidden neurons:15

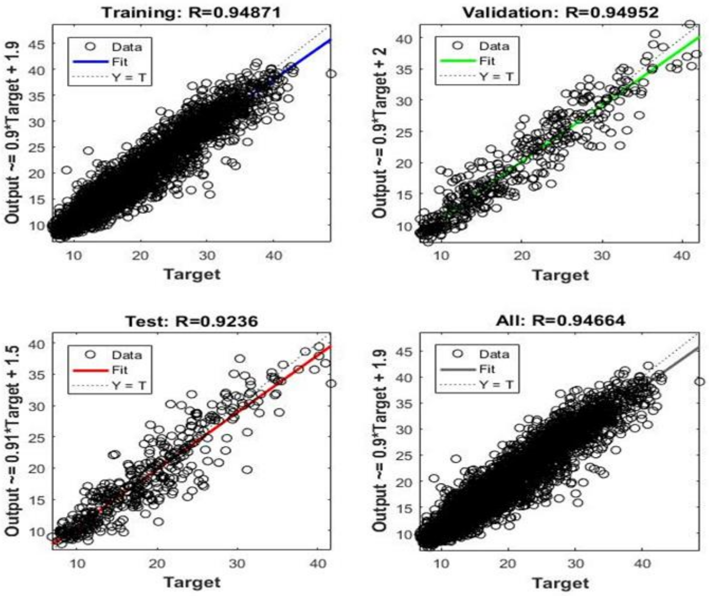


Fig 3.4: ANN regression analysis Validation Sample:15, Testing Sample: 5 and Hidden neurons:15

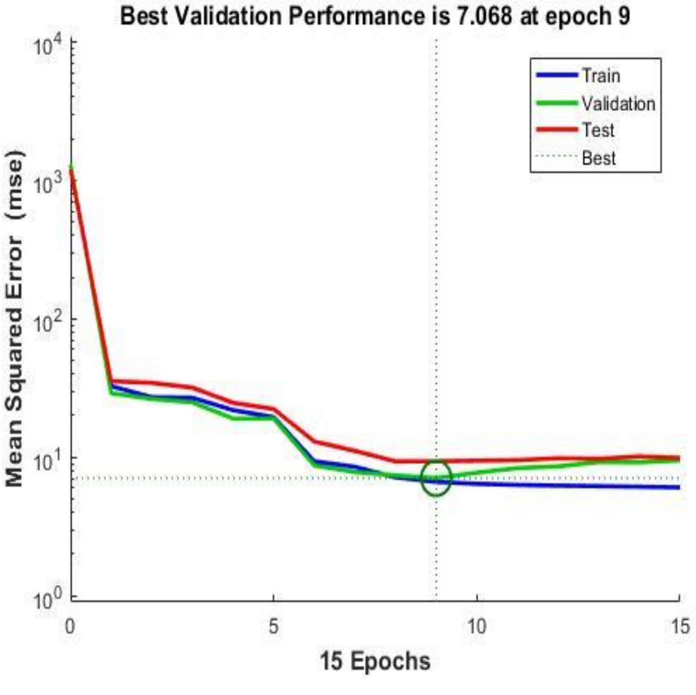


Fig. 3.5: ANN validation performance

In fig.3.5 the best validation performance occurred at epoch 9 and according to the MSE and also the gap between the validation and test curve (green and red), the parameters allocated to this training cannot meet the satisfaction for an accurate load prediction, though the regression parameters in fig. 3.4 can be acceptable. After the first implementation the ANN is tested by the dataset in table 3.3. Table 3.4 shows the output data for the first trial. The MSE for this step calculated as MSE= 2.4110.

	2/25/2014	5/17/2014	7/23/2014	11/14/2014	12/28/2014	1/18/2015	4/6/2015	7/11/2015	8/26/2015	10/29/2015
T1	26.5771	12.4053	10.3934	20.5202	17.2186	21.7481	16.6651	12.5494	8.41434	13.9979
T2	27.2829	13.3368	12.0524	21.6816	17.4573	22.1969	18.3702	12.2947	9.3785	14.6964
T3	27.6728	14.2403	12.0791	21.7387	19.4774	23.8218	17.4620	13.7487	9.8565	15.3595
T4	30.0834	17.6311	13.9628	24.7608	23.4582	27.2830	18.0679	17.7538	12.2716	18.74891
T5	34.2173	19.6732	16.3347	28.4676	27.7938	31.6070	22.0264	20.6708	14.8907	22.0411
T6	31.7473	18.1150	14.8920	24.7784	26.2487	30.3755	19.0079	18.9834	14.1405	19.3163

Table 3.4: Estimated data for different dates for Validation: 15, Testing: 5 and Hidden neurons:10

According to above performance, to make a visual comparison between the actual data and ANN estimation, by the command of ‘stairs’, related plots for the first and also the best simulations for the dates 23/07/2014 and 18/01/2015 are printed and shown in fig.3.6 and fig.3.8.

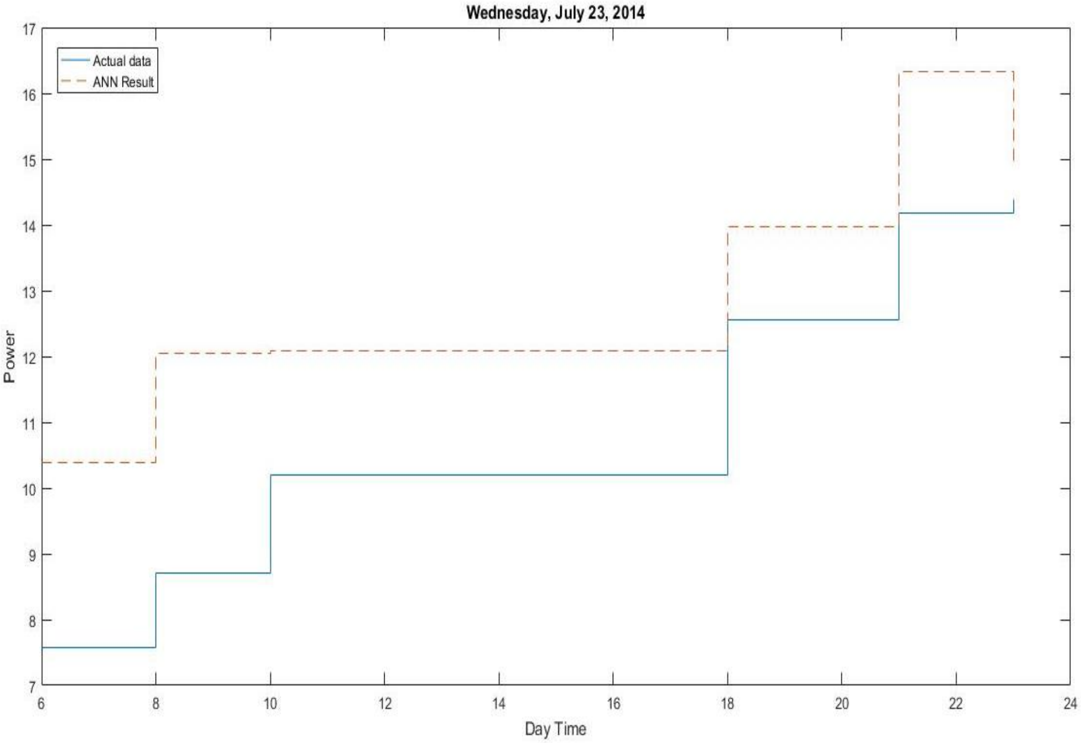


*Fig. 3.6: Actual VS estimation for Validation 15, Testing 5, Hidden Neurons: **10**- Date3: 23/7/2014*

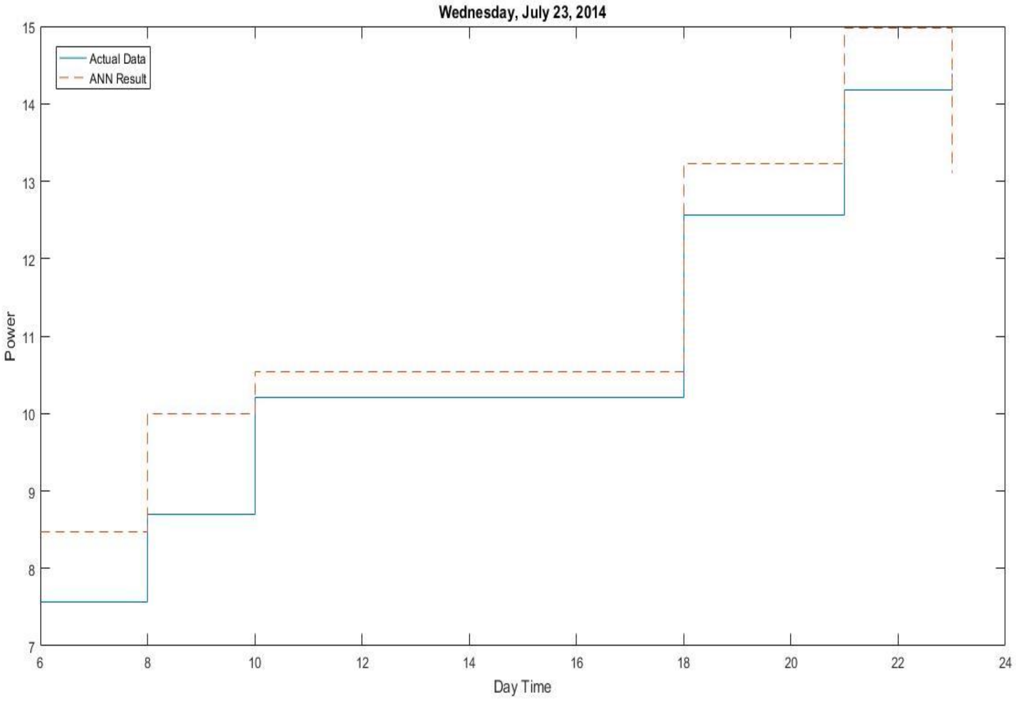


Fig. 3.7: Actual VS estimation for Validation 15, Testing 15, Hidden Neurons: ***15***\- *Date3: 23/7/2014*

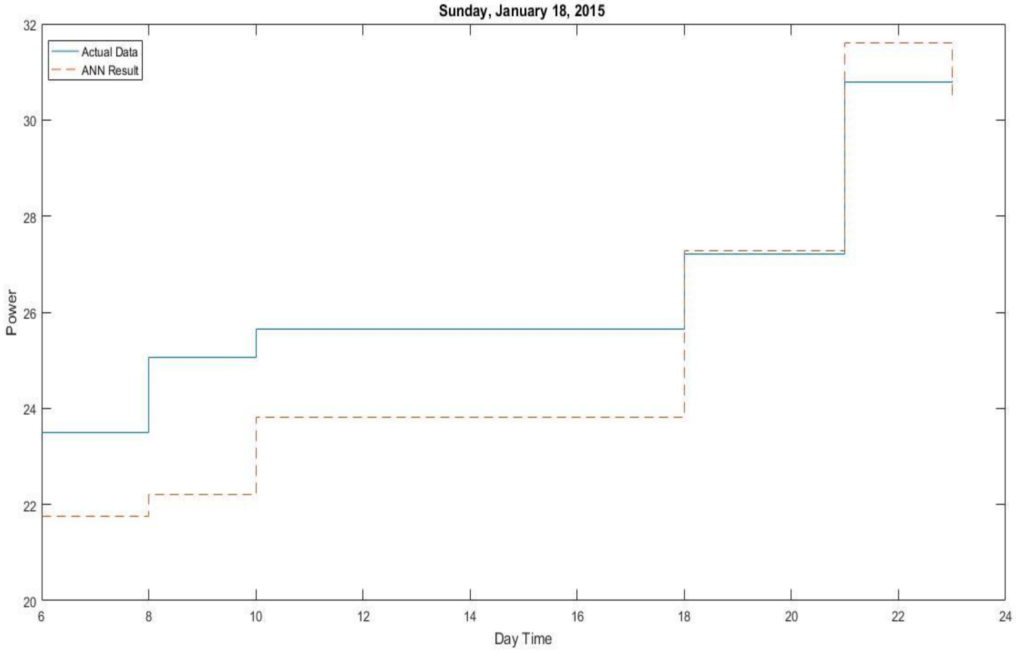


Fig. 3.8: Actual VS estimation for Validation 15, Testing 5, Hidden Neurons: 10- Date6: 18/1/2015

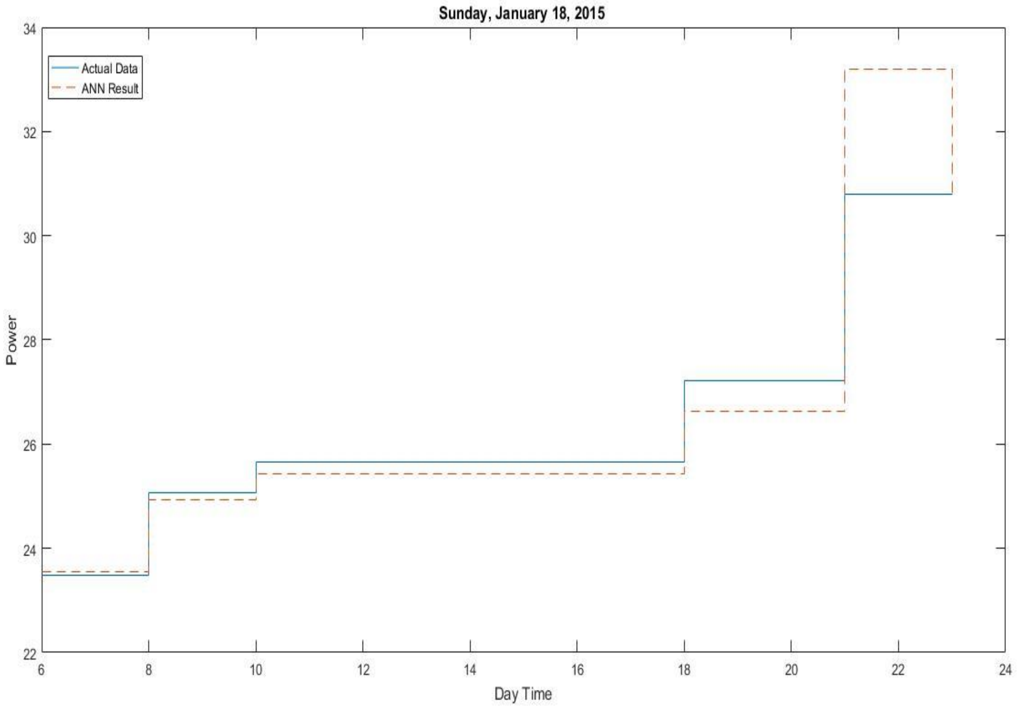


Fig. 3.9: Actual VS estimation for Validation 15, Testing 15, Hidden Neurons: 15- Date6: 18/01/2015

According to several results achieved from different training and testing, the most accurate result can be gained from NNFT technique can be shown in table 3.5. The comparison between the first and the last simulation is also provided in table 3.6.

	2/25/2014	5/17/2014	7/23/2014	11/14/2014	12/28/2014	1/18/2015	4/6/2015	7/11/2015	8/26/2015	10/29/2015
T1	26.6526	13.6855	8.46452	20.2074	19.69627	23.5397	16.4822	12.6688	8.1557	14.3952
T2	27.7317	14.5947	9.99672	21.1243	20.73791	24.9430	19.0960	12.4344	9.0826	15.2236
T3	28.0549	15.7401	10.5331	21.6828	21.3828	25.4324	18.2973	13.9671	9.8332	15.9969
T4	29.875	18.6777	13.2236	25.1499	23.0119	26.6326	17.9765	17.8226	12.4807	18.7237
T5	34.5247	20.5601	14.9787	28.4483	28.2094	33.1908	22.4312	20.2681	15.1469	22.5527
T6	32.1562	18.8656	13.1105	25.6300	25.5501	30.7803	19.4803	18.5878	14.2452	20.0675

Table 3.5: Estimated data for different dates for Validation: 15, Testing: 15 and Hidden neurons:15

	MSE
The 1st trial	2.4110
The last trial	2.2833

Table 3.6: MSE for trials

3.2. Conclusion

More accurate forecasting makes a good contribution to adjust an optimal scheduling to have an efficient energy storage system. Optimal scheduling of energy storage systems requires two pieces of information to be known. First is the available energy in the system at the time of scheduling that is determined through state-of-charge (SOC) estimation which will be explained in the section. Whereas, second is an accurate prediction of the load profile over a time period where the ES unit will operate. In residential load applications, the scheduled ES operating period is normally a day. Therefore, day-ahead accurate load forecasting for residential customers becomes critical to the task of ES scheduling.

In this part, by the favor of Artificial Neural Network (ANN) an artificial intelligence is modeled to forecast the load consumption. For this purpose, by the help of daily hypothetical load for 2 years which are divided by 15 minutes, related data sets including training of the neural network is prepared.

The mechanism of ANN forecasting for a specific date is training the system by the data provided for the day prior to that date. Due to different consumption pattern in a day, all 4 data set organized based on 6 different periods in 24 hours. By taking advantage from ANN toolbox in MATLAB, the system is trained by the initial values for Validation Sample:15, Testing Sample:5 and Hidden neurons:10. After calculation the MSE and the monitoring the regression and errors for this result, we tried to retrain the system several times to decrease the MSE and more accurate values. At the end, after several trial and error implementations, the best result achieved as below: Validation Sample:15, Testing Sample: 15 and Hidden neurons:15 with the lowest MSE as 2.2833 among all implementations.

Chapter 4:

Accuracy Improvement of SOC Estimation in Lithium-Ion Batteries by ANFIS and ANN Modeling of Nonlinear Cell Characteristics

As mentioned before, to achieve an optimal scheduling of energy storage system, in addition to have an accurate load prediction, there is a precise estimation for state of charge (SOC) is needed. This section presents two separate intelligent systems for the battery management system (BMS) to schedule a series of lithium-ion battery for energy storage purpose. Because of high energy density, good self-discharge rate and better discharge voltage in comparison with other battery types and also wide usage in portable electronic device and considerable advancement in weight, power and capacity, lithium-ion battery pack is considered for this project.

The essential part of this section is based on an adaptive neuro-fuzzy inference systems (ANFIS) and Artificial Neural Network (ANN). In this simulation we train the ANFIS and ANN separately based on the technical specifications provide by manufacturer as shown in fig.4.1. This technical data is non-linear relation between Open Circuit Voltage (OCV) and SOC for 6 different temperatures from -30°c to 55°c provided by the manufacturer. By the curve fitting technique, accurate 5 order polynomial equations of the curves are extracted as below in eq. 14 [23] and related coefficients are shown in table 4.1.

$$\mathbf{y = a_5\, x^5 + a_4\, x^4 + a_3\, x^3 + a_2\, x^2 + a_1\, x + a_0} \qquad (14)$$

Temperature	a_5	a_4	a_3	a_2	a_1	a_0
55 ^{0}C	−4.1599	8.3420	−4.6414	0.5083	0.6186	3.4710
40 ^{0}C	−2.1649	3.2388	−0.0489	−1.1715	0.8145	3.4770
25 ^{0}C	−2.6941	4.2599	–0.5109	−1.2740	0.8955	3.4613
0 ^{0}C	−0.2981	0.5460	1.1475	−1.6024	1.0298	3.3935
−20 ^{0}C	−0.5971	0.6231	0.8659	−0.5327	0.3995	3.4431
−30 ^{0}C	25.5020	−78.174	92.857	−52.03	14.117	2.0284

Table 4.1: Coefficient of approximating polynomial

The result is the nonlinear model in which there OCV and SOC are available in any arbitrary different temperatures for a lithium-ion battery pack. With all the pieces in place, the comparison between two aforementioned methods is made to choose the efficient method based on MSE calculation to increase the accuracy of Coulomb counting at the cell level.

4.1. Theory

Balancing power supply and demand is always a complex process. Lithium-ion (li-ion) batteries in particular are the subject of much interest as they have a high energy density for energy storage. State of charge estimation is of great significance for the safe operation of a lithium-ion battery (LIB) pack. The efficiency and accuracy of the algorithm are two important indicators of its performance. [24] One of the challenges facing the lithium-ion battery technology is the accurate estimation of state of charge (SOC). The SOC is related to the open-circuit voltage (OCV) of the battery.

There are numerous researches conducted for SOC estimation of batteries. For several implementations complex mathematical algorithms are used. One of these algorithms is Linear Quadratic Estimation (LQE) which is known as Kalman Filtering considered as liner model in [25], nonlinear electrochemical model in [26] in order to reduce the measurements and have more accuracy. An adaptive extended Kalman Filtering model is provided in [27] and [28] to accurately identify nonlinear parameters of Lithium-ion batteries.

Another tool to estimate the SOC is Artificial Neural Network (ANN). In [29] by 480 charge/discharge cycle the capacity of a lithium-ion battery pack is evaluated and in [30] a hybrid intelligent model based on fuzzy logic neural network and genetic algorithm is introduced. A combination of ANN which is trained offline with Kalman Filtering to eliminate the noise is also conducted in [31] In all aforementioned implementations the main disadvantage is mathematical aspect and equations which shall be considered and as a result, the complexity will increase accordingly.

Besides, neuro fuzzy inference system (ANFIS) is another way to model an intelligent system for SOC estimation. According to [32] over training with huge data is an obstacle for ANFIS to achieve an accurate estimation. In [33] a model introduced based on the nonlinear relationship between SOC and OCV with considering the considering RC model of battery.

In addition, there are several online estimation techniques have been performed with taditional Coulomb Counting method. In [34] and [35] because the models are designed based on pack level instead of cell level, it is assumed that all cells are working in the same voltage and temperature. This concept can reduce the accuracy.

In [23], a model based on Coulomb Counting and ANFIS is designed during offline mode. In comparison with all aforementioned techniques, this method is easy to implement and is a great deal more accurate than traditional Coulomb counting. Moreover, the relation between temperature and voltage is considered based on cell level to consider the battery characteristic supplied by related manufacturer. As already mentioned, ANFIS will be get confused by the huge training data, so this technique can take advantage from lower data during shorter cycle time.

The standard method adopted by the industry uses Coulomb counting, which initializes the SOC corresponding to OCV during an idle operation state. The DC current is continuously measured during charging/discharging at definite time intervals. The current times time is added/subtracted to/from the stored energy to update its value, hence SOC. The SOC is corrected at the next idle state to neutralize the effect of accumulated integration error. This technique has two drawbacks which add to its inaccuracy. First, it does neglect the effect of temperature. Second, it works at the pack level such that cell imbalances are neglected.

It is strongly believed that working with Coulomb counting at the cell level accounting for different cell temperatures would increase the accuracy of the Coulomb counting method which remains preferable by the industry. The major obstacle is to model the nonlinear cell characteristics which relate cell SOC to cell OCV at different temperatures. Nevertheless, the battery management system (BMS) usually monitors cell voltages and temperatures. The objective is to develop a nonlinear modeling tool which yields the cell SOC if the cell OCV and temperature are known. Coulomb counting will continue at the cell level.

With all the pieces in place, the method which is introduced in [23] is implemented again and tested to evaluate the MSE. In the next step by the same training data, another model based on ANN is introduced and in the end, there is an analogy is drawn between these two techniques to choose more accurate estimation for lithium-ion battery SOC.

4.2. Problem statement

The traditional Coulomb counting technique for battery SOC estimation is preferred by most BMS manufacturers due to its simplicity and ease of implementation. [23] As for each cell the level of voltage and temperature varies during all operation modes, this method is inaccurate. To increase the estimation accuracy, we utilize a nonlinear modeling tool will employ adaptive neuro-fuzzy inference systems (ANFIS) and Artificial Neural Network (ANN) which learns the cell characteristics as given by the manufacturer, fig.4.1. The characteristics show how SOC varies with OCV at different temperatures. The curves slightly outside, the training range. ANFIS and ANN should be able to generate other curves at different temperatures, which is part of the testing procedures.

4.3. ANFIS modelling of nonlinear cell characteristic

The nonlinear modeling tool will employ adaptive neuro-fuzzy inference systems (ANFIS) which learns the cell characteristics as given by the manufacturer, fig. 4.1. The characteristics show how SOC varies with OCV at different temperatures. The curves are approximated within the operating range through fifth order polynomials. Polynomial points are generated to form ANFIS training data. ANFIS testing is carried out within, and slightly outside, the training range. ANFIS should be able to generate other curves at different temperatures, which is part of the testing procedures.

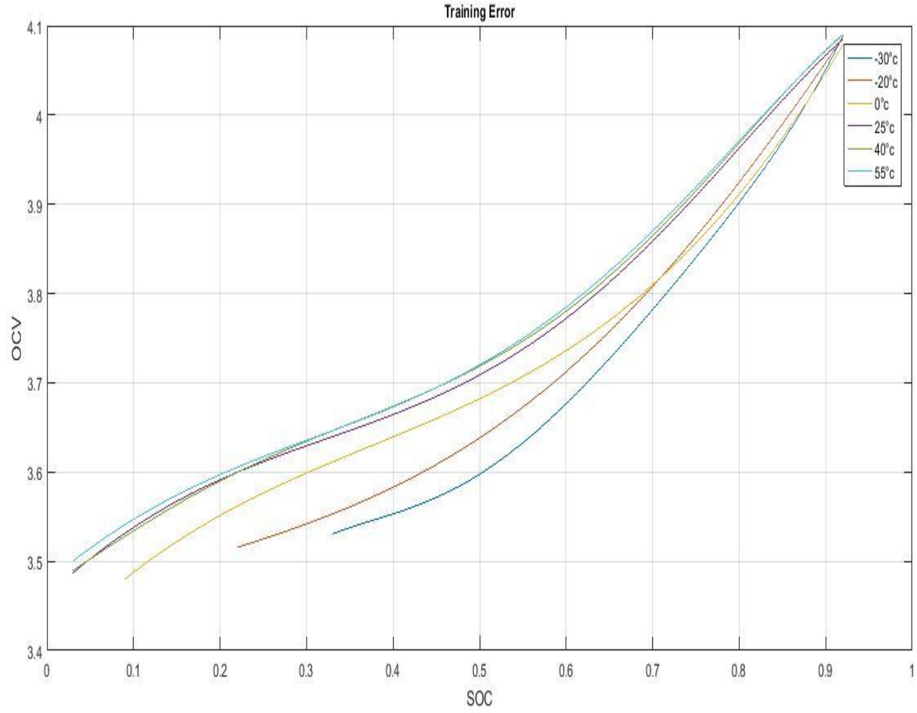
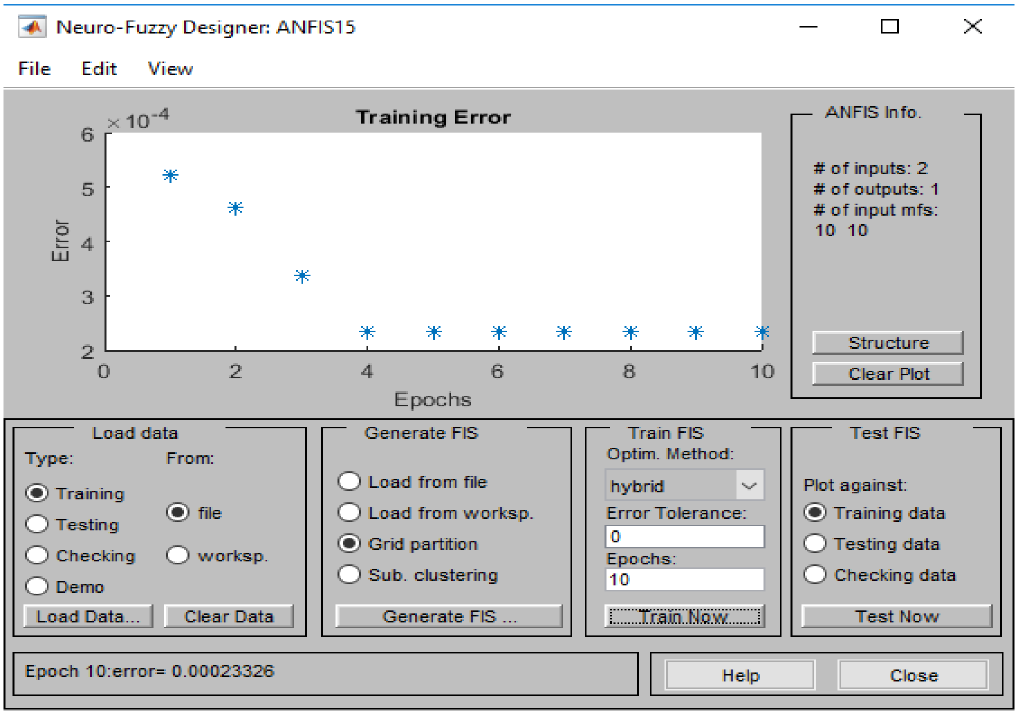


Fig. 4.1: SOC and OCV based on different temperatures

4.3.1. Implementation by ANFIS

As providing the data list based on the manufacturer technical specification, we proceeded to start our simulation in MATLAB. Firstly, by the command ‘anfisedit’ we import the aforementioned data file into the ANFIS in 2 different ways. Firstly, SOC and temperature as input and OCV as target and secondly, OCV and temperature as inputs and SOC as target. Our file includes 486 data set. For MF type ‘gaussmf’ is utilized. According to fig.4.2 We begin with the number of MFs for both temperature and OCV are 10 and 10 and start to train and test the ANFIS. In fig.4.2 the training process of the ANFIS based on 2 inputs and mfs 10:10 is shown. Afterwards, the testing process starts to evaluate the testing error based on the training data as is illustrated in fig.4.3.

Along the same line, after testing process by the help of rule viewer different values for inputs imported to analysis design performance. Based on the command 2 which can be seen in appendix, for each trial relevant output exported for MSE and visual evaluation as well. As a test, plots for different temperatures including those which are indicated in manufacturer technical data and a few temperatures between them are shown in fig. 4.5 and fig. 4.6.



# of inputs: 2
# of outputs: 1
# of input mfs:
10 10

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ● file ○ worksp.

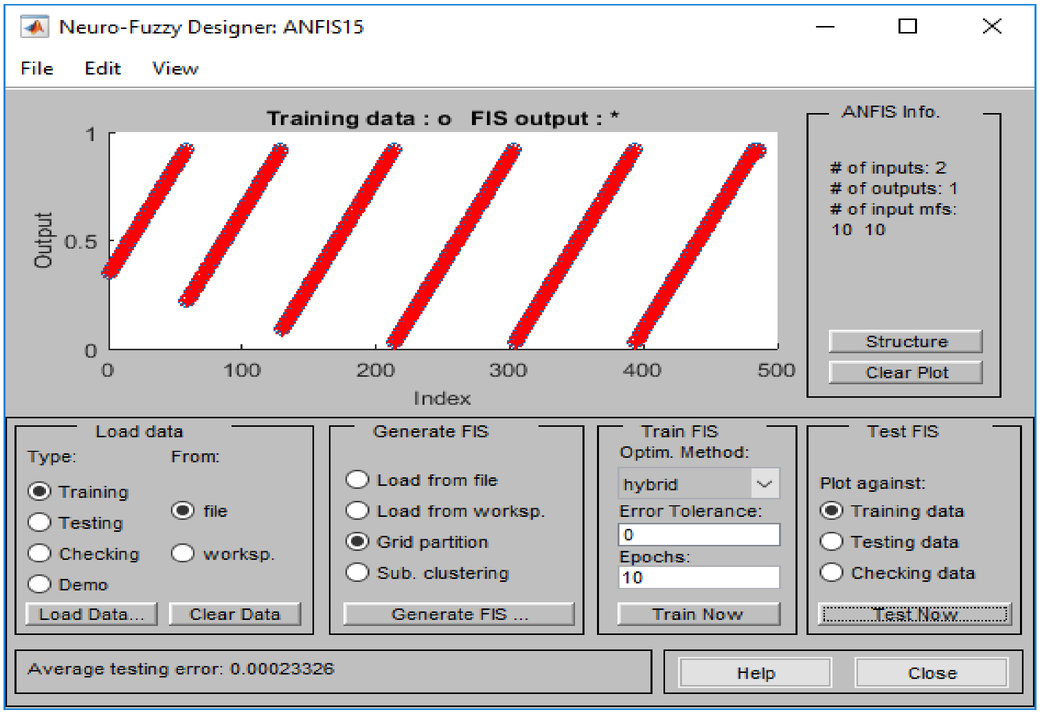
○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 10

Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 10:error= 0.00023326

Fig. 4.2: Training of first-order ANFIS with 10 and 10 MF per input



ANFIS Info.

# of inputs: 2
# of outputs: 1
# of input mfs:
10 10

Load data

Type:
● Training
○ Testing
○ Checking
○ Demo

From:
● file
○ worksp.

Generate FIS

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Train FIS

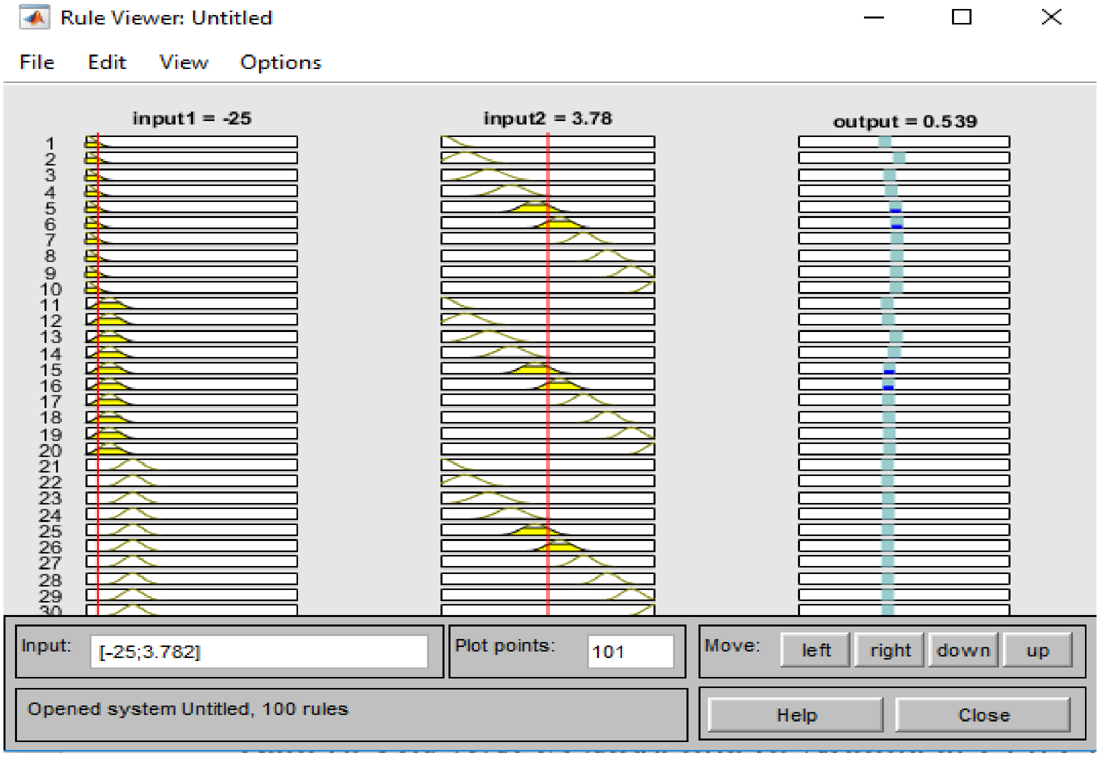
Optim. Method:
hybrid
Error Tolerance:
0
Epochs:
10

Test FIS

Plot against:
● Training data
○ Testing data
○ Checking data

Average testing error: 0.00023326

Fig. 4.3: Testing of first-order ANFIS with 10 and 10 MF per input



Input: [-25;3.782]

Plot points: 101

Move:

Opened system Untitled, 100 rules

Fig. 4.4: Rule viewer of the ANFIS with 10 and 10 MF per input

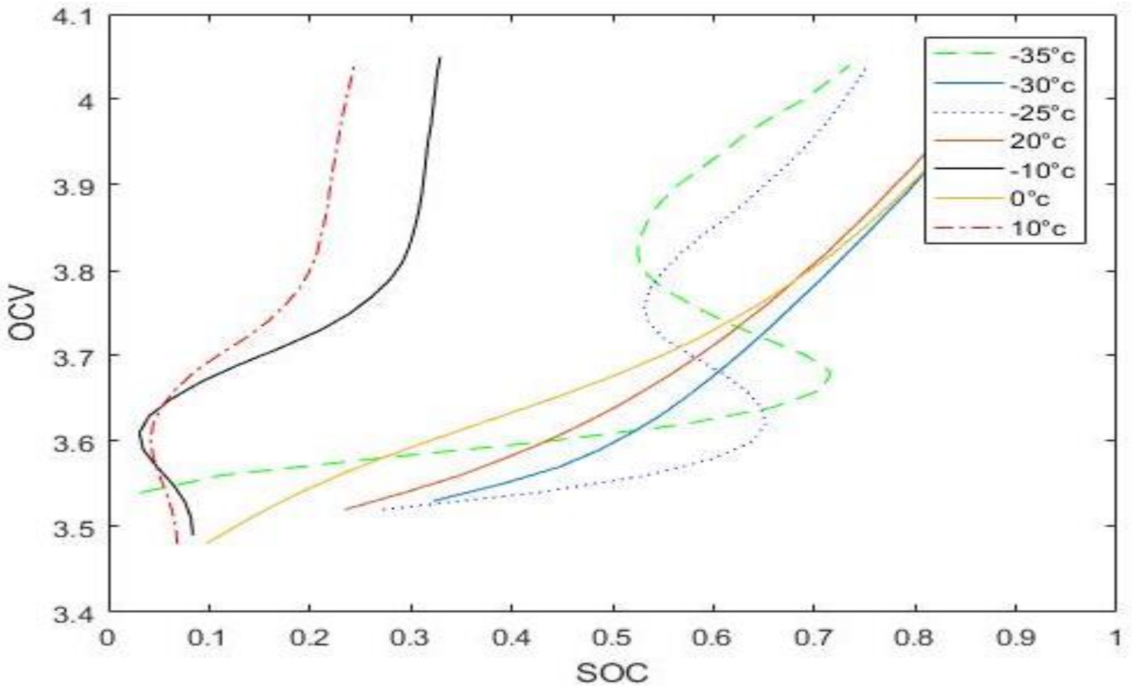


Fig. 4.5: ANFIS test 10:10 MFs VS training data for temperatures 10 to -35

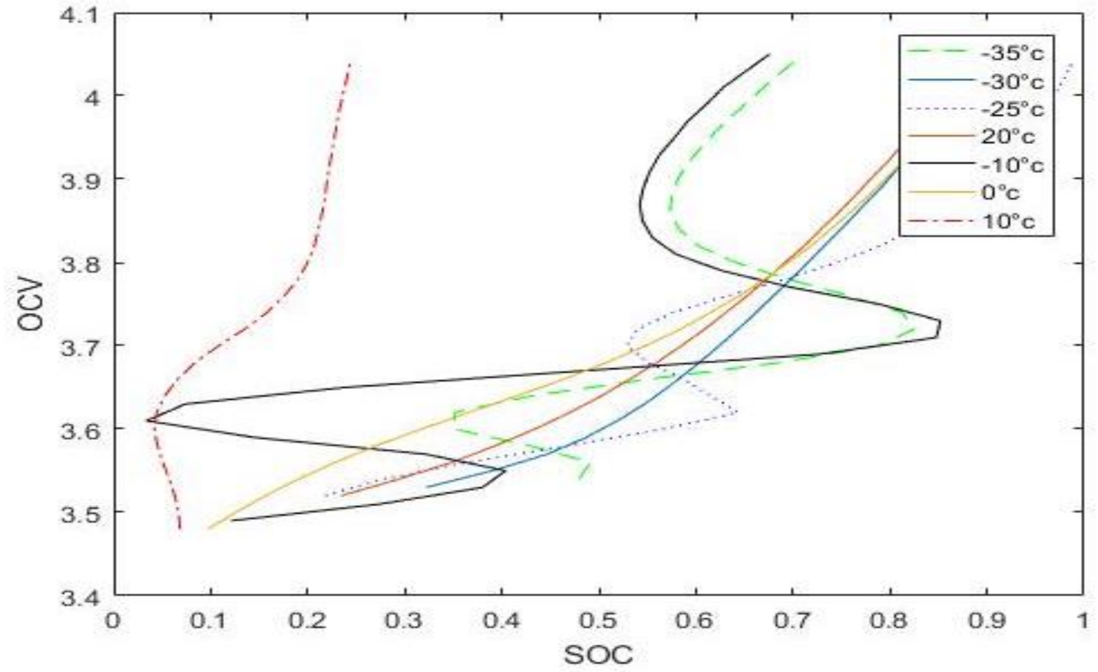


Fig. 4.6: ANFIS test 5:15 MFs VS training data for temperatures 10 to -35

As it is shown that by changing the number of MFs the change in output is made. After several implementations [23] it the end, the most appropriate MFs would be considered as 3 for temperature and 20 for open circuit voltage. Referring to section 3.2, the MFs make the partitioning over dataset. It is reasonable to indicate that because of the training data volume, the MFs of the SOC or OCV with 490 datasets shall be great deal more than the temperature with 6 datasets. Fig.4.7 shows the plot related to the training data and fig4.8 illustrate the test of ANFIS performance based on the training data.

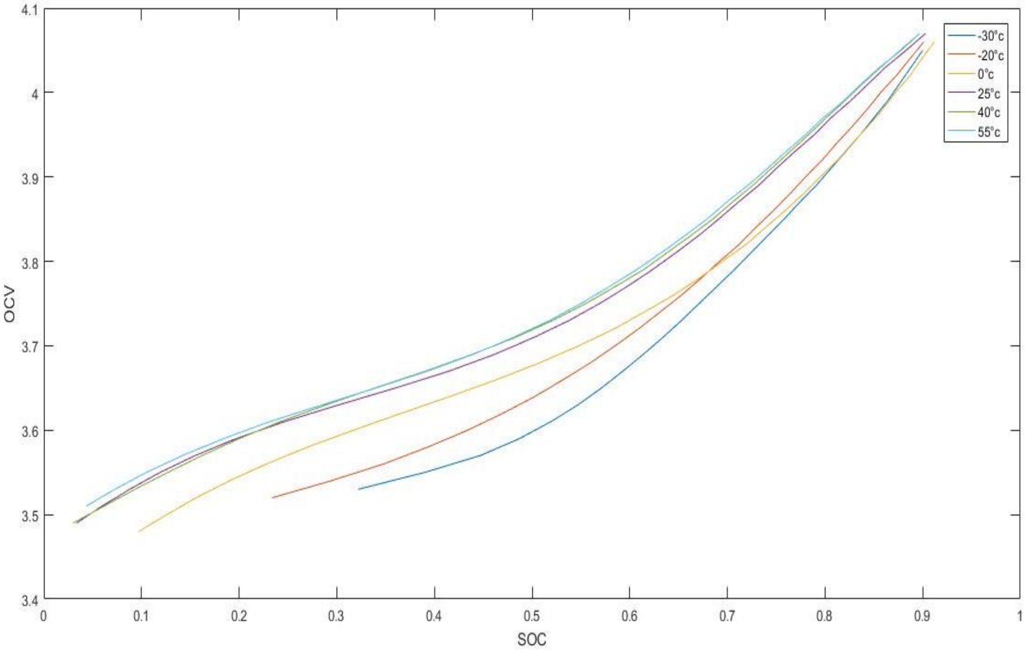


Fig. 4.7: ANFIS testing on points from the training dataset

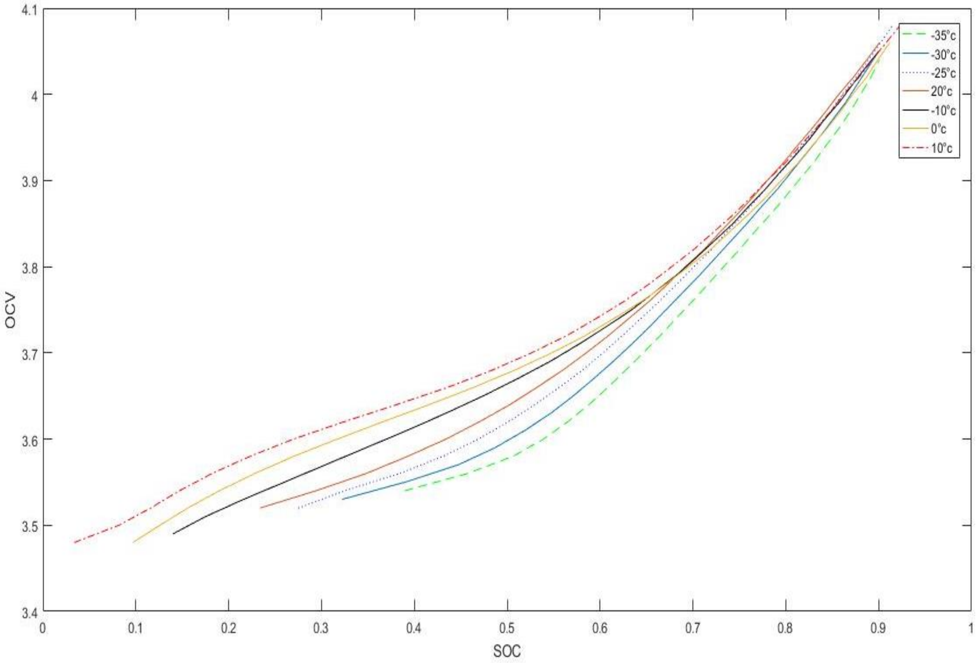


Fig. 4.8: ANFIS test 3:20 MFs VS training data

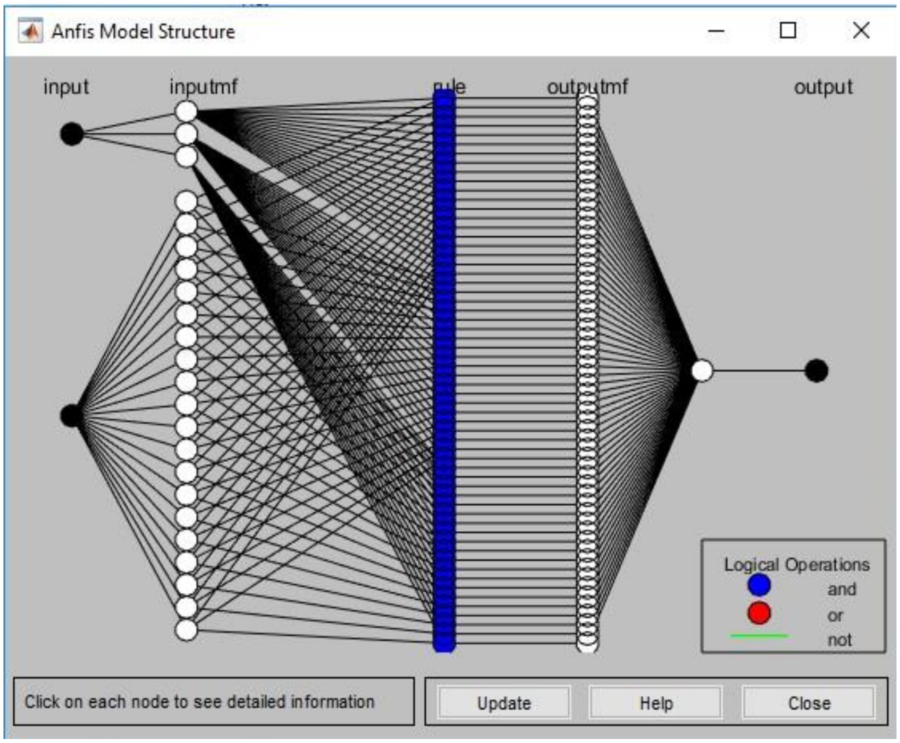


Fig. 4.9: ANFIS structure for 3:20 MFs

It should be mentioned that all the training data is based on below information. Fig.4.9 clarify the ANFIS structure which is modeled in this section.

-Number of nodes: 171

-Number of linear parameters: 180

-Number of nonlinear parameters: 46

-Total number of parameters: 226

-Number of training data pairs: 486

-Number of checking data pairs: 0

-Number of fuzzy rules: 60

4.3.2. ANFIS Result

The accuracy of SOC estimation is a crucial part of using Lithium-Ion batteries pack in energy storage to make a contribution with Battery Management System (BMS) to keep the storage system in a safe and efficient condition in order to reduce the depreciation and also increase the cycle of the batteries. In this part the author proceeds to design an intelligent system by using the manufacturer technical specification [23]. This data includes a nonlinear relation between OCV and SOC in 6 different temperatures. Since providing these values always faces limits for manufacturer, it is necessary to have aforementioned system to estimate the OCV and SOC for any arbitrary temperature. For this project we have an adaptive neuro-fuzzy inference system (ANFIS) to be trained by the manufacturer dataset. For set the ANFIS we used 490 datasets and choose Gaussian type for MFs. The value for Epochs is set as 10 to train the ANFIS 10 times for each number of member functions.

The first value for MFs was 10:10 for each inputs. The result is not satisfactory and does not have any similarity to the reference graphs and after changing the MFs for several time we figure out MFs depends on the number of dataset for each input. As we have only 6 different temperatures for the first input, and almost more than this amount for OCV, the number of the MFs for temperature shall be smaller. As a result, we try to train the system by smaller MFs for temperature. The value 5:15 is considered for temperature and OCV and we figure out the trend of SOC improves. It can be noted that for each step of training, ANFIS results are evaluated by 'evalfis' command which can be seen in appendix as command no.2. Although, the mean square error (MSE) in the first step (10:10) is reasonable, in subsequent steps it improves in a small range. Consequently, by trial and error we reached the MFs 3:20[23] to temperature and OCV or for temperature and SOC. It should be mentioned that as for each cell the level of voltage and temperature varies during all operation modes, this nonlinear tool is more accurate than traditional Coulomb counting at the pack level [23].

Regarding the less error during training for epoch 10 also less MSE for different implementations, it should be noted that considering the SOC and temperature as input and OCV as output has been found closer to actual data and more accurate. This result is shown in table 4.2.

Model	Epoch 10 Error	MSE
ANFIS_Input [OCV, Temperature] Output: SOC	0.0003052	0.0561
ANFIS_Input [SOC, Temperature] Output: OCV	0.00014615	0.0272

Table 4.2: MSE and Error for epoch 10 during ANFIs training methods

4.4. Implementation by ANN

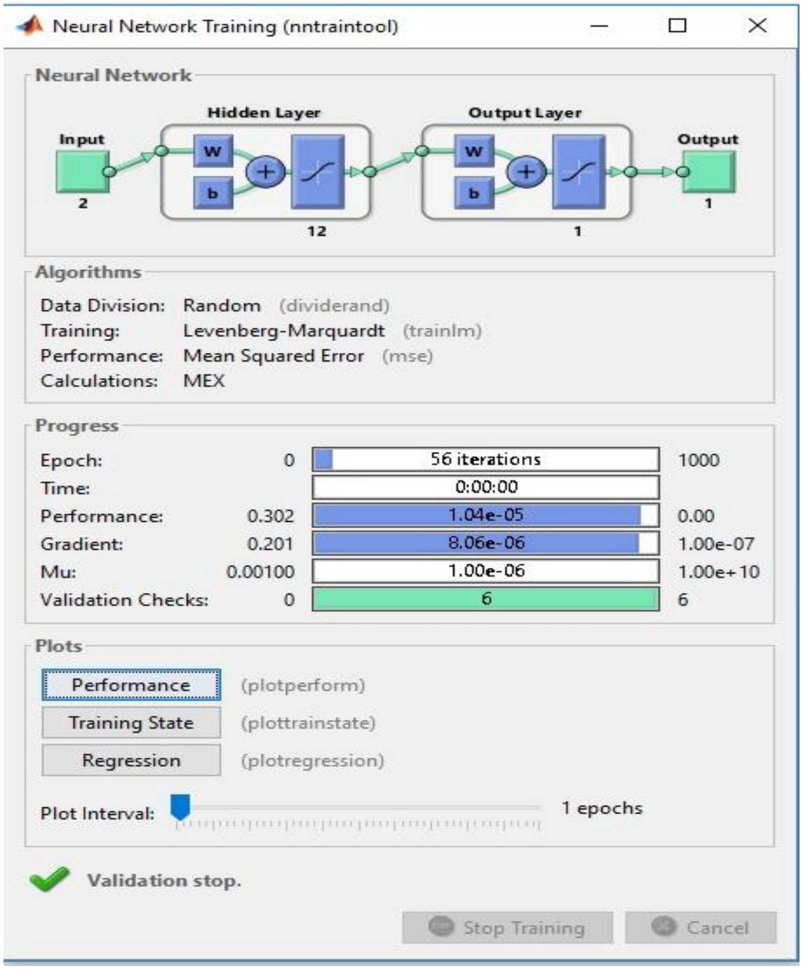
Along the same line, for the SOC estimation another model is introduced by the help of Artificial Neural Network (ANN). Since neural networks are self-learning and adaptive, they can deal with nonlinear systems easily and they are not supposed to consider any physical aspect of the system so they can be used for complex systems. As ANNs are self-learning tools, learning data plays an important rule for them and its amount depends on the system complexity. The estimation of SOC in batteries can be more efficient by the contribution of ANNs. Different techniques of ANN are conducted in academic researches and industrial project as well. In [36] a Back Propagation Neural Network (BPNN) is modelled for SOC estimation of an NIMH battery pack based on relationship between the open circuit voltage (OCV) and SOC based on the 4 situations of the battery, charging, discharging, laying aside after charging or laying aside after discharging. In [37] another BPNN is introduced to for SOC estimation of LiFePO4 batteries. In [38] an Adaptive Wavelet Neural Network (AWNN) provide an accurate SOC estimation for Lithium batteries. [39] draws an analogy between the result of an Elman Neural Network (ENN) and BPNN for SOC estimation. In [40] a Radial Basis Function Neural Network (RBFNN) is performed on a Lead-Acid battery.

The disadvantages of all aforementioned implementations is that the battery characteristic such as battery behavior during different temperatures which is provided by the manufacturer in not considered. Besides, battery is analyzed in pack level instead of cell level. Pack level consideration means that all cells charge and discharge at the same situation with the equal voltages. As a result, the accuracy of the estimation decreases.

In this project a multi-layer artificial neural network (ANN) is utilized for SOC estimation Lithium-ion battery pack based on the characteristics of SOC which varies with OCV at different temperatures. A multi-layer artificial neural network (ANN) typically consists of an input layer, one or more hidden layers and an output layer, [20].

This section is also handled by MATLAB. There are two choices as “nntool: and “nftool” to create an ANN model. Any model that can be performed by the “nftool” it can also be conducted by the “nntool”, but there are some models can be done by the “nntool” but not by the “nftool”. On the other hand, since the “nntool” is more general, there are more parameters to be adjusted for running the neural network to achieve more efficient estimation. Network type is used for this model is a 2 layer "Feed-forward backprop" with 10 neurons. "TRAINLM" is considered as training function and "LEARNFGM" is set for adaption learning function.

Data set have been already used in previous section to train the ANFIS is used for ANN. By taking advantage from 'nntool, the desired ANN is provided. In this section there are two implementations exactly according to the previous section. One ANN for SOC and temperature as inputs and OCV as output, another one based on OCV and temperature as inputs and SOC as output. Fig.4.9, illustrates the training process for methods which take advantage from 2-layer training technique. Minimum the value of gradient coefficient better will be training and testing of networks. From fig.4.12 which shows plot training state of the neural network it can be seen that gradient value goes on decreasing with increase in number of epochs. Fig.4.11 also shows the performance of the training process which illustrate the accuracy of the training.



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division:	Random	(dividerand)
Training:	Levenberg-Marquardt	(trainlm)
Performance:	Mean Squared Error	(mse)
Calculations:	MEX	

Progress

Epoch:	0	56 iterations	1000
Time:		0:00:00	
Performance:	0.302	1.04e-05	0.00
Gradient:	0.201	8.06e-06	1.00e-07
Mu:	0.00100	1.00e-06	1.00e+10
Validation Checks:	0	6	6

Plots

Performance (plotperform)

Training State (plottrainstate)

Regression (plotregression)

Plot Interval: 1 epochs

✓ Validation stop.

Fig. 4.10: ANN training for OCV and temperature as input and SOC as output

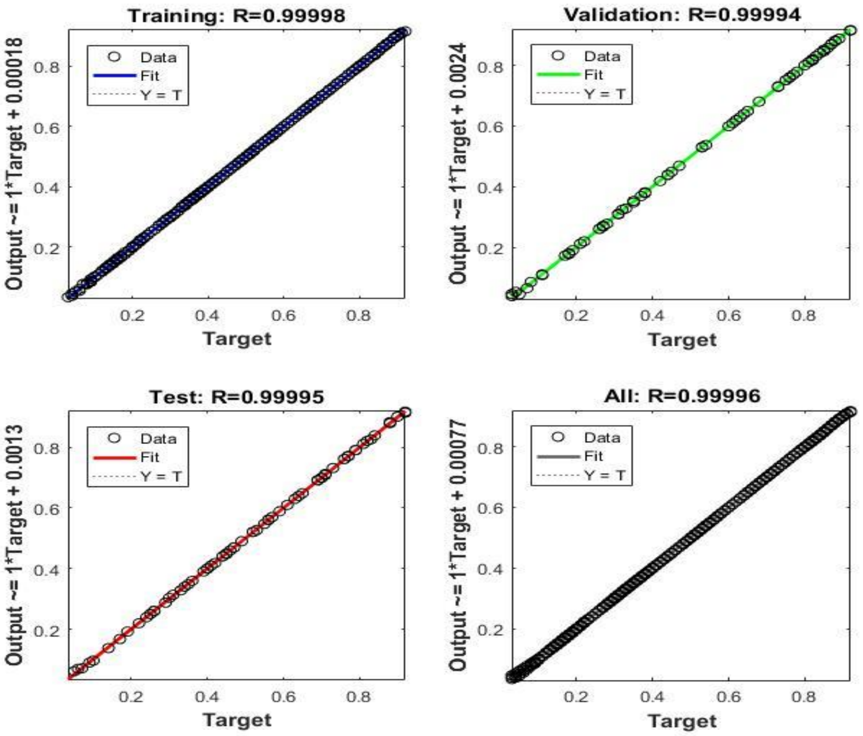


Fig. 4.11: ANN training, regression and validation performance of SOC estimation (plotperform)

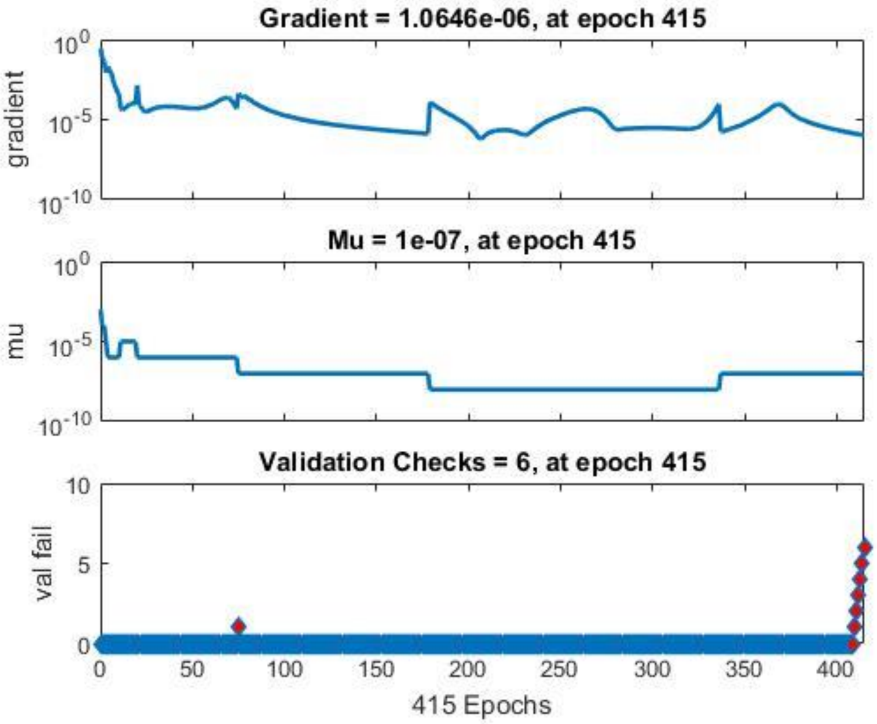


Fig. 4.12: ANN training state for SOC estimation (plottrainstate)

4.4.1. ANN Result

Table 4.3 shows the calculated MSE between ANN results and actual data which are great deal less than the MSE achieved in both ANFIS implementations. As it is shown, the MSE for the ANN that SOC and temperature are inputs and OCV is output is closer to the actual data extracted from the curve by curve fitting technique.

Model	**MSE**
ANN Input [OCV, Temperature] Output: SOC	0.0017
ANN Input [SOC, Temperature] Output: OCV	0.0013

Table 4.3: MSE result for ANN implementations

Chapter 5: Conclusion and Future Work

To gain an optimal energy storage system there are 2 separate point shall be considered. The first is an accurate prediction for load demand and the second is precise estimation of SOC. For the first point, by the help of artificial neural networks there is a model introduced according to the daily based data set for 2 years load consumption to help the energy storage covers the demand in case of need. Based on the graphical and also mathematical comparison, the ANN result is close to the actual data but the system needs more accuracy for more optimization of energy storage system.

Besides, for the second point, there are 2 separate models introduced based on ANFIS and ANN. Each technique has 2 implementations. MSE for each implementation is shown in table 5.1. Although the result of ANFIS for SOC estimation is accurate enough, as it is crystal clear, the MSE for the ANN in which the SOC and temperature are input and OCV is output has been found the most accurate technique among all 4 implementations and can be considered as a substitution for the ANFIS method. With all the pieces in place, the achievement of lower MSE results in more safety, lower depreciation, longer cycle life and more reliable lithium-ion battery packs for the energy storage system.

Model	**MSE**
ANN Input [OCV, Temperature] Output: SOC	0.0017
ANN Input [SOC, Temperature] Output: OCV	0.0013
ANFIS Input [OCV, Temperature] Output: SOC	0.0561
ANFIS Input [SOC, Temperature] Output: OCV	0.0272

Table 5.1: MSE result for ANFIS and ANN implementation

5.1. Future Work

This project is a concept that adaptive neuro fuzzy inference system and artificial neural network is able to design a model for accurate estimation of SOC and load prediction in energy storage systems such as batteries. This concept will be used to develop this project to a hybrid model which is combined ANN and ANFIS for the estimation of lithium-ion batteries based on cell level and in accordance with the characteristics supplied by the manufacturer to achieve more accurate model for SOC estimation. Moreover, this hybrid design can be implemented for load prediction and as a result, the desired ES system is aimed to be optimized and more efficient to meet the grid satisfaction accordingly.

Appendices

A.1. MATLAB Command for MSE calculation, Section 4.1

```
function rmse (data, estimate)
r = sqrt (sum((data(:)-estimate (:)).^2)/numel(data))
rmse (data, estimate)
```

A.2. MATLAB Command for ANN model for load prediction, Section 4.1

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 08-
Nov-2017 20:55:23.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = 8xQ matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = 6xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number
of timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset =
[6.865714286;6.72;6.855;9.009375;10.815;9.7575;10;0];
x1_step1.gain =
[0.058318753645408;0.0601684717208183;0.0630566721841255;0.06332
24497872761;0.0528890651857728;0.0619290911905868;0.066666666666
6667;2];
x1_step1.ymin = -1;

% Layer 1
```

```
b1 = [1.6688018676241783;2.3198783867080759;-
0.9073173102144706;1.5147867588231203;-
0.61094623012872185;0.13253425680472447;-0.69014178045747598;-
0.061421005767732073;-0.82699145169695554;-0.84331240366466065;-
0.81066523707402371;-1.5029524469273532;-1.5593522942131139;-
2.1624476614513788;1.3450576062053876];
IW1_1 = [0.32789452536458152 0.095085939651819662
0.62024469435877028 -1.2393150963843409 0.2316578383590015
0.15869111270377068 1.3803290274258151 -
0.53755326600501963;0.65932806281063461 -0.45955301146081573
0.12571353390004442 -0.39072881011712618 0.14756201632783428 -
0.13681790047612469 0.31140901222939388 -
1.1113551640937813;0.28067554516028975 -0.16453820151899948
0.36665844241283474 -0.51561691423615463 -0.28311424579562178
0.3786815139504171 0.41094228447333647 -0.16292982158951397;-
0.82917957837258949 1.0198016999896444 1.3690699031195674
0.70796540936189023 0.34796591433475532 -0.15683051836332487 -
0.76930927306682517 -1.0712978537472446;1.5980240540686903 -
1.3159219935059354 0.0092969371002519161 0.46595359945467951
1.2230826190429673 0.092469701929221812 0.23292337098868912
0.077651877690495336;-0.86187490989861615 0.211311821302545
0.58039051025200661 -0.18685659671725732 0.81950000287587044
1.2256678984989287 0.61954317417439819
0.41386555904917 51;1.0269176014987613 1.076681432388424
0.047026929083136777 -0.6964628193107365 0.2459177218498558
0.37365549839103956 -1.9853302530444037
0.15144310272564843;0.33232751079822942 -0.66649528051247819 -
0.27654415396655008 -0.43711767409356317 0.33573395004301132
0.95420550940422277 0.06225653410947888 0.15842739982699358;-
0.37735483217856858 -1.6254945586656979 -0.75733199903514614 -
0.6391044073965646 0.60363613613074596 1.7748566264539045 -
0.18798050254408732 -0.043607488715741 32;-0.28626216307245167
0.43692533567949915 0.042760192068174427 0.41920298730326599 -
0.34841559560979324 0.096438986619627479 -0.80571883869194671 -
1.4794887654338584;-0.33499469456573622 1.2848321060625669 -
0.42957672129140684 0.87292174588339722 0.65876282858508961 -
0.18097638133660163 0.08952471990758 8257
0.044287339508 90907;0.36871806101329196 -1.1381388907183185
0.071140948536311865 1.2838953513829168 1.9351239941035807
1.8333333842934749 -0.64985363350308234 0.91532428699310842;-
0.81349408360249531 0.7026504424520551 0.19513974495408595
0.4005369478502615 0.40196408011863832 0.30152400614651548 -
1.2867934619468324 1.4853279428338955;-1.3186553914391661 -
0.20364498962248498 -0.27859230379784972 0.041214426350606195 -
0.29647710210664002 -0.45062046338515915 -0.66988416971421 96 -
0.67215424121377532;-0.079000443231553169 1.0572815359148957
0.61929417068138182 0.25996219254625674 0.47855872780053832 -
0.053560255331131137 -0.73598884353439031 -0.6805327546786416];

% Layer 2
b2 =
[0.46868642536080884;0.24778808839980054;0.965088203611O312;-
0.22955128360028479;-0.80932987485693408;0.61640003227954288];
LW2_1 = [-0.1336120059432466 -0.51059944546397862
0.10783449545159116 0.2155571583250136 -0.042557760158794068
0.14554379740495765 0.1219134158401065 0.68620021358032945 -
0.32794668620962825 0.1812602835347821 0.47215928513227345 -
0.081324426052363882 -0.11652115589090627 -0.064852880605638638
-0.1599664553719054;-0.19910104395101569 -0.20880496375242874
0.11997132739626823 0.25656746514149209 -0.093841971995448109
0.1937161059334741 0.18248123084226198 0.63520465870435872 -
0.38675279586029121 0.12003243582434252 0.45706684132007397 -
0.095433741882829187 -0.1556832395710416 -0.043543189189657827 -
0.17866343492180739;-0.22685063047939952 -1.1136581511354087 -
0.043753836459923583 0.42759171221292935 -0.059224425407783422
0.15976425628701671 0.059422243730743868 0.868662628018812 -
0.44754651066305362 0.22787534103068563 0.49324616625491363 -
0.13749736453133587 -0.19801671201906529 -0.03903692175765569 -
0.33969699284368321;-0.42049309765419596 -0.19944910035750993 -
0.64238001298227176 0.60874606808107812 -0.0065706753696241071
0.22345357319630185 -0.11397200319670189 1.0458128674644911 -
0.47401144112892768 0.29322620264298976 0.40031547445706622 -
0.13481327358067327 -0.15104720494843965 -0.13962863556273516 -
0.5574150743445091;-0.24869761841946544 0.34487296424587721 -
0.59947720686182449 0.46828101095138863 0.10924066579671421
0.22326180171706839 -0.058499709786872078 0.76188153180104046 -
0.32024116571535999 0.19986832222284842 0.26149607542948494 -
0.15696279607942848 -0.068516788937278597 -0.12683730104540794 -
0.38500250639983224;-0.29499450261544347 -0.71104564773195789 -
0.28226308928057431 0.57282252772719633 0.20044265886159573
0.19213936102318471 -0.034010835594407714 1.2179315872200283 -
0.51413076633643207 0.37581627560262365 0.36395214874895687 -
0.2778990535510581 -0.083512160198936744 0.0039964447557942415 -
0.60106724550138113];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain =
[0.058318753645408;0.0601684717208183;0.0630566721841255;0.06332
24497872761;0.0528890651857728;0.0619290911905868];
y1_step1.xoffset =
[6.865714286;6.72;6.855;9.009375;10.815;9.7575];
```

```
% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
```

```
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

A.3. MATLAB Command for ANFIS evaluation, Section 5.3.1

```
a=readfis(‘ANFIS1010’)
b=evalfis([Input_Test],a)
```

A.4. MATLAB Command for ANN Feed-forward backprop, Section 5.4

```
function createfigure(X1, YMatrix1, X2, Y1, YMatrix2, X3, X4,
YMatrix3, X5, YMatrix4, X6)
%CREATEFIGURE(X1, YMATRIX1, X2, Y1, YMATRIX2, X3, X4, YMATRIX3,
X5, YMATRIX4, X6)
%  X1:  vector of x data
%  YMATRIX1:  matrix of y data
%  X2:  vector of x data
%  Y1:  vector of y data
%  YMATRIX2:  matrix of y data
%  X3:  vector of x data
%  X4:  vector of x data
%  YMATRIX3:  matrix of y data
%  X5:  vector of x data
%  YMATRIX4:  matrix of y data
%  X6:  vector of x data

%  Auto-generated by MATLAB on 08-Nov-2017 20:10:40

% Create figure
figure1 =
figure('Tag','TRAINING_PLOTREGRESSION','NumberTitle','off',...
    'Name','Neural Network Training Regression (plotregression),
Epoch 415, Validation stop.');

% Create subplot
subplot1 = subplot(2,2,1,'Parent',figure1);
hold(subplot1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(X1,YMatrix1,'Parent',subplot1);
set(plot1(1),'DisplayName','Y = T','LineStyle',':','Color',[0 0
0]);
set(plot1(2),'DisplayName','Fit','LineWidth',2,'Color',[0 0 1]);

% Create plot
plot(X2,Y1,'Parent',subplot1,'DisplayName','Data','Marker','o',.
..
    'LineStyle','none',...
    'Color',[0 0 0]);

% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);

% Create title
title('Training: R=0.99998','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('Output ~= 1*Target +
0.00018','FontWeight','bold','FontSize',12);

% Uncomment the following line to preserve the X-limits of the
axes
% xlim(subplot1,[0.03 0.92]);
% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(subplot1,[0.03 0.92]);
box(subplot1,'on');
axis(subplot1,'square');
% Create legend
legend1 = legend(subplot1,'show');
set(legend1,'Location','northwest');

% Create subplot
subplot2 = subplot(2,2,2,'Parent',figure1);
hold(subplot2,'on');
```

```
% Create multiple lines using matrix input to plot
plot2 = plot(X1,YMatrix2,'Parent',subplot2);
set(plot2(1),'DisplayName','Y = T','LineStyle',':','Color',[0 0
0]);
set(plot2(2),'DisplayName','Fit','LineWidth',2,'Color',[0 1 0]);

% Create plot
plot(X3,Y1,'Parent',subplot2,'DisplayName','Data','Marker','o',.
..
    'LineStyle','none',...
    'Color',[0 0 0]);

% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);

% Create title
title('Validation:
R=0.99994','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('Output ~= 1*Target +
0.0024','FontWeight','bold','FontSize',12);

% Uncomment the following line to preserve the X-limits of the
axes
% xlim(subplot2,[0.03 0.92]);
% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(subplot2,[0.03 0.92]);
box(subplot2,'on');
axis(subplot2,'square');
% Create legend
legend2 = legend(subplot2,'show');
set(legend2,'Location','northwest');

% Create subplot
subplot3 = subplot(2,2,3,'Parent',figure1);
hold(subplot3,'on');

% Create multiple lines using matrix input to plot
plot3 = plot(X4,YMatrix3,'Parent',subplot3);
set(plot3(1),'DisplayName','Y = T','LineStyle',':','Color',[0 0
0]);
```

```
set(plot3(2),'DisplayName','Fit','LineWidth',2,'Color',[1 0 0]);

% Create plot
plot(X5,Y1,'Parent',subplot3,'DisplayName','Data','Marker','o',.
..
    'LineStyle','none',...
    'Color',[0 0 0]);

% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);

% Create title
title('Test: R=0.99995','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('Output ~= 1*Target +
0.0013','FontWeight','bold','FontSize',12);

% Uncomment the following line to preserve the X-limits of the
axes
% xlim(subplot3,[0.0337333083488191 0.92]);
% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(subplot3,[0.0337333083488191 0.92]);
box(subplot3,'on');
axis(subplot3,'square');
% Create legend
legend3 = legend(subplot3,'show');
set(legend3,'Location','northwest');

% Create subplot
subplot4 = subplot(2,2,4,'Parent',figure1);
hold(subplot4,'on');

% Create multiple lines using matrix input to plot
plot4 = plot(X1,YMatrix4,'Parent',subplot4);
set(plot4(1),'DisplayName','Y = T','LineStyle',':','Color',[0 0
0]);
set(plot4(2),'DisplayName','Fit','LineWidth',2,'Color',[0.4 0.4
0.4]);

% Create plot
plot(X6,Y1,'Parent',subplot4,'DisplayName','Data','Marker','o',.
..
```

```
    'LineStyle','none',...
    'Color',[0 0 0]);

% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);

% Create title
title('All: R=0.99996','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('Output ~= 1*Target +
0.00077','FontWeight','bold','FontSize',12);

% Uncomment the following line to preserve the X-limits of the
axes
% xlim(subplot4,[0.03 0.92]);
% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(subplot4,[0.03 0.92]);
box(subplot4,'on');
axis(subplot4,'square');
% Create legend
legend4 = legend(subplot4,'show');
set(legend4,'Location','northwest');

% uicontrol currently does not support code generation, enter
'doc uicontrol' for correct input syntax
% In order to generate code for uicontrol, you may use GUIDE.
Enter 'doc guide' for more information
% uicontrol(...);
```

References

[1] EPRI-DOE Handbook of Energy Storage for Transmission and Distribution Applications, 1001834, Final Report, December 2003, Cosponsor U. S. Department of Energy 1000 Independence Ave., S.W. Washington, DC 20585 Project Manager I. Gyuk

[2] http://energystoragesense.com/pumped-hydroelectric-storage-phs/

[3]https://www.newcivilengineer.com/future-tech/going-underground-energy-storage/10006946.article

[4] Heussen, K., You, S., Biegel, B., Hansen, L. H., & Andersen, K. B. (2012). Indirect Control for Demand Side Management – A Conceptual Introduction. Innovative Smart Grid Technologies. Berlin: IEEE.

[5] R. Schlögl, "Chemical Energy Storage", Berlin: De Gruyter, 2012.

[6] C. Higman and M. van der Burgt, "Gasification", USA: Gulf Professional Publishing, 2003.

[7] F. Joensen, P. Højlund Nielsen and M. Sørensen, "Biomass to green gasoline and power", Biomass Conv. Bioref., vol. 1, pp. 85-90, 2011.

[8] J. Rostrup-Nielsen, P. Højlund Nielsen, F. Joensen and J. Madsen, "Polygeneration – Integration of Gasoline Synthesis and IGCC Power Production Using Topsøe's TIGAS Process", in Proc. Risø Int. Energy Conference, 2007, pp. 56-68.

[9] J. Ahrenfeldt, T. Thomsen, U. Henriksen and L. Clausen, "Biomass gasification cogeneration– A review of state of the art technology and near future perspectives", Appl. Therm. Eng., vol. 50, pp. 1407-1417, 2013.

[10] P. Sørensen, N. Cutululis, A. Vigueras-Rodríguez, L. Jensen, J. Hjerrild and M. M. H. Donovan, "Power fluctuations from large wind farms", IEEE Transactions on Power Systems, vol. 22, pp. 958-965, 2007

[11]Chemical energy storage, Christensen, Jakob Munkholt; Hendriksen, Peter Vang; Grunwaldt, Jan-Dierk; Jensen, Anker Degn, Published in: DTU International Energy Report 2013, Publication date:2013

[12] J. Christensen, P. Jensen and A. Jensen, “Effects of Feed Composition and Feed Impurities in the Catalytic Conversion of Syngas to Higher Alcohols over Alkali-Promoted Cobalt–Molybdenum Sulfide”, Ind. Eng. Chem. Res., vol. 50, pp. 7949-7963, 2011.

[13] C.-W. Chen, "Modeling and control for nonlinear structural systems via a NN-based approach," Expert Systems with Applications, vol. 36, pp. 4765-4772, 2009.

[14] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," Neural networks, vol. 4, pp. 565-588, 1991.

[15] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," Neural Networks, IEEE Transactions on, vol. 1, pp. 4-27, 1990.

[16] http://www.sciencedirect.com/topics/neuroscience/axon

[17] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to’modeling and control,” IEEE Trans. Syst., Man, Cybern., vol. 15, pp. 116132, 1985.

[18] W. X. Shen, C. C. Chan, E. W. C. Lo and K. T. Chau, "Adaptive Neuro-Fuzzy Modeling of Battery Residual Capacity for Electric Vehicles", IEEE Transactions on Industrial Electronics, Volume 49, No. 3, June 2002.

[19] H. Fekry, "A Novel Approach to Estimate The State of Charge of Batteries Using Adaptive Neuro Fuzzy Inference System ", M. Sc. Thesis submitted to Graduate Studies in Faculty of Engineering, Cairo University, Giza, Egypt, Nov., 2012.

[20] ANFIS: Adap tive-Ne twork-Based Fuzzy Inference System Jyh-Shing Roger Jang, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 23, NO. 3, MAYIJUNE 1993

[21] Mamdani Model based Adaptive Neural Fuzzy Inference System and its Application in Traffic Level of Service Evaluation, Yuanyuan Chai, Limin Jia, Zundong Zhang State Key Lab of Rail Traffic Control and Safety Beijing Jiaotong University Beijing, China 978-0-7695-3735-1/09 © 2009 IEEE DOI 10.1109/FSKD.2009.76

[22] Short-Term Load Forecasting Using Artificial Neural Network, Muhammad Buhari Bayero University, Kano- Sunusi Sani Adamu Bayero University, Kano, Manuscript received September

26, 2011: revised October 09, 2011 Muhammad Buhari is with Bayero University, Kano, P.M.B 3011 Kano, Nigeria, as an Assistant Lecturer, mobile: +2348065533305; muhdbuhari@gmail.com. Sanusi Sani Adamu is also with Bayero University, Kano, P.M.B 3011 Kano, Nigeria, as a senior lecturer; adamus1664@buk.edu.ng

[23] Accuracy improvement of SOC estimation in lithium-ion batteriesMohamed A. Awadallah, Bala VenkateshCentre for Urban Energy, Ryerson University, Toronto, Canada, Journal of Energy Storage 6 (2016) 95–104, Contents lists available at ScienceDirect, Journal of Energy Storage journa l home page: www.e lsevier.com/loca te/est

[24] State of charge estimation is of great significance for the safe operation of a lithium-ion battery (LIB) pack.- Zhihao Yu 1, Ruituo Huai 2 and Linjing Xiao 1, ISSN 1996-1073

[25] M. Corno, N. Bhatt, S.M. Savaresi, M. Verhaegen, Electrochemical model-based state of charge estimation for li-ion cells, IEEE Trans. Control Syst. Technol. 23 (1) (2015) 117–127.

[26] Z. Chen, Y. Fu, C.C. Mi, State of charge estimation of lithium-ion batteries in electric drive vehicles using extended Kalman filtering, IEEE Trans. Veh. Technol. 62 (3) (2013) 1020–1030.

[27] R. Xiong, H. He, F. Sun, K. Zhao, Evaluation on state of charge estimation of batteries with adaptive extended Kalman filter by experiment approach, IEEE Trans. Veh. Technol. 62 (1) (2013) 108–117.

[28] H. He, R. Xiong, X. Zhang, F. Sun, J. Fan, State-of-charge estimation of the lithium-ion battery using and adaptive extended Kalman filter based on an improved Thevenin model, IEEE Trans. Veh. Technol. 60 (4) (2011) 1461–1469.

[29] A.A. Hussein, Capacity fade estimation in electric vehicle li-ion batteries using artificial neural networks, IEEE Trans. Ind. Appl. 51 (3) (2015) 2321–2330.

[30] I.H. Li, W.Y. Wang, S.F. Su, Y.S. Lee, A merged fuzzy neural network and its applications in battery state-of-charge estimation, IEEE Trans. Energy Convers. 22 (3) (2007) 697–708.

[30] H. Gholizade-Narm, M. Charkhgard, Lithium-ion battery state of charge estimation based on square-root unscented Kalman filter, IET Power Electron. (2013) 1833–1841.

[31] M.F. Tsai, Y.Y. Peng, C.S. Tseng, N.S. Li, Modeling and estimation of state of charge for

lithium-ion batteries using ANFIS architecture, Hangzhou, China, Proc. IEEE International Symposium on Industrial Electronics, 28–312012, pp. 863–868.

[32] M. Gholizadeh, F.R. Salmasi, Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model, IEEE Trans. Ind. Electron. 3 (2014) 1335–1344.

[33] M.W. Cheng, Y.S. Lee, M. Liu, C.C. Sun, State-of-charge estimation with aging effect and correction for lithium-ion battery, IET Electr. Syst. Transp. 5 (2) (2015) 70–76.

[34] C. Zhang, L.Y. Wang, X. Li, W. Chen, G.G. Yin, J. Jiang, Robust and adaptive estimation of state of charge for lithium-ion batteries, IEEE Trans. Ind. Electron. 62 (8) (2015) 4948–4957.

[35] BATTERY STATE OF CHARGE ESTIMATION USING AN ARTIFICIAL NEURAL NETWORK, Mahmoud Ismail, Rioch Dlyma, Ahmed Elrakaybi, Ryan Ahmed, and Saeid Habibi Mechanical Engineering Department McMaster University, Hamilton, Canada, 2017

[36] B. Sun and L. Wang, "The SOC estimation of NIMH battery pack for HEV based on BP neural network," in Intelligent Systems and Applications, 2009. ISA 2009. International Zorkshop on, 2009, pp. 1-4.

[37] W.-Y. Chang, "State of charge estimation for LiFePO4 battery using artificial neural network," INTERNATIONAL REVIEW OF ELECTRICAL ENGINEERING-IREE, vol. 7, pp. 5874-5880, 2012.

[38] F. Zhou, L. Wang, H. Lin, and Z. Lv, "High accuracy state-ofcharge online estimation of EV/HEV lithium batteries based on Adaptive Wavelet Neural Network," in ECCE Asia Downunder (ECCE Asia), 2013 IEEE, 2013, pp. 513-517.

[39] S. Qingsheng, Z. Chenghui, C. Naxin, and Z. Xiaoping, "Battery state-of-charge estimation in electric vehicle using elman neural network method," in Control Conference (CCC), 2010 29th Chinese, 2010, pp. 5999-6003.

[40] L. Yanwei, Z. Kegang, H. Xiangdong, and P. Feng, "A new method based on RBFNN in SOC estimation of HEV battery," in Control Conference (CCC), 2010 29th Chinese, 2010, pp. 49234927.

Glossary

-ANFIS:	*Adaptive Neuro Fuzzy Inference System*
-ANN:	*Artificial Neural Network*
-AWNN:	*Adaptive Wavelet Neural Network*
-BPNN:	*Back Propagation Neural Network*
-CASH:	*Compressed air energy storage*
-ENN:	*Elman Neural Network*
-ES:	*Energy Storage System*
-DS:	*Demand Response*
-GAS:	*Grid Angular Stability*
-GFS:	*Grid Frequency Excursion Suppression*
-GVS:	*Grid Voltage Stability*
-LPQ:	*Long Duration Power Quality*
-LS3:	*3-hr Load Shifting*
-LS10:	*10-hr Load Shifting*
-MFR:	*Manufacturer*
-NNFT:	*Neural Network Fitting Tool*
-NN:	*Neural Network*
- MSE:	*Mean Square Error*
-OCV:	*Open Circuit Voltage*
-PHES:	*Pump hydro energy storage*
-RBFNN:	*Radial Basis Function Neural Network*

-RC:	*Regulation Control*
-SNG:	*Synthetic Natural Gas*
-SOC:	*State of Charge*
-SR:	*Spinning Reserve*