2014-04-30

Short-Term Peak Demand Forecasting Using an Artificial Neural Network with Controlled Peak Demand Through Intelligent Electrical Loading

Jason L. Grant
University of Miami, jgrant@miami.edu

UNIVERSITY OF MIAMI

SHORT-TERM PEAK DEMAND FORECASTING USING AN ARTIFICIAL NEURAL NETWORK WITH CONTROLLED PEAK DEMAND THROUGH INTELLIGENT ELECTRICAL LOADING

By

Jason Lee Grant

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2014

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

SHORT-TERM PEAK DEMAND FORECASTING USING AN ARTIFICIAL NEURAL NETWORK WITH CONTROLLED PEAK DEMAND THROUGH INTELLIGENT ELECTRICAL LOADING

Jason Lee Grant

Approved:

____
Shihab Asfour, Ph.D.
Professor and Associate Dean of the
College of Engineering

____
Moataz Eltoukhy, Ph.D.
Assistant Professor of Kinesiology
and Sport Sciences

____
Saman Aliari Zonouz, Ph.D.
Assistant Professor of Electrical and
Computer Engineering

____
Khaled Abdel-Rahman, Ph.D.
Director of Academic
Computing of the College of
Engineering

____
M. Brian Blake, Ph.D.
Dean of the Graduate School

GRANT, JASON LEE (Ph.D., Industrial Engineering)
Short-Term Peak Demand Forecasting using an Artificial Neural Network with Controlled Peak Demand through Intelligent Electrical Loading (May 2014)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Shihab Asfour.
No. of pages in text. (139)

The power output capacity of a local electrical utility is dictated by its customers' cumulative peak-demand electrical consumption. Most electrical utilities in the United States maintain peak-power generation capacity by charging for end-use peak electrical demand; thirty to seventy percent of an electric utility's bill. To reduce peak demand, a real-time energy monitoring system was designed, developed, and implemented for a large government building. Data logging, combined with an application of artificial neural networks (ANNs), provides short-term electrical load forecasting data for controlled peak demand. The ANN model was tested against other forecasting methods including simple moving average (SMA), linear regression, and multivariate adaptive regression splines (MARSplines) and was effective at forecasting peak building electrical demand in a large government building sixty minutes into the future. The ANN model presented here outperformed the other forecasting methods tested with a mean absolute percentage error (MAPE) of 3.9% as compared to the SMA, linear regression, and MARSplines MAPEs of 7.7%, 17.3%, and 7.0% respectively. Additionally, the ANN model realized an absolute maximum error (AME) of 18.2% as compared to the SMA, linear regression, and MARSplines AMEs of 26.2%, 45.1%, and 22.5% respectively.

DEDICATION

I would like to dedicate this work to my grandfather Donald, grandmother Joni, father Timothy, brothers Christopher, Ryan, and Johnathan, and son Winston. I thank them kindly for their unconditional support, resolute understanding, and steadfast patience during the coursework, research, and writing of this dissertation.

ACKNOWLEDGMENTS

It is with utmost respect and admiration that I thank all of my committee members; Dr. Shihab Asfour, Dr. Khaled Yahia Zakaria Abdel-Rahman, Dr. Saman Aliari Zonouz, and Dr. Moataz Eltoukhy. Dr. Shihab Asfour, as chair, has been a pillar of strength during this enlightening scholarly process. I cannot express ample gratitude for his financial and emotional support, enduring patience, positive encouragement, academic and personal mentoring, and friendship. I thank Dr. Khaled Yahia Zakaria Abdel-Rahman for his friendship and support both academically and emotionally. Without his input, this research would not have been possible.

My appreciation also extends to the U.S. Department of Energy on behalf of its grant to the University of Miami Industrial Assessment Center. Through collaboration with many Florida & Puerto Rican manufacturing companies, the knowledge gained while working under this grant has been greatly appreciated and valued.

Gratitude is also conveyed to the Miami-Dade County General Services Administration Department on behalf of its grant to the University of Miami Department of Industrial Engineering. Without this grant, the research conducted within this dissertation would not have been possible.

I would like to thank Dr. Moataz Eltoukhy for his technical support as well as Lila Asfour for her academic support.

I kindly thank the University of Miami faculty and staff, for without them this research project would not have been possible.

Finally, my deepest gratitude goes to my entire family, immediate and extended, for their unconditional, loving support throughout this entire scholastic endeavor.

TABLE OF CONTENTS

DESCRIPTION PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AME – Absolute Maximum Error

ANN – Artificial Neural Network

ASM – Ancillary Services Market

CM – Capacity Market

CPP – Critical-Peak Pricing

cRIO – Compact Real-Time Input Output

CT – Current Transducer

DBB – Demand Bidding Buyback

DLC – Direct Load Control

DR – Demand Response

DSM – Demand Side Management

EDR – Emergency Demand Response

HP – Horsepower

HVAC – Heating Ventilation and Air-Conditioning

HZ – Hertz

ICS – Iterruptible/Curtailable Service

KW – Kilowatt

KWH – Kilowatt Hour

MAPE – Mean Absolute Percentage Error

MARSplines - Multivariate Adaptive Regression Splines

MCC1 – Motor Control Center 1

MCC2 – Motor Control Center 2

MLP – Multilayer Perceptron

MSE – Mean Square Error

MSE 3 – Main Service Entrance 3

MSE1 – Main Service Entrance 1

MSE2 – Main Service Entrance 2

NI – National Instruments

PE – Processing Element

RMS – Root Mean Square

RTP – Real-Time Pricing

SMA – Simple Moving Average

TOU – Time of Use

VI –Virtual Instrument

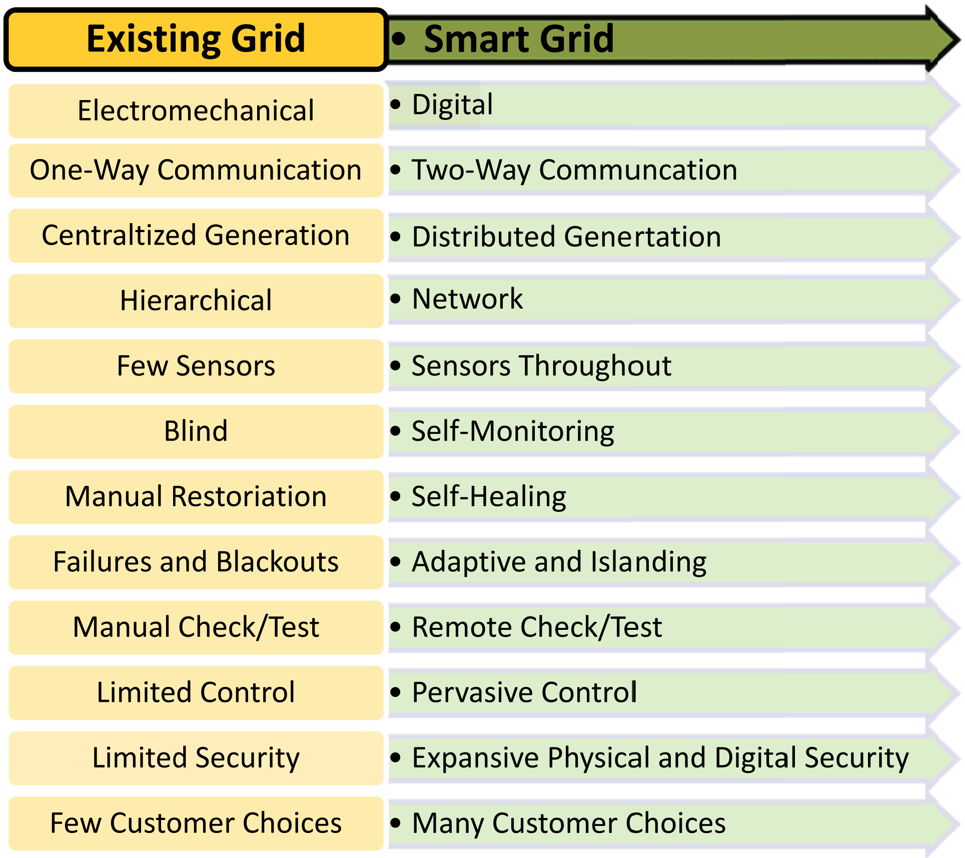
CHAPTER 1: INTRODUCTION

The Smart Grid

The term “Smart Grid” has recently entered the vernacular of most economies around the world. Whether developed or developing, nations are currently focusing on long-term and short-term energy policy including energy sources, power conversion/ production, methods of transmission and distribution, generation diversification, demand response, and adoption of energy conservation and efficiency initiatives in an effort to reduce total carbon footprint. The electricity grid, as it currently exists, cannot effectively address all of these issues. The U.S. Department of Energy’s modern grid initiative states that a “smart grid” integrates advanced sensing technologies, control methods, and integrated communications into the existing electricity grid (Rahman, 2009). It is important to distinguish today’s transmission grid from tomorrow’s comprehensive energy smart grid. Contrary to popular belief, today’s electrical transmission grid in developed countries is actually efficient, smart, and intelligent. Distribution of electricity to customer end-use, however, is inefficient and unintelligent. In order to better integrate distributed power generation, more data, automation, and intelligent control are required.

The current electrical grid was designed and developed with electrical power flowing in only one direction. Of the total fuel energy used for electrical power, approximately two-thirds are lost as waste heat and other conversion inefficiencies (Farhangi, 2010). For example, transmission lines account for approximately 8% utility output loss while 20% of electrical generation capacity exists solely to meet peak demand when on average only 5% is required (Farhangi, 2010). Additionally, since utility assets are tightly linked, any failure in the grid chain results in systematic failure from the point of failure all the way to end-use. An effective smart grid will provide utility companies with extensive visibility and widespread control over assets and services. It will be self-healing and will effectively adapt to system irregularities.

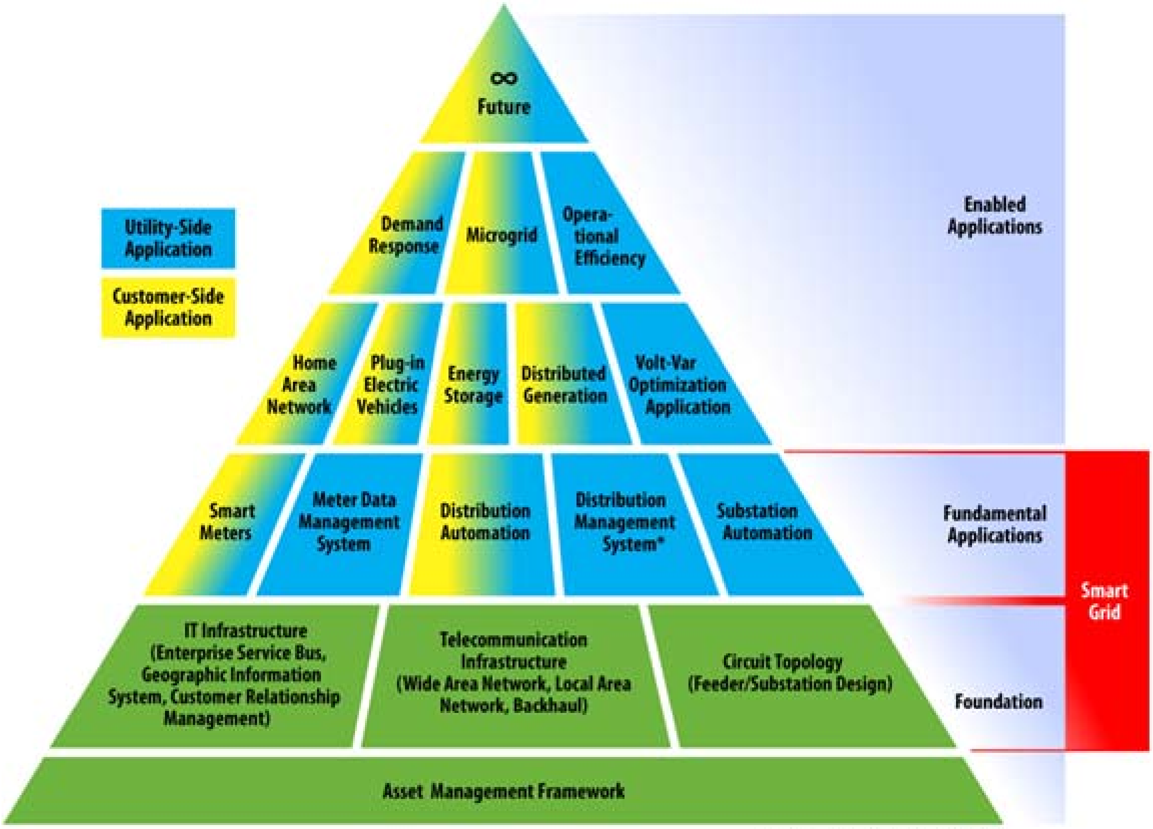
The advancement of the smart grid has recently been accelerated due to the emergence of advanced meter infrastructure and internet usage (Hart, 2008). In order to build a state-of-the-art smart grid, however, several technologies must coalesce: information technology, communication technology, and power system engineering. Figure 1-1 depicts the complete transition between the existing grid and a smart grid (Farhangi, 2010).



Existing Grid	• Smart Grid
Electromechanical	• Digital
One-Way Communication	• Two-Way Communcation
Centraltized Generation	• Distributed Genertation
Hierarchical	• Network
Few Sensors	• Sensors Throughout
Blind	• Self-Monitoring
Manual Restoriation	• Self-Healing
Failures and Blackouts	• Adaptive and Islanding
Manual Check/Test	• Remote Check/Test
Limited Control	• Pervasive Control
Limited Security	• Expansive Physical and Digital Security
Few Customer Choices	• Many Customer Choices

Figure 1-1 Existing grid to smart grid evolutions (Farhangi, 2010).

Fundamentally, a well designed smart grid layers intelligence over utility assets, thereby utilizing current assets more effectively and efficiently.   Figure 1-2 illustrates the fundamental building blocks of a smart grid (BChydro, 2009).   While basic lateral integration provides for a firm smart grid foundation, vertical integration is necessary for advanced smart grid function. For example, effective demand response requires universal assimilation of smart meters and end-use area networks. Given current infrasture design and political sentiment, smart grid evolution will undoubtedbly be evolutionary as opposed to revolutionary. The current grid will be modified in steps. Distributed control and monitoring systems will form the basis for total system integration and strategic decision and policy formulation.


∞
Future
Utility-Side Application
Customer-Side Application
Demand Response
Microgrid
Opera-tional Efficiency
Enabled Applications
Home Area Network
Plug-in Electric Vehicles
Energy Storage
Distributed Generation
Volt-Var Optimization Application
Smart Meters
Meter Data Management System
Distribution Automation
Distribution Management System*
Substation Automation
Fundamental Applications
Smart Grid
IT Infrastructure (Enterprise Service Bus, Geographic Information System, Customer Relationship Management)
Telecommunication Infrastructure (Wide Area Network, Local Area Network, Backhaul)
Circuit Topology (Feeder/Substation Design)
Foundation
Asset Management Framework

*includes Energy Management System

Figure 1-2 Smart grid pyramid (BChydro, 2009).

The closest model resembling a smart grid is the Circuit of the Future at Southern California Edison (Yinger, 2006).

The present electrical grid, for the most part, is blind, unidirectional, and passive. The source of generation currently receives no real-time information regarding end-usage. In order to maintain function during periods of peak demand, the grid has been engineered beyond what is truly required in order to function during periods of maximum anticipated peak demand. The peak demand capacity is thus rarely utilized making the system intrinsically inefficient. Additionally, expedited global capacity expansion as the result of soaring demand for electricity has typically neglected intelligent investment in utility infrastructure. The lack of sensible planning during recent utility growth has undoubtedly contributed to the overall instability of the grid system.

In North America, information and communication systems are limited to less than 25% of the distribution network. Approximately 90% of all blackouts/brownouts occur within the distribution network realm. By addressing demand-side management with advanced technologies, controlled growth and development of a smart utility grid is possible.

The International community is beginning to establish standards with regard to smart grid design and implementation. In 2001, the Electric Power Research Institute (EPRI) established IntelliGrid as a technical foundation for a smart power grid that links electricity with communications and computer control to achieve tremendous gains in reliability, capacity, and customer services. IntelliGrid's five main components include: IntelliGrid architecture, fast simulation and modeling (FSM); communications for distributed energy resources (DER); consumer portal; and advanced monitoring systems.

EPRI Advanced Distribution Automation (ADA) focuses on the distribution system of the future. The U.S. Department of Energy's (DOE) Modern Grid Initiative attempts to standardize the modern grid as an innovatively comprehensive model of electricity delivery. Also, the U.S. DOE created the GridWise program to modernize distribution grid infrastructure and operations by incorporating bidirectional flow of electricity and information. The Advanced Grid Application Consortium (GridApps), formed by Concurrent Technologies Corporation and paid for by the DOE in 2005, applies utility technologies and practices to modernize the grid. GridWorks, also within the DOE, focuses on improving the reliability of the grid by upgrading critical grid infrastructure. Below, Table 1-1 lists details of Title 13 (Smart Grid Section) of the U.S. Energy Policy and Security Act of 2007 (Saint, 2009).

Table 1-1 Title 13 U.S. Energy Policy and Security Act of 2007 (Saint, 2009).

Section	Mandate
1301	Establishes a federal policy to modernize the electric utility transmission and distribution system to maintain reliability.
1302	Calls for DOE to report to Congress on the deployment of Smart Grid technologies and any barriers to deployment.
1303	Directs DOE to establish a Smart Grid Advisory Committee and a Smart Grid Task Force to assist with implementation.
1304	Directs DOE to conduct Smart Grid RD&D and to develop measurement strategies to assess energy savings and other aspects of implementation.
1305	Directs the National Institute of Standards and Technology to establish protocols and standards to increase the flexibility of use for Smart Grid equipment and systems.
1306	Directs DOE to create a program that reimburses 20% of qualifying Smart Grid investments.
1307	Directs states to encourage utilities to employ Smart Grid technology and allows utilities to recover Smart Grid investments through rates.
1308	Requires DOE to prepare a report to Congress on the effect of private wire laws on the development of combined heat and power facilities.
1309	Directs DOE to report to Congress on the potential impacts of Smart Grid deployment on the security of electricity infrastructure and operating capability.

Distribution Vision 2010 (DV2010) addresses interruption of service as a result of malfunctioning and/or poorly designed feeders. Most recently, the ANSI C12.22 standard addresses smart meters while the IEC 61850 standard deals with substation automation. The United States National Institute of Standards and Technology (NIST) Framework and Roadmap for Smart Grid Interoperability Standards (Framework, N. I. S. T., 2010) is currently identifying priority areas for standardization. Finally, the DOE's Visualizing Energy Resources Dynamically on Earth (VERDE) will provide wide-area grid awareness, integrating real-time sensor data, weather information and grid modeling with geographical information. Both micro and macro activity can be monitored and analyzed simultaneously with detailed reporting on blackouts and power quality.

So far, the prevailing technology of the Smart Grid is the Smart Meter or Automated Metering Infrastructure (AMI). AMI serves as a bidirectional information conduit to the Home Area Network (HAN) or to Commercial/Industrial energy automation systems. With AMI, end-users can select rate options based on time-of usage as well as benefit from rate discounts by relinquishing control of high-energy consuming appliances to the utility in order to reduce utility demand. Eventually instantaneous pricing information will allow end-users to curtail loads either manually or through automation and policy set points. For the utility, AMI provides real-time demand/consumption information, grid performance, remote disconnect/ reconnect control, and outage, blink and voltage monitoring across the grid (Saint, 2009). The large amount of data from smart meters will be analyzed by operators and planners which will be used to achieve better reliability and asset management (Backer, 2007).

The smart grid of the future will become decentralized where distributed generation (DG) plays a critical role (Divan & Johal, 2006). DG relies on local power generation where proximity to the end-user is much less compared with the current average distance between the end-user and the utility's source. Such power can be derived from any decentralized energy source including renewables. Generation technology in a DG model is small-scale, modular, distributed, efficient, and cost effective. Each DG system varies in size from a few kilowatts to hundreds of megawatts. Their design focuses on efficiency, reliability, safety, security, low cost, and environmental impact. Inherent in the design of a DG system is the increased utilization efficiency of all available energy sources. Figure 1-3 compares the traditional power supply system with a distributed generation system (Wei, Yu-hui, & Jie_lin, 2009). Also, when actual load differs from the computed forecasted demand, a DG system is more adaptable to real-time load requirements, utilizes energy sources more efficiently, and reduces overall negative environmental impact. By integrating advanced metering, robust communications capability, extensive automation, distributed generation, and distributed storage, smart grids of the future will self-heal, provide high reliability and power quality, be resistant to cyber-attacks, operate with multi-directional power flow, increase equipment utilization, operate with lower cost, and offer customers a variety of service choices (Brown, 2008).

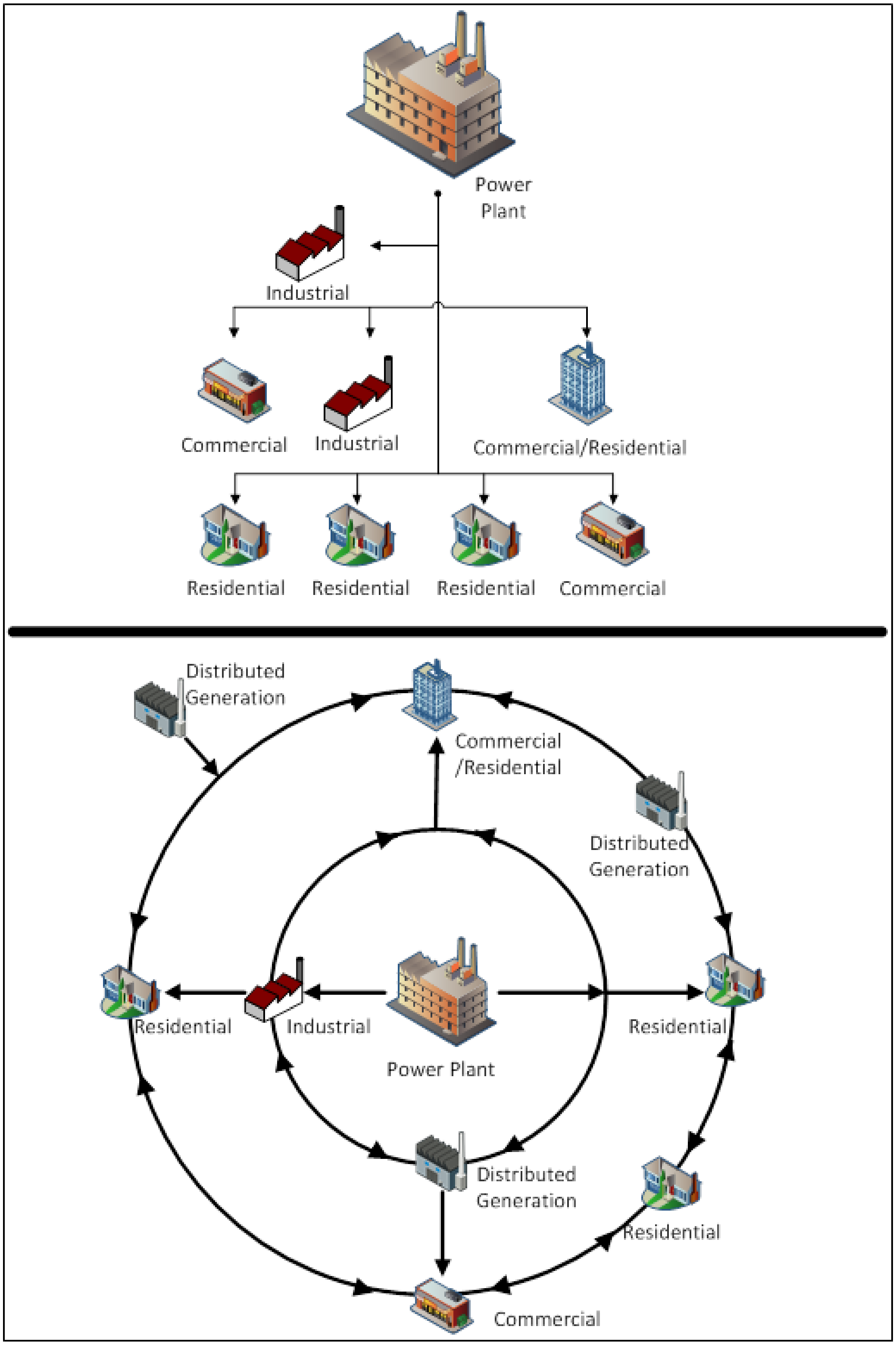


Figure 1-3 Current Vertical Power Generation vs. Distributed Generation (Wei et al., 2009).

A critical component of the smart grid with distributed generation is peak demand forecasting. Given accurate, real-time demand information, utilities are able to meet demand more efficiently by building appropriately sized power plants and support infrastructure. Real-time demand information reduces energy wastage, thereby lessening the overall environmental impact of energy conversion. The research conducted as a result of this dissertation aims to further develop as well as complement existing peak demand forecasting methodologies in an effort to better understand and control peak-demand occurrences experienced by the end-user.

An electrical utility's demand charge, measured in kilowatts (KW), is the price charged for the peak amount of power demanded/consumed at a particular instant by an end-user during one billing cycle. The KW demand charge is not to be confused with the more commonly known utility's power consumption charge, which is the amount of power consumed over a period of time; otherwise referred to as kilowatt hours (KWH). The KW demand charge, commonly incurred by large buildings, industrial and commercial complexes, and large manufacturers but more recently also being incorporated into modern residential pricing structures, is the measurement of peak power demanded/consumed at a particular moment during one billing cycle. For most utilities, KW demand is metered throughout the billing cycle in 15 or 30 minute intervals. Demand–related charges represent anywhere from 30 to 70 percent of most commercial and industrial customers' electric bills.

Existing peak-demand forecasting research literature focuses primarily on the utility conversion level for aggregate demand. This research, however, aims to further develop as well as complement existing peak demand forecasting methodologies in an effort to better understand and control peak-demand occurrences experienced not only by the power utilities but more specifically by the end-user.

Demand Response

In developed economies, mostly in the western hemisphere, Electricity Markets were spawned as a result of deregulation with respect to power system generation coupled with restructuring of the power sector in general (Kirschen, 2003). As a result of monopoly collapse within the energy sector, electricity markets were supposed to dramatically improve collective power system efficiency while reducing the cost of electricity to the end user (Kirschen, 2003). Since electrical energy is a unique commodity which functions differently from other commodities, however, electricity markets do not operate similarly to others (Torriti, Hassan, & Leach, 2010). More specifically, electrical energy supply and demand must always be in equilibrium (Vale, Pinto, Praca, & Morais, 2011). Complicating the matter, due to technical and economic factors, electrical energy innately has a limited storage capacity.

Demand Response (DR) is one aspect of Electricity Markets that has much growth potential in the near future (Walawalkar, Fernands, Thakur, & Chevva, 2010). While delivering consumer benefits, DR encompasses many advantages for system-wide electric energy system management. Due to the capital-intensive nature of power system’s infrastructure, DR offers a relatively inexpensive approach to improving the efficiency of system-wide electrical energy systems (Albadi & El-Saadany, 2008). In order to deal with unexpected variances in supply and demand levels, DR technology is capable of providing the system operator with fixed load curtailment capacities. Global adaptation and implementation has been documented (Woo & Greening, 2010). Wholesale market DR is occurring in the United States (Cappers, Goldman, & Kathan, 2010), Europe (Torriti et al., 2010), China (Wang, Bloyd, Hu, & Tan, 2010), as well as other places (River, 2005). The lack of DR had contributed to difficulties with transitioning from a traditional regulated industry to a competitive one. Time dependent pricing or real-time pricing, however, has had a positive impact on overall operation efficiency and industry investment (Woo et al., 2010).

Demand Side Management (DSM), DR, and/or load management all pertain to the management and actions of end-use behavior (Faria & Vale, 2011). When a utility experiences high demand, these load management programs facilitate system load balancing by avoiding peak occurrences (Electricity Advisory Committee, 2009). When wholesale electricity market prices are high or when overall grid system reliability is compromised, DR programs offer incentives to end-users in order to affect time of use, instantaneous demand level, and/or aggregate electricity consumption (International Energy Agency, 2003).

Demand response are categorized into two groups: price-based and incentive-based DR (QDR, Q. ,2006; Avci, Erkoc, Rahmani, & Asfour, 2013). With time-of-use (TOU), real-time pricing (RTP), and critical-peak pricing (CPP) rate structures, price based DR motivates customers to alter their consumption in repsonse to fluctuations in their purchase prices.

By shifting consumption from periods of higher energy prices to periods of lower energy prices, end users can reduce their energy cost. It is noted that price based DR is entirely voluntary.

Time-of-use rates typically vary over a twenty four hour cycle and are determined by the average cost of producing and supplying electricity during the cycle. Rate information is provided one cycle, or twenty four hours, ahead. Real-time pricing follows wholesale prices on a much shorter cycle: typically one hour (Sioshansi & Short, 2009). Rate information is provided one cycle, or one hour, ahead. Critical-peak-pricing combines TOU and RTP. For the most part, it is a TOU model, but with significantly higher peak period pricing when system integrity is jeopardized or when supply costs escalate dramatically.

Incentive-based demand response involves fixed or fluctuating time incentives coupled with defined electricity rates. All players, including the district grid operators, load-serving entities, and/or utilities, dictate such incentives. End-users that fail to respond during a peak event in a manner previously agreed upon are penalized financially.

There are six typical incentive-based DR programs: Direct Load Control (DLC), Interruptible/Curtailable Service (ICS), Demand Bidding/Buyback (DBB), Emergency Demand Response (EDR), Capacity Market (CM), and Ancillary Services Market (ASM) (Faria et al., 2011). Primarily targeted to residential and small commercial accounts, DLC gives a program operator the ability to remotely power down end-users' electrical equipment. Targeted towards large-scale industrial account, ICS requires an end-user to curtail their power demand by shutting off equipment during periods of extreme total system demand. Rate discounts and bill credits serve as the incentive for compliance while penalties discourage failure to respond according to pre-determined contractual obligations. Targeted towards large accounts in general, DBB has end-users offering curtailment capacity bids. EDR is just a combination of DLC and ICS and is employed whenever supply reserves are inadequate. With CM, end-users curtail their loads to add to system capacity in an effort to compensate for otherwise conventional power generation. And ASM is like DBB, but with direct offers made in the ancillary services markets. Figure 1-4 demonstrates DR incorporation in the electric system based on time.

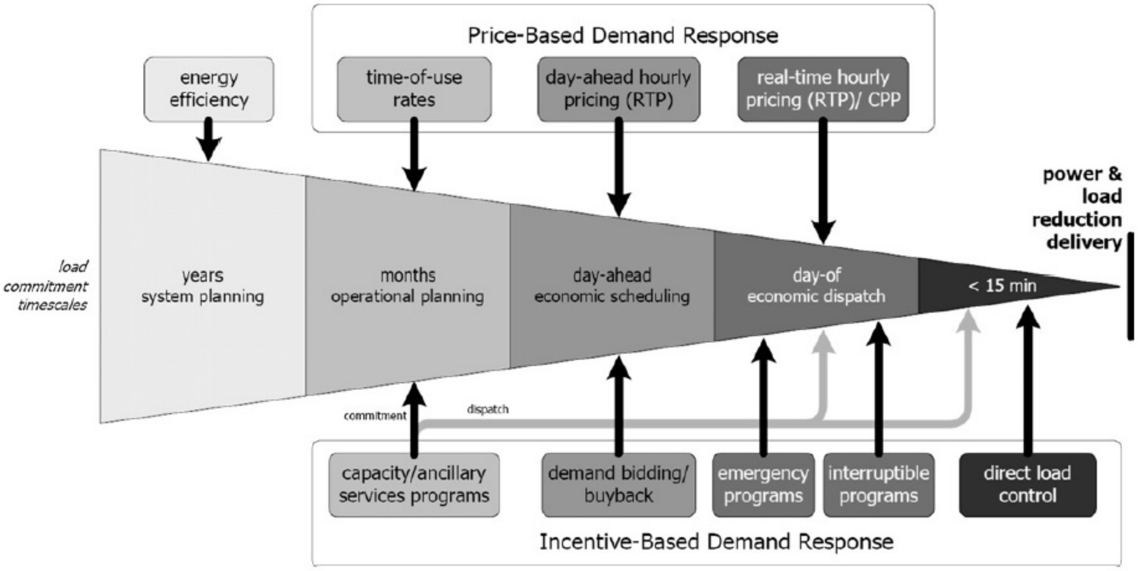

Price-Based Demand Response
energy efficiency
time-of-use rates
day-ahead hourly pricing (RTP)
real-time hourly pricing (RTP)/ CPP
power & load reduction delivery
load commitment timescales
years system planning
months operational planning
day-ahead economic scheduling
day-of economic dispatch
< 15 min
commitment
dispatch
capacity/ancillary services programs
demand bidding/ buyback
emergency programs
interruptible programs
direct load control
Incentive-Based Demand Response

Figure 1-4 DR incorporation in the electric system (U.S. Department of Energy, 2006).

This research focuses primarily to augment current ICS DR programs by giving localized peak-demand forecasting of the end-user to the end-user. Currently ICS DR is initiated based on utility side supply constraints in an effort to better manage total system-wide peak power production capability. This research, however, fosters a real-time approach to curtailment whereby action is taken by the end-user when forward predicted demand by the end-user approaches some predetermined peak. Such action would thereby empower the end-user to lessen their overall peak-demand and its corresponding cost during each billing cycle. The goal of this research is to develop a new method for forecasting peak demand in large buildings using ANNs and a specified period of training days. The approach is intended for medium to large-scale building systems, including government and corporate building campuses, where overall forward demand knowledge is of interest in order to facilitate effective and efficient building electrical utilization and loading. A developed real-time electrical monitoring system prototype capable of forecasting using ANNs is applied to a large government building.

CHAPTER 2: LITERATURE REVIEW

Neural Networks – History

Within his metaphysical writings, Aristotle (384-322 BC) embraced two principles of propositional logic: the *Law of Excluded Middle* wherein every statement is either true or false, and the *Law of Contradiction* which states that no statement is both true and false. Chrysippus (280-205 BC est.) then advanced propositional logic by listing valid inference schemata for complex argument structure (Klement, 2005):

1. If the first, then the second; but the first, therefore the second.
2. If the first, then the second; but not the second; therefore, not the first.
3. Not both the first and the second; but the first; therefore, not the second.
4. Either the first or the second [and not both]; but the first, therefore, not the second.
5. Either the first or the second; but not the second; therefore the first.

Surprisingly, it then took almost another two millennia before the next major advancement in propositional logic. While Augustus DeMorgan (1806-1871) worked to create symbolic logic, it was George Boole (1815-1864) who developed a mathematical algebra in an attempt to update the syllogistic logic first established by Aristotle and Chrysippus. Boolean logic, named after Boole, was based on binary as opposed to decimal arithmetic. Such logic was before its time, however, since it was not really utilized until the birth of integrated circuitry which ultimately led to the microprocessor and the modern computer.

Neural Networks – Biological History

A neuron, translated in Greek as nerve cell, is the most basic component of the brain and nervous system. As a basic processing system, a neuron communicates with other neurons via filamentary input paths called dendrites. These signals are transmitted as electric impulses across a synaptic gap by means of a chemical process (Fausett, 1994). Dendrites are tree-like in structure and combine to form dendritic trees which are attached to the body of the nerve cell called the soma. The soma, in turn, combines all signals being received by the dendrites. Once an acceptable amount of information is receive and its threshold level has been reached, the cell then transmits a new signal, or action potential, via its axon to additional cells. This is performed by neurotransmitters or chemical messengers which are stored in vesicles. The union between a neuron's axon and another neuron's dendrite is called the contact, often referred to as the synapse which is comprised of the presynaptic terminal, the cleft or the synaptic junction, and the postsynaptic terminal (Kartalopoulos & Kartakapoulos, 1997). Figure 2-1 demonstrates a generic biological neuron (Fausett, 1994) and Figure 2-2 shows synapse detail (Kartalopoulos et al., 1997). Signal transmission between neurons occurs as a result of dynamic potassium, sodium, and chloride ion levels on opposite sides of the axon sheath also known as white brain matter. This depolarization ensues once a neurotransmitter has crossed the synaptic junction hence changing postsynaptic potential. Two postsynaptic potential states exist depending on the neurotransmitter type: excitatory (+) or inhibitory (-) (Kartalopoulous et al., 1996).

Since the signal is either sent or not sent, neuron signaling is considered binary, although frequency timing of signal transmission can be interpreted as additional information.

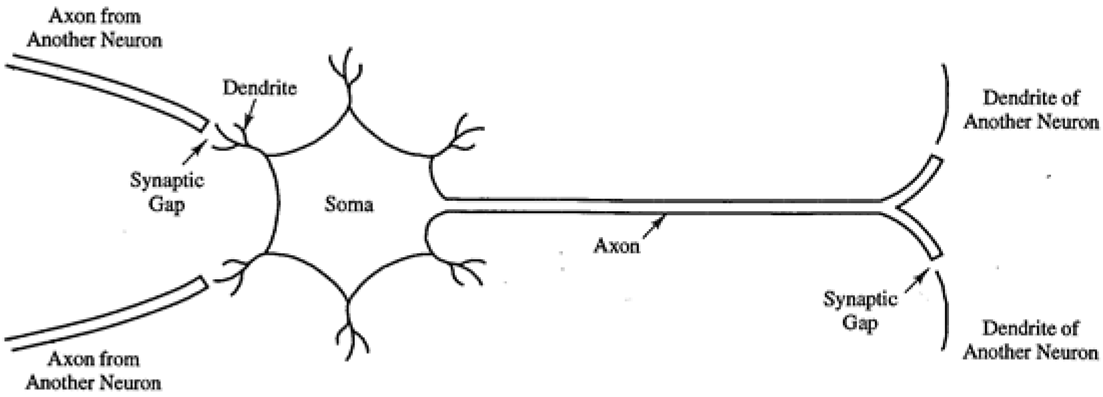


Figure 2-1 Generic biological neuron.

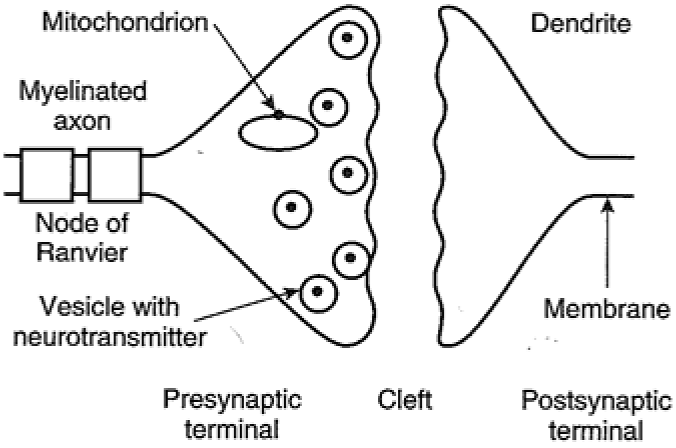


Figure 2-2 Generic biological synapse.

According to Fausett, the following properties of biological neurons are incorporated into artificial neural networks:

1. The processing element receives many signals.
2. Signals may be modified by a weight at the receiving synapse.
3. The processing element sums the weighted inputs.
4. Under appropriate circumstances (sufficient input), the neuron transmits a single output.

5. The output from a particular neuron may go to many other neurons (the axon branches.
6. Information processing is local (although other means of transmission, such as the action of hormones, may suggest means of overall process control).
7. Memory is distributed:
    a. Long-term memory resides in the neurons’ synapses or weights.
    b. Short-term memory corresponds to the signals sent by the neurons.
8. A synapse’s strength may be modified by experience.
9. Neurotransmitters for synapses may be excitatory or inhibitory.
10. Fault tolerance

Neural Net Applications

Neural nets are currently being adapted to address a plethora of problems across many disciplines including but not limited to: signal processing, control, optimization, pattern recognition, medicine, speech production, speech recognition, business processes and analysis, forecasting, data mining, music, and astrology.

Artificial Neural Networks

An artificial neural network (ANN) attempts to simulate biological neural networks via advanced data manipulation ANNs were initially developed as standard mathematical models of human cognition or neural biology, based on the assumptions that (Fausett,1994):

1. Information processing occurs at many simple elements called neurons
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal

According to Fausett, a neural network is characterized by:

1. Its pattern of connection between the neurons (called its architecture)
2. Its method of determining the weights on the connections (called its training, or learning algorithm
3. Its activation function

Hassoun describes neural networks as parallel computational models comprised of densely interconnected adaptive processing units (Hassoun, 1995). In the past, computational programming, which follows a pre-specified set of instructions, was used to solve various mathematical problems. Neural networks, unlike programming techniques, are adaptive in nature and learn by analyzing historical system data. They can be used for pattern classification, speech synthesis and recognition, adaptive interfaces between humans and complex physical systems, function approximation, image compression, associate memory, clustering, forecasting and prediction, combinatorial optimization, nonlinear system modeling, and control (Hassoun, 1995). In reference to pattern computation, neural network applications are capable of storing and recalling data or patterns, classifying patterns, performing general mappings from input patterns, classifying patterns, performing general mappings from input patterns to output patterns, grouping similar patterns, or finding solutions to constrained optimization. Since neural networks can perform parallel computation, they are able to produce results much more quickly than comparable computational models. According to Hassoun, ANNs:

> Are "neural" in the sense that they may have been inspired by neuroscience, but not because they are faithful models of biologic neural or cognitive phenomena. In fact, the majority of the network models…are more closely related to traditional mathematical and/or statistical models such as optimization algorithms, nonparametric pattern classifiers, clustering algorithms, linear and nonlinear filters, and statistical regression models than they are to neurobiologic models.

In a neural network, there are a significant number of basic processing elements called neurons, units, cells, or nodes. These nodes are interconnected, each with an associated weight (bias) or level of importance. Each node can be described by its current state which is represented by its degree or magnitude of action. This state status is continuously broadcasted from one to many other nodes. While a node can only send one state signal at a time, it often sends its state status to many other nodes. Figure 2-1 depicts a simple neuron Y receiving input signals x_1-x_n from nodes X_1-X_n. The weights for each node's signal are represented by w_1-w_n.

The cumulative input for node Y (*Ycumulative*) is simply the sum of the weighted signals from all nodes,

$$\mathit{Ycumulative} = \sum_{j=1}^{N} x_j w_j$$

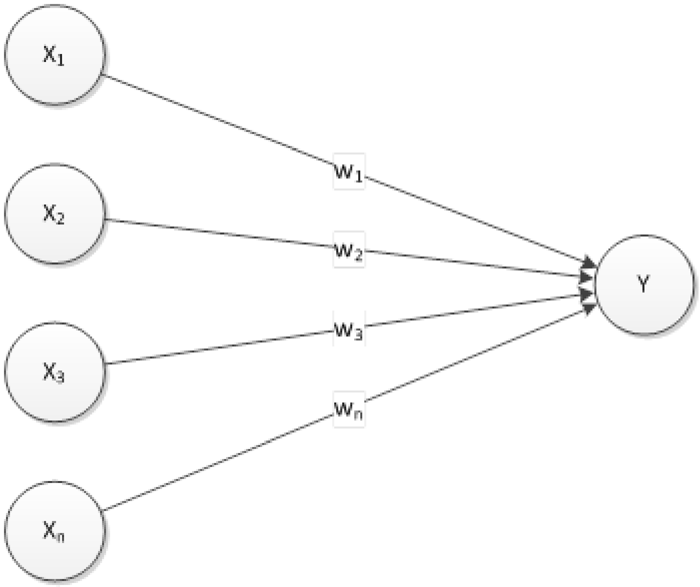


Figure 2-3 Basic node/neuron (artificial).

Neural Network Functions

The basic neuron model owes its widely accepted derivation from research conducted in 1943 by McCulloch, a neurobiologist, and Pitts, a statistician (McCulloch & Pitts, 1943). Their neuron model is shown in Figure 2-4.

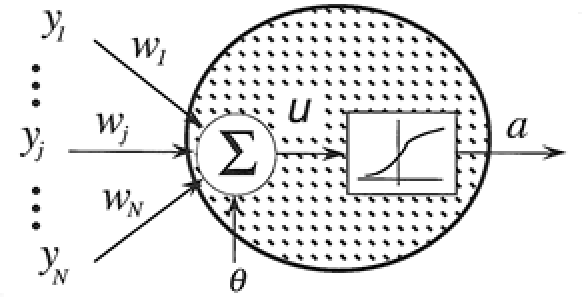


Figure 2-4 Neuron Model (McCulloch et al., 1943).

The nonlinearity function, represented as the function curve in the neuron model in Figure 2-4, ensures a bounded controlled output of the neuron's actual response.

Depending on the algorithm employed, different nonlinearity functions are applied. Figure 2-5 represents typical bounded nonlinearity functions (Kartalopoulos et al., 1997).

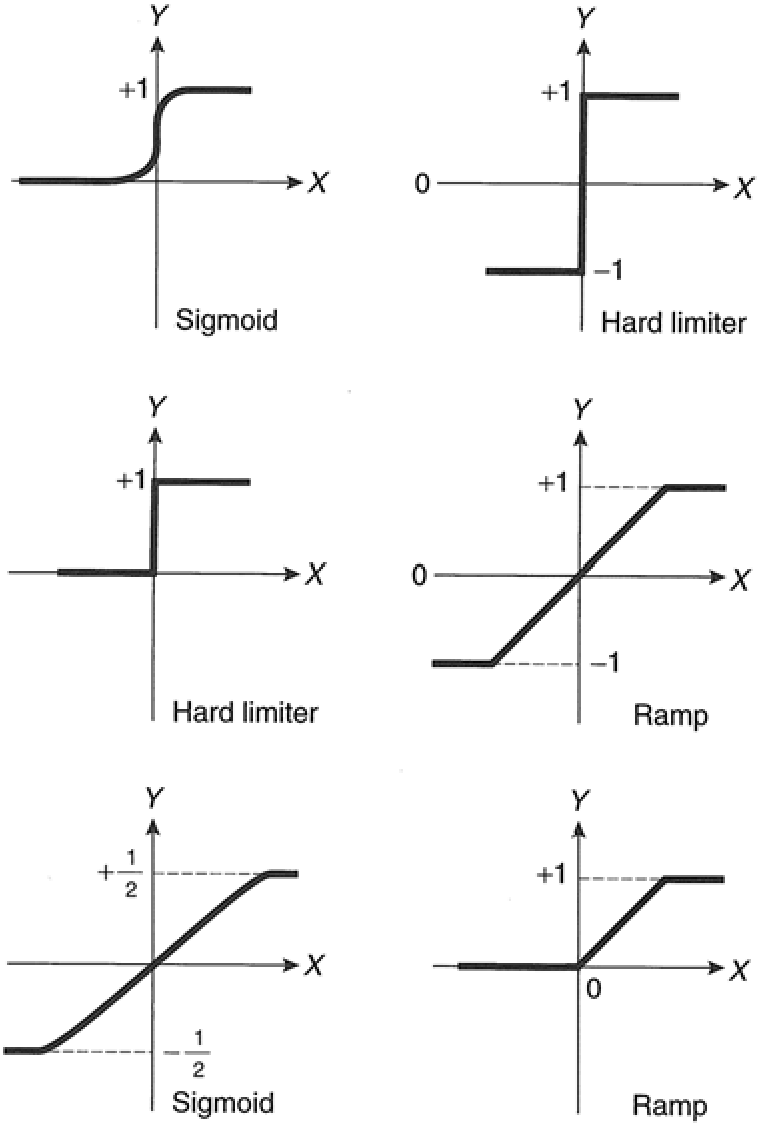


Figure 2-5 Nonlinearity functions (Kartalopoulos et al., 1997).

Similarly, for McCulloch's model:

$$u = \sum_{j=1}^{N} y_j w_j + \theta$$

Here, θ represents the bias or threshold.

Other net functions and neuron activation functions employed are shown in Table 2-1 and Table 2-2 (Hu & Hwang, 2010).

Table 2-1 Net Functions (Hu et al., 2010).

Net Functions	Formula	Notes
Linear	$u = \sum_{j=1}^{N} w_j y_j$	Most popular
Higher order (2^{nd} order shown)	$u = \sum_{j=1}^{N} \sum_{k=1}^{N} w_{jk} y_j y_k + \theta$	u_i is a weighted linear combination of higher order polynomial terms of input variable. The number of input terms equals N^d, where d is the order of the polynomial
Delta ($\sum$ - $\prod$)	$u = \prod_{j=1}^{N} w_j y_j$	Rarely used

Table 2-2 Neuron Activation Functions (Hu et al., 2010).

Activation Function	Activation Formula $a = f(u)$	Derivatives $\frac{df(u)}{du}$	Notes
Sigmoid	$f(u) = \frac{1}{1+e^{{-u}/_{T}}}$	$f(u)\,[1 - f(u)]/T$	Commonly used; derivative can be computed from *f(u)* directly.
Hyperbolic tangent	$f(u)\ \tanh\left(\frac{u}{T}\right)$	$(1 - [f(u)]^2)/T$	*T* = temperature parameter.
Inverse tangent	$f(u) = \frac{2}{\pi} tan^{-1}\left(\frac{u}{T}\right)$	$\frac{2}{\pi T} \cdot \frac{1}{1 + (u/T)^2}$	Less frequently used.
Threshold	$f(u) = \begin{cases} 1, & u > 0; \\ -1, & u < 0. \end{cases}$	No derivative at u=0	
Gaussian radial basis	$f(u) = exp[-\|u - m\|^2/\sigma^2]$	$-2(u - m) \cdot f(u)/\sigma^2$	Used for radial basis neural network; *m* and σ^2 are parameters to be specified
Linear	$f(u) = au + b$	a	

Neural Network Architecture

Further expansion of the basic node by combining several unique nodes creates a simple neural network. Simplistically, inputs and their respective weights yield Y's state. Y then sends signals, with respective weights (u_1.,u_2,…u_n), to the next set of nodes. Figure 2-5 shows a basic neural network. Despite its simplicity, the middle (hidden node 'Y') coupled with a nonlinear action function results in a more dynamic system capable of solving more problems than a traditional network with only inputs and outputs.

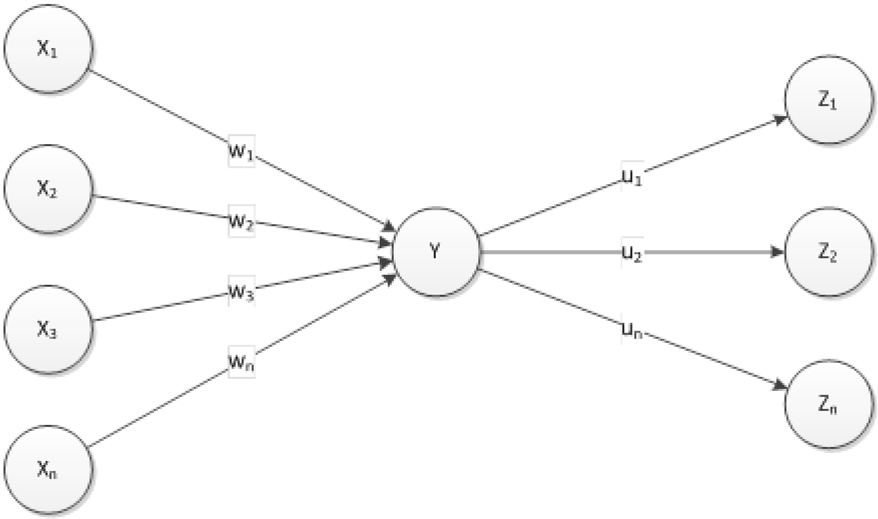


Figure 2-6 Basic neural network.

Generally, input nodes may connect output nodes, nodes may mutually connect with each other and with reciprocal inhibition it is even possible for a node to connect to itself (Croall & Mason, 1992).

Figure 2-7 depicts a single layer neural network with one layer of weights. Here, the input units (X_1…X_i…X_n) capture external signals. The single layer neural network, upon applying weights to the input signals, outputs its response (Y_1…Y_j…Y_m).

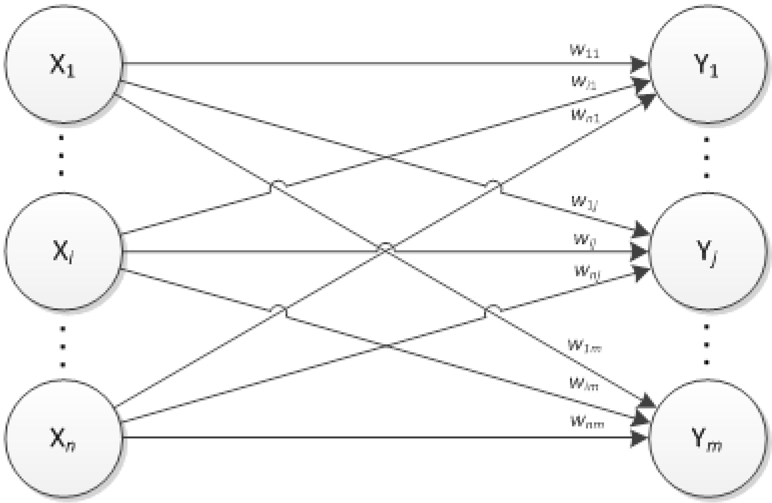


Figure 2-7 Single layer neural network.

Figure 2-8 expands the single layer neural network model to a multiple layer neural network wherein the ‘Z’ nodes are hidden. As with the single layer network, input signals come into the network via input units ($X_1 \ldots X_i \ldots X_n$). These signals are weighted and enter the hidden layer ($Z_1 \ldots Z_j \ldots Z_q$) for another weighted manipulation before exiting as output signals ($Y_1 \ldots Y_k \ldots Y_m$). Multiple layer networks can be applied to more complex problems in comparison to a single layer approach. However, with multiple layer networks, training is more challenging. With effective training, multiple layer networks often solve problems that single layer networks cannot.

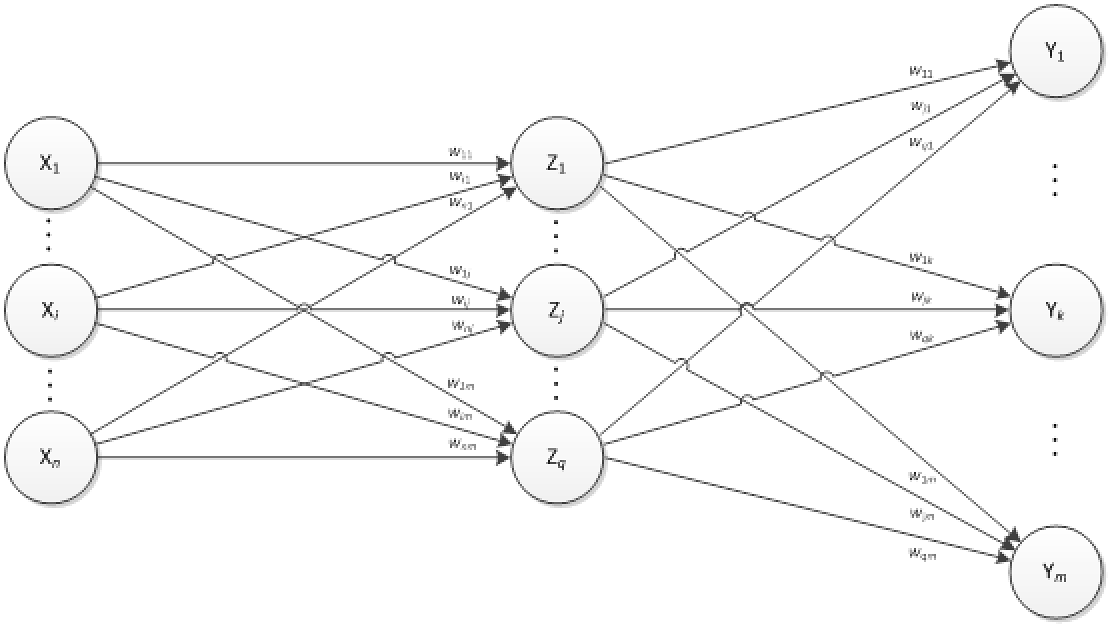


Figure 2-8 Multiple layer neural network.

Artificial Neural Network Learning

In order for an ANN to learn, it is important to first understand how humans learn, the processes involved, the speed at which humans learn, and any impediments that might interfere with human learning. Given unique learning methods which are not universal, behaviorists study different species, including humans, to better understand how learning differs. In an attempt to mimic biological learning processes, a neural network engineer applies and integrates the most efficient learning process observed by behaviorists to the digital circuitry and logic design of the ANN.

In a neural network, learning occurs when the network adjusts itself in response to a stimulus in an effort to produce a valued response. Being a continuous classification process, a stimulus is both recognized and matched to a current classification in the network or if it is not recognized, a new classification set is created. An ANN learns dynamically and responds to a stimulus by adjusting its synaptic weights in an effort to make the output response converge towards the anticipated response. In an ANN, learning has thus occurred and knowledge gained when actual response equals the anticipated response.

Unsupervised ANN Learning

An ANN that learns without instruction from a teacher has no objective output. The training method entails interpreting all input signals as patterns which are then categorized. When new signals arrive, their pattern is compared with ones already classified. If there is no match, a new category is created. It is important to note that the term "unsupervised" is a bit misleading. Despite the lack of instruction, the system does require a pre-defined policy for how categories are created. Additionally, this policy may need to be updated even after the system is implemented. Commonly, network programmers will tailor a network so that the categorization process is actually built into the overall network design.

Supervised ANN Learning

An ANN that learns from a teacher has an objective output. When output differs from the objective output, the ANN adjusts the influence of the bias weights in order to make the output converge toward the objective output. The goal is to ultimately eliminate any deviation from the objective output. When humans learn to speak, we first hear pronunciation of a word from an instructor. The observation of the teacher's example is then stored in memory. When the student attempts to repeat the pronunciation, if the deviation is too significant from the one stored in memory, another attempt is made. This process continues until the deviation becomes insignificant. The overall processing power required to minimize the deviation depends on the algorithm used. Many mathematical optimization techniques assist with this computation.

Reinforced Learning

Reinforced learning does implement a teacher but does not consider convergence; rather, reinforced learning requires a binary (true or false) indication of an input signal pattern matching a particular desired objective output (Kartalopoulos et al., 1997). The teacher withholds the target output and simply administers a true or false response. If the teacher's response is incorrect, the network adapts by modifying its parameters until it responds with the desired output. This method differs from slightly from the supervised method since there is no indication whether the output response is converging towards the desired output. Creating boundaries prevents infinite attempts at finding the solution.

Competitive Learning

Another derivation of supervised learning is competitive learning. Here, there are multiple output neurons which all compete to produce an output nearest the desired objective output. The winning output neuron is then established as the dominant neuron for the associated stimuli and all other output neurons discontinue their respective output. For each input stimulus, there is a unique dominant output neuron.

Hebbian Learning

Hebbian learning was adopted when (Kartalopoulos et al., 1997):

> In 1949, Donald Hebb stated that when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes place in firing it, some growth process or

metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased. Thus, the synaptic strength (known as weight *w*) between cell A and cell B is modified according to the degree of correlated activity between input and output. This type of learning is called Hebbian learning, a term encountered frequently in ANNs. Anti-Hebbian learning refers to artificial neural networks where the synaptic contacts are inhibitory only.

ANN Characteristics

A neural network is essentially a dynamic mathematical system that can be represented as a set of combined differential equations. Similar to feedback control theory, modest changes to the values of the model's parameters will result in stability, oscillating instability or chaotic instability. Kartalopoulos et al. characterizes neuronal networks by:

1. Collective and synergistic computation (or neurocomputing).
    a. Program is executed collectively and synergistically.
    b. Operations are decentralized.
2. Robustness.
    a. Operation is insensitive to scattered failures.
    b. Operation is insensitive to partial inputs or inputs with inaccuracies
3. Learning.
    a. Network makes associations automatically.
    b. Program is created by the network during learning/
    c. Network adapts with or without a teacher; no programmer intervention.
4. Asynchronous operation; biological neural nets have no explicit clock to synchronize their operation. A number of ANNs require a clock.

When the network continuously produces the objective output for every trained input signal, its performance is considered 100% effective. Kartalopoulos et al., 1997, states when designing an ANN, one should focus on:

1. Network topology
2. Number of layers in the networks
3. Number of neurons or nodes per layer
4. Learning algorithm to be adopted (in the supervised case only)
5. Number of iterations per pattern during training

6. Number of calculations per iteration
7. Speed to recall a pattern
8. Network performance
9. Network plasticity (i.e., the number of neurons failing and the degree of functionality of the ANN)
10. Network capacity, or the maximum pattern signals that the network can recall
11. Degree of adaptability of the ANN (i.e., to what extent the ANN is able to adapt itself after training)
12. Bias terms (occationally set a priori to some fixed value, such as +1)
13. Threshold terms (occasionally set to some a priori fixed value, such as 0 or 1)
14. Boundaries of the synaptic weights (for best performance and noise immunity, boundaries should be determined based on the actual implementation of the ANN)
15. Choice of the nonlinearity function and the range of operation of the neuron.
16. Network noise immunity (i.e.,the degree of corruption of an input stimulus signal or the degree of signal loss (i.e., partial signal) that produces the desired target output pattern)
17. Steady-state or final values of the synaptic weights (this is the program of the ANN)

Mathematical modeling of an ANN is an attempt to describe network architecture. Kartalopoulos et al., 1997, states that mathematical analysis of an ANN describes the network's:

1. Complexity – how large the ANN can be in order to execute a task
2. Capacity – how many bits of information can be stored in the ANN
3. Paradigms choice – which ANN implementation is more suitable for the application
4. Performance – which ANN performs best
5. Learning efficiency – how fast an ANN "learns"
6. Response – how fast an ANN provides an output from the time a stimulus is present
7. Reliability – whether the ANN can reach the same desired solution for the same stimulus
8. Noise sensitivity – how accurately an ANN provides the desired output at the presence of noise
9. Failure sensitivity – how accurately an ANN associates if it partially fails

CHAPTER 3: SYSTEM DESIGN

Building Environment

The research conducted within this dissertation first began as part of a research grant awarded on behalf of Miami-Dade County's General Services Administration, Enterprise Technology Services Department, and Office of Sustainability to the University of Miami's Department of Industrial Engineering. This grant was an extension of the American Recovery and Reinvestment Act which included federal funds (appropriated by the Obama Administration and the U.S. Congress in 2009) awarded to local municipalities and earmarked for green energy initiatives. Part of the federal funding awarded to Miami-Dade County included $12.5 million for 13 different sustainability projects. In an effort to reduce energy consumption, Miami-Dade County planned to employ advanced smart meter technologies throughout several county-owned and operated buildings. In conjunction with the smart meter hardware installation, Miami-Dade County awarded a grant to the University of Miami with the following milestones for several major county buildings:

1. Install electrical data-logging sensors at the main service entrance and motor control center levels to monitor building energy consumption.
2. Perform comprehensive building energy assessment
3. Verify and validate new smart meter hardware
4. Provide education and training for electrical data-logging
5. Suggest methods for energy use behavior modification

While the University of Miami, Department of Industrial Engineering studied multiple county buildings under the awarded grant, the research conducted in this dissertation focused on one building; The Richard E. Gerstein Justice Building (REGJB) – Figure 3-1. Located at 1351 North West 12th Street, Miami, FL 33125, this Miami-Dade County building serves the community with the following services:

1. Criminal Court – Felony Cases
2. District Court – Hearings
3. Jury Service
4. Traffic Violations and Misdemeanor Cases

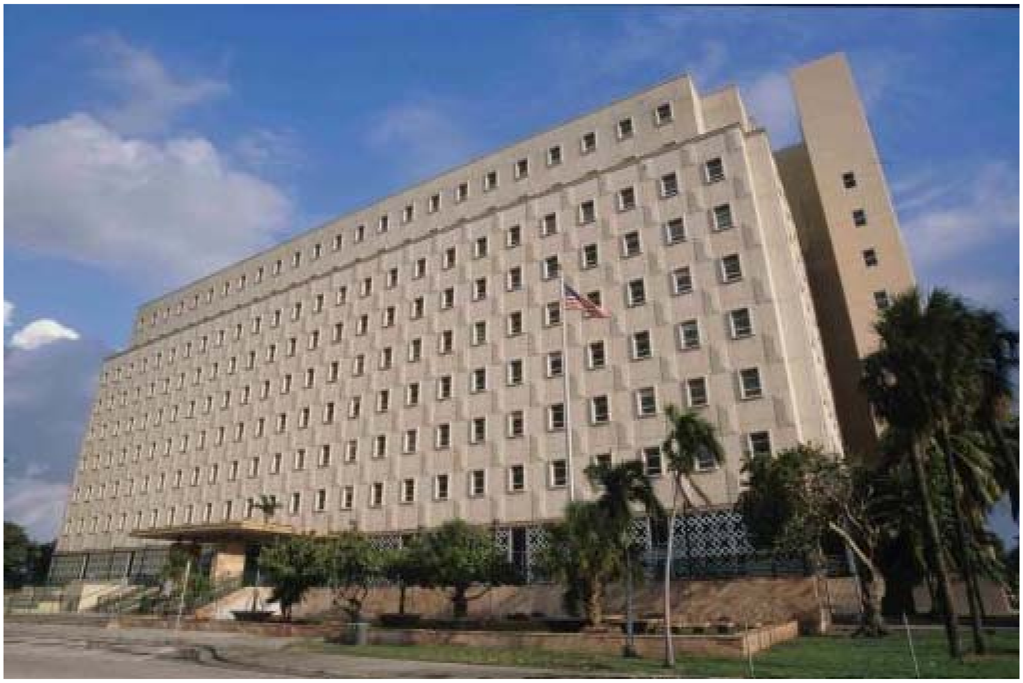


Figure 3–1 – REGJB Miami, FL.

Constructed at a cost of $8.5 million, when it opened in 1962, it was known as the Metro Justice Building. In 1992 it was renamed to honor the late Richard E. Gerstein; Dade County's 6-term State Attorney who held office from 1956 to 1977. Before being formally educated or beginning his career, Gerstein served in World War II, and for his valor he received a Distinguished Flying Cross and a Purple Heart. At the end of the war, he earned his undergraduate and law degrees from The University of Miami. He then went on, according to The Miami Herald, to become, "the most powerful, controversial, durable and well-known politician in [Miami] Dade County." Gerstein gained national prominence in 1973 by winning the first conviction in the Watergate scandal that would eventually force the resignation of President Richard Nixon. Gerstein's conviction of Bernard Barker on money-laundering charges in Miami linked the White House directly with the 1972 burglary at the Democratic Party's headquarters in the Watergate complex in Washington, D.C. During his career, he prosecuted several other high-profile case while relentless crusading against deep corruption which was at that time rampant throughout the Miami political and regulatory landscape.

The REGJB is built on a 2.11 acre parcel located within a few miles of downtown metropolitan Miami. It is an eleven story building comprised of approximately 500,000 square feet of internal air-conditioned space described in Table 3-1.

Table 3-1 REGJB Composition by Floor.

Floor(s)	Usage
Basement	Parking, Electrical Service Entrance, Chillers' Mechanicals
1	Main Entrance, Building Management, Parking Violations, Cafeteria
2-4	Courtrooms
5-7	Judges Chambers
8-9	Administrative
10	HVAC Mechanicals

Due to its size, the REGJB utilizes industrial sized equipment to service the building. The building's heating, ventilation, and air conditioning (HVAC) system includes one 200-ton centrifugal chiller, two 200-ton screw chillers, two 300-ton cooling towers, ten 75-horsepower (HP) air handling units, four 25-HP chiller water pumps, and four 7.5-HP air compressors for pneumatic control. The building is traversed by three main elevators, two freight elevators, and two sets of escalators. There are also four 5-HP water pumps in the building. Figures 3-2 and 3-3 shows the building's three main chillers.

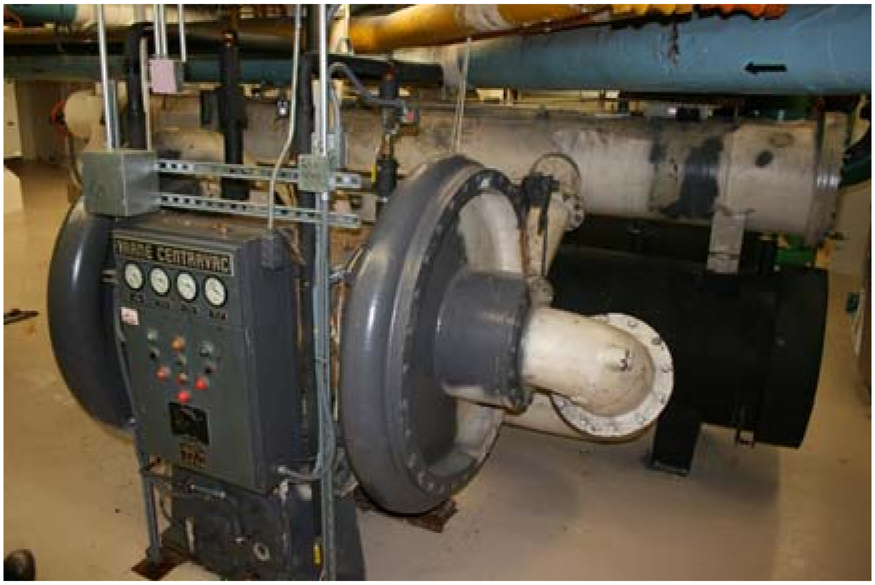


Figure 3-2 200-ton centrifugal chiller.

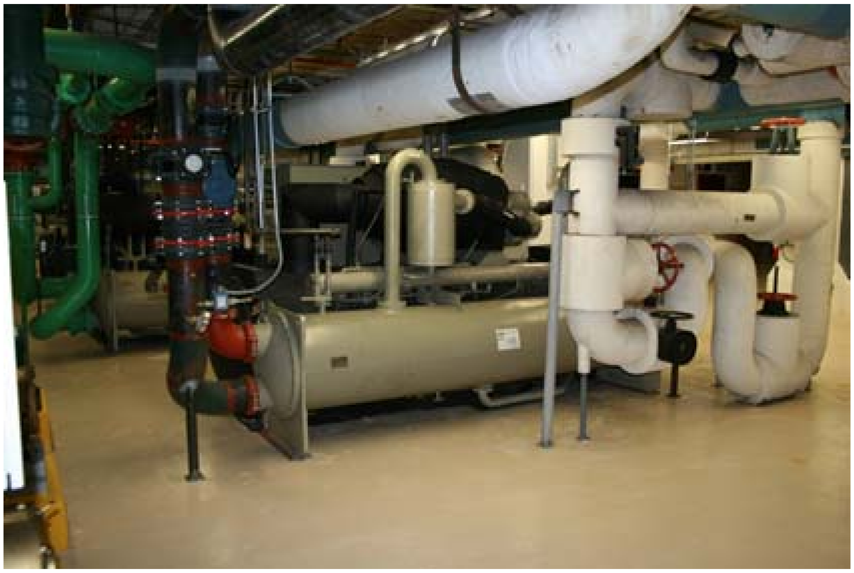


Figure 3-3 200-ton screw chiller.

Figure 3-4 demonstrates one of the chiller pump motors while Figure 3-4 depicts 1 of 2 200-ton cooling towers for the building's HVAC system.

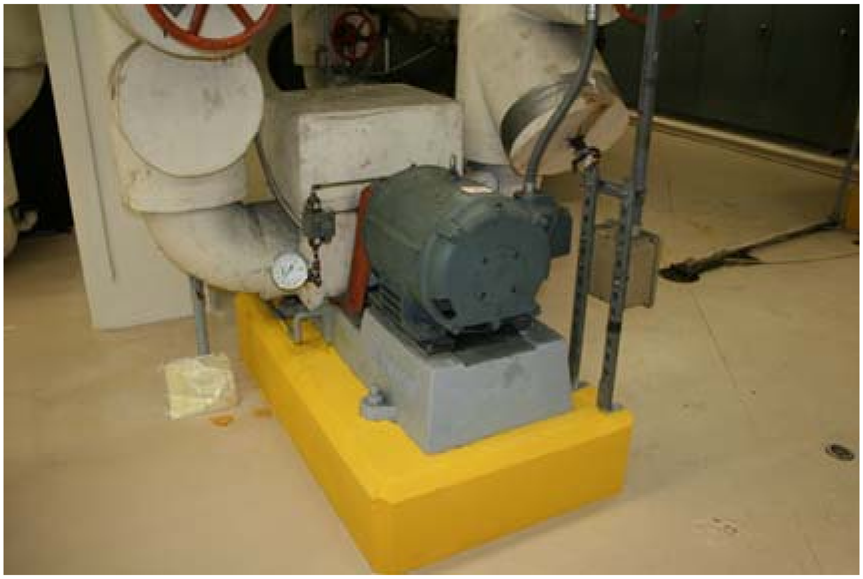


Figure 3-4 25 HP chiller pump motor.

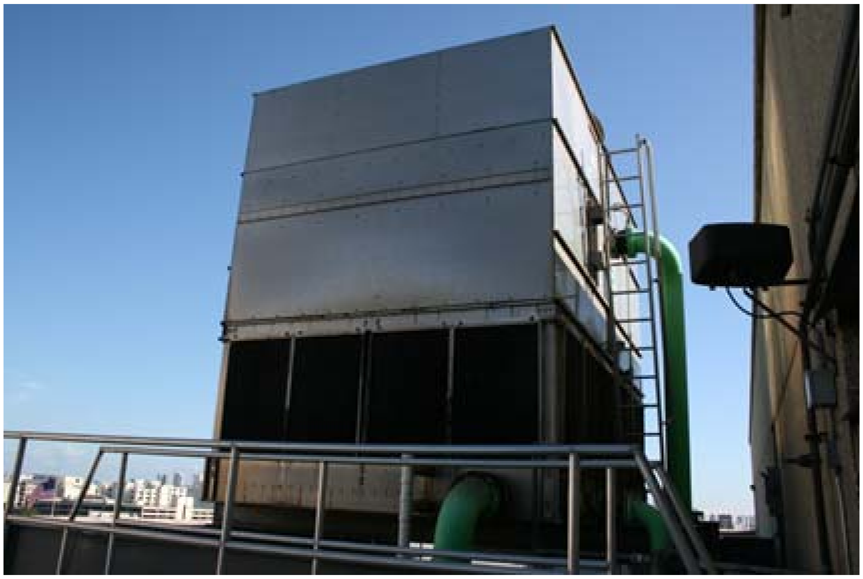


Figure 3-5 200-ton HVAC cooling tower.

Figure 3-6 shows one of ten 75 HP HVAC air handling units. And, Figure 3-7 depicts the four 7.5HP air compressors with two compressors mounted on each of two 240 gallon air storage receivers.

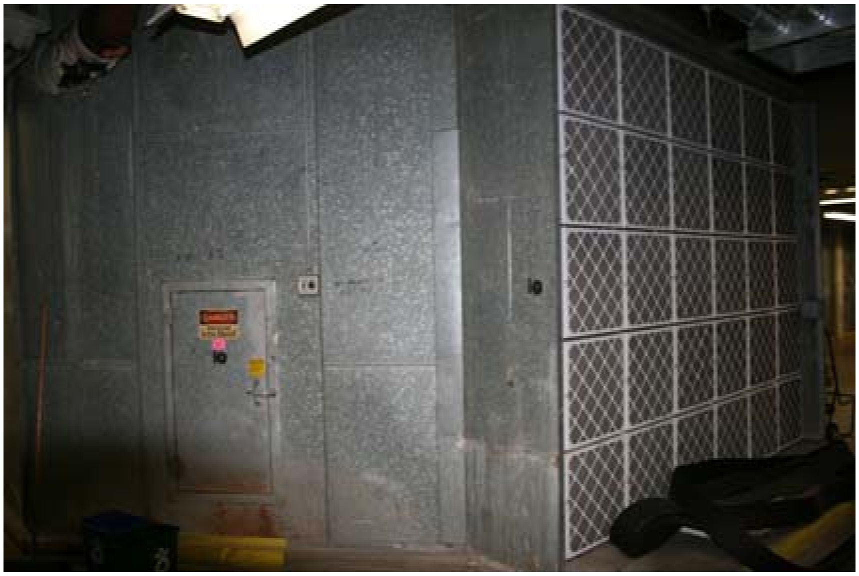


Figure 3-6 75 HP HVAC air handling unit.

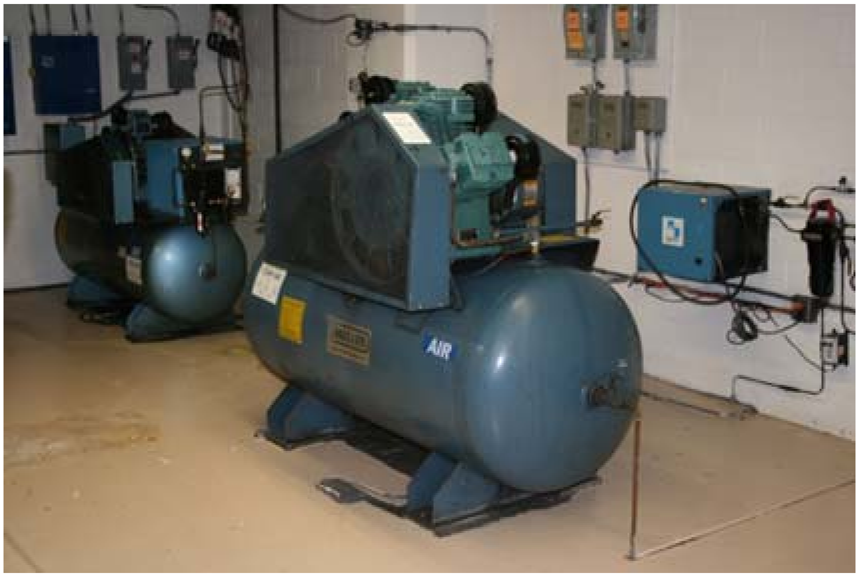


Figure 3-7 4 x 7.5 HP air compressors.

Electrical Distribution

The building includes three main electrical service entrances which are three phase, three wire Delta configuration, 480 Volt. Two entrances supply 3000 amp service, while one supplies 1200 amp service. For Delta configuration, as opposed to WYE configuration, there is no neutral; therefore, the voltage is always a line to line measurement. One advantage of a Delta configuration is that it does not have a neutral and therefore if a phase winding should fail the phase voltage at the load remains constant. Figure 3-2 depicts this delta configuration.

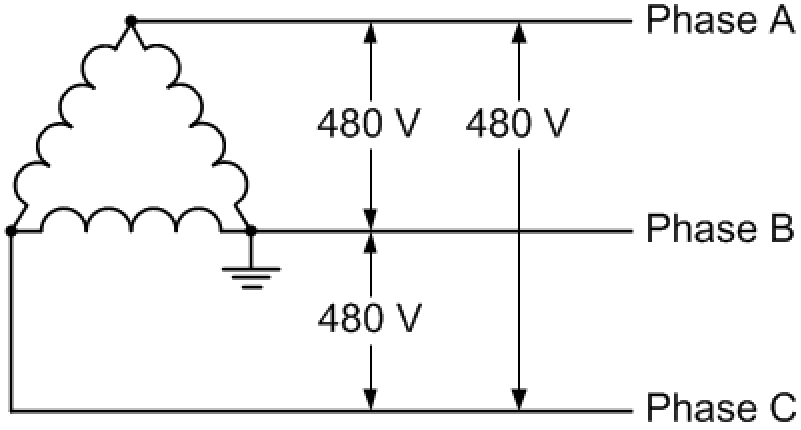


Figure 3-8 – 480 Volt Delta Configuration Circuit.

The main service entrances also supply power to two motor control centers which support the HVAC system. One is rated at 1600 amp capacity while the second is rated at 600 amp capacity.

The original service entrance, Main Service Entrance #1 (MSE1), installed when the building was first built services the entire building excluding the building's HVAC and backup generator systems. This service entrance provides lighting and receptacle power to all floors of the building. It also powers the elevators and escalators. It includes three large transformers which convert voltage from 480 volt to 208/120 volt. MSE1 is shown in Figure 3-3.

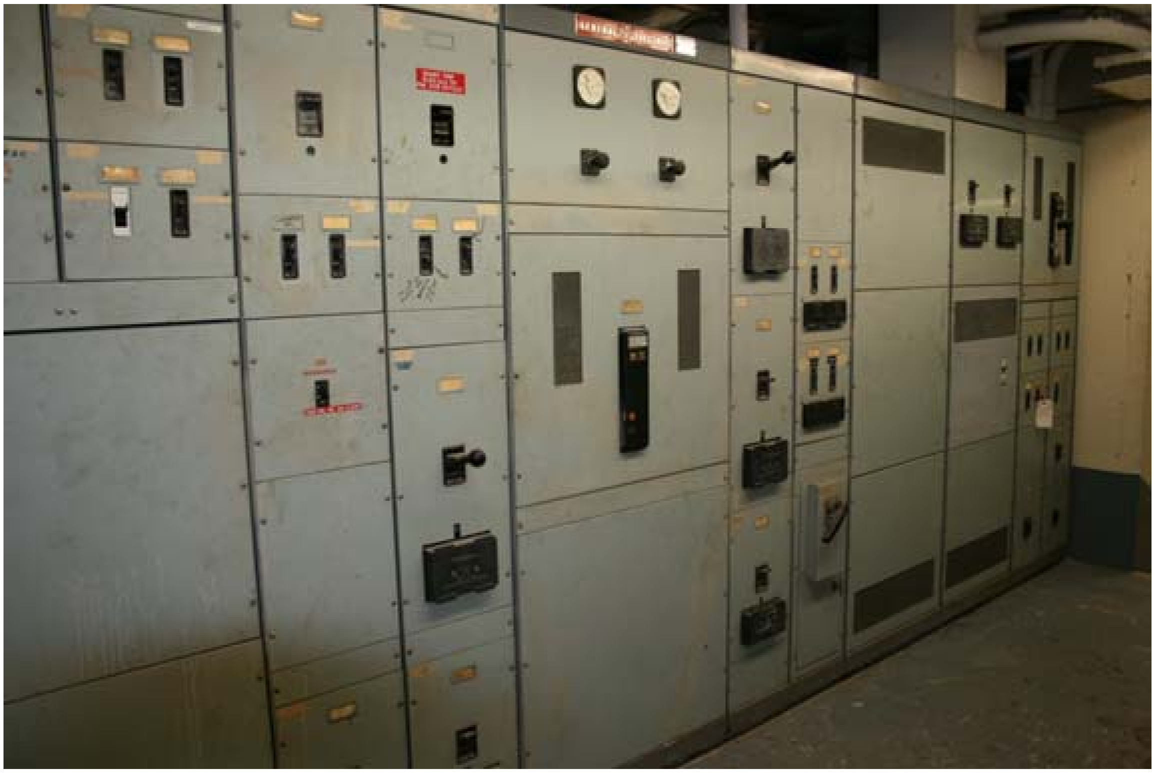


Figure 3-9 – Main Service Entrance #1 (480V, 3000amp).

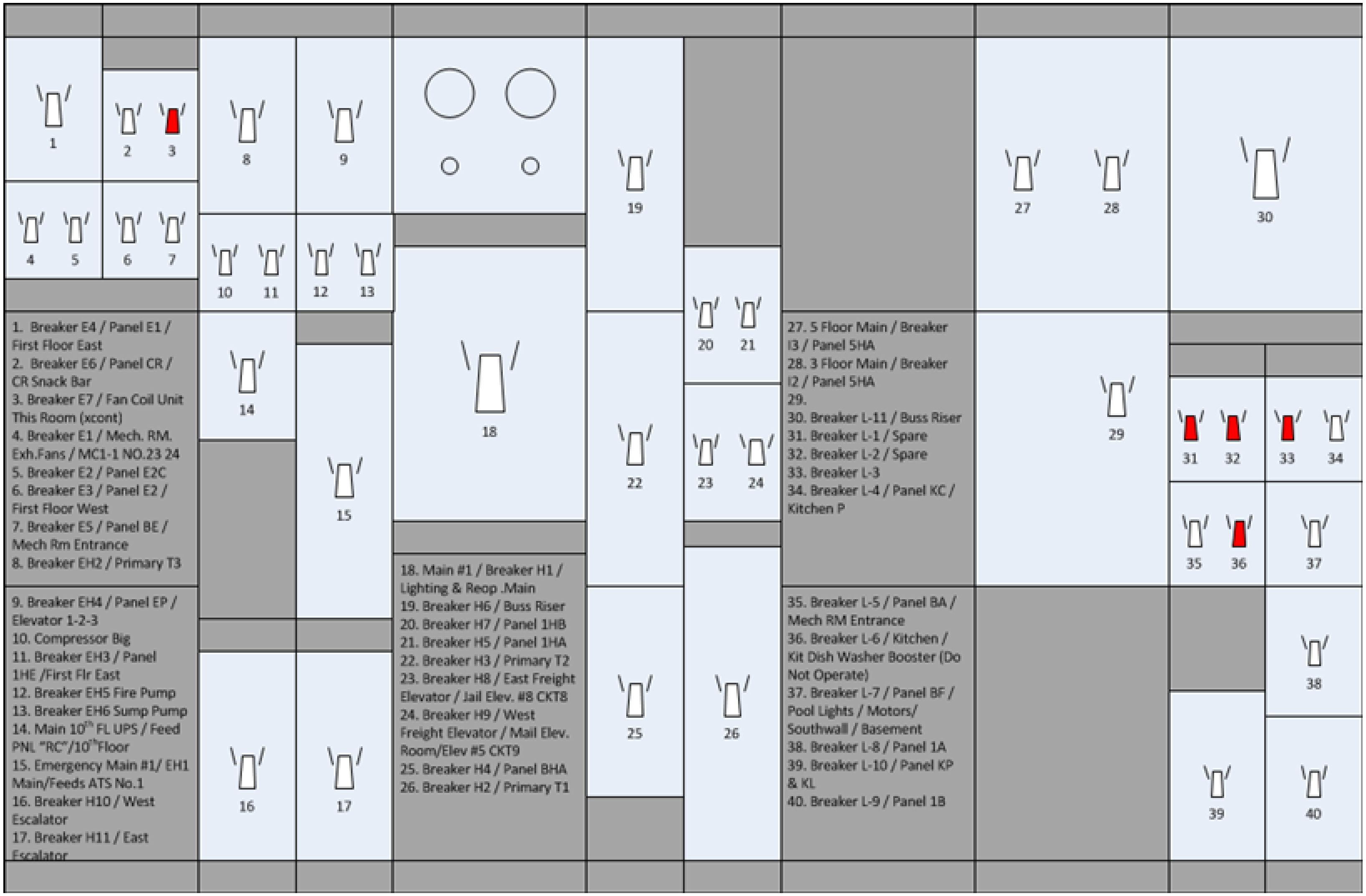

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
1. Breaker E4 / Panel E1 / First Floor East
2. Breaker E6 / Panel CR / CR Snack Bar
3. Breaker E7 / Fan Coil Unit This Room (xcont)
4. Breaker E1 / Mech. RM. Exh.Fans / MC1-1 NO.23 24
5. Breaker E2 / Panel E2C
6. Breaker E3 / Panel E2 / First Floor West
7. Breaker E5 / Panel BE / Mech Rm Entrance
8. Breaker EH2 / Primary T3
9. Breaker EH4 / Panel EP / Elevator 1-2-3
10. Compressor Big
11. Breaker EH3 / Panel 1HE /First Flr East
12. Breaker EH5 Fire Pump
13. Breaker EH6 Sump Pump
14. Main 10th FL UPS / Feed PNL "RC"/10thFloor
15. Emergency Main #1/ EH1 Main/Feeds ATS No.1
16. Breaker H10 / West Escalator
17. Breaker H11 / East Escalator
18. Main #1 / Breaker H1 / Lighting & Reop .Main
19. Breaker H6 / Buss Riser
20. Breaker H7 / Panel 1HB
21. Breaker H5 / Panel 1HA
22. Breaker H3 / Primary T2
23. Breaker H8 / East Freight Elevator / Jail Elev. #8 CKT8
24. Breaker H9 / West Freight Elevator / Mail Elev. Room/Elev #5 CKT9
25. Breaker H4 / Panel BHA
26. Breaker H2 / Primary T1
27. 5 Floor Main / Breaker I3 / Panel 5HA
28. 3 Floor Main / Breaker I2 / Panel 5HA
29.
30. Breaker L-11 / Buss Riser
31. Breaker L-1 / Spare
32. Breaker L-2 / Spare
33. Breaker L-3
34. Breaker L-4 / Panel KC / Kitchen P
35. Breaker L-5 / Panel BA / Mech RM Entrance
36. Breaker L-6 / Kitchen / Kit Dish Washer Booster (Do Not Operate)
37. Breaker L-7 / Panel BF / Pool Lights / Motors/ Southwall / Basement
38. Breaker L-8 / Panel 1A
39. Breaker L-10 / Panel KP & KL
40. Breaker L-9 / Panel 1B

Figure 3-10 Main Service Entrance #1 digital map (disconnected circuits shown in red).

Main Service Entrance #2 (MSE2), installed relatively recently and before this research commenced, services the entire HVAC system including the chillers and Motor Control Center #1 (MCC1) located on the basement floor as well as Motor Control Center #2 (MCC2) located on the tenth floor. MSE2 is shown in Figure 3-4. Main Service Entrance #3 (MSE3) services the emergency backup generator and its support equipment. MSE3 is shown in Figure 3-5. MCC1, seen in Figure 3-6, powers ancillary HVAC equipment located in the basement. MCC1 serves chiller and water pumps in the basement while MCC2 serves the air handlers and cooling towers on the 10th floor.

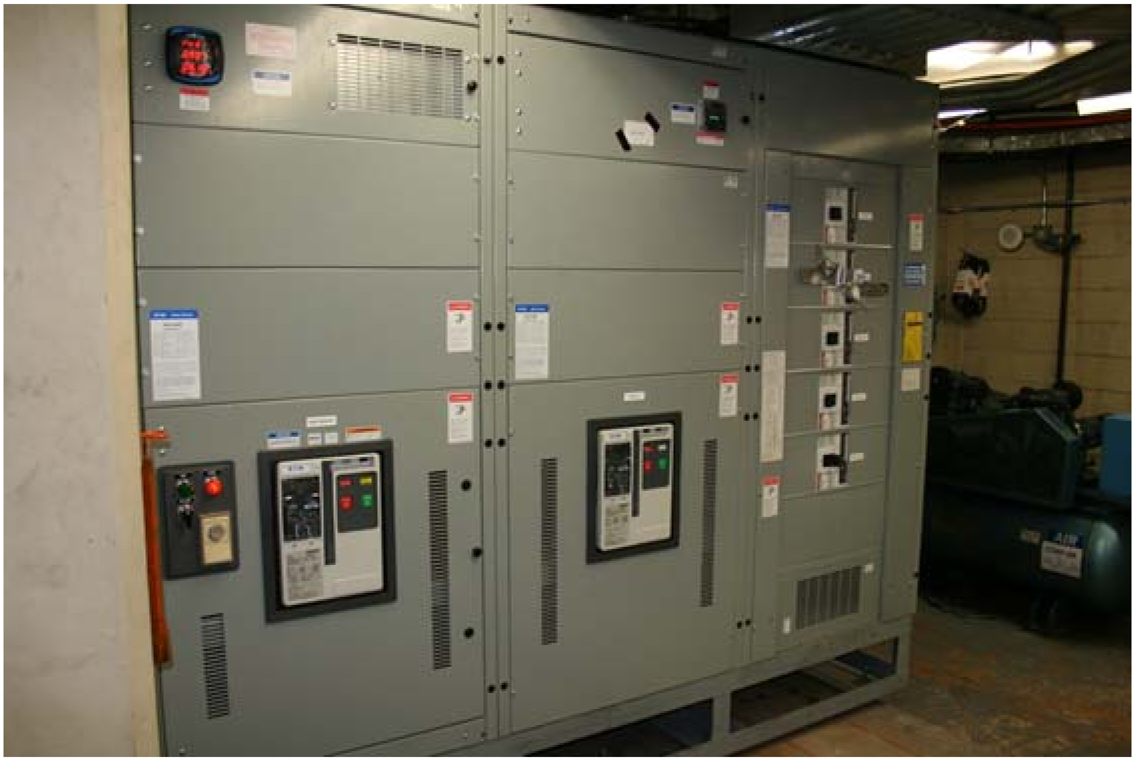


Figure 3-11 – Main Service Entrance #2 (480V, 3000amp).

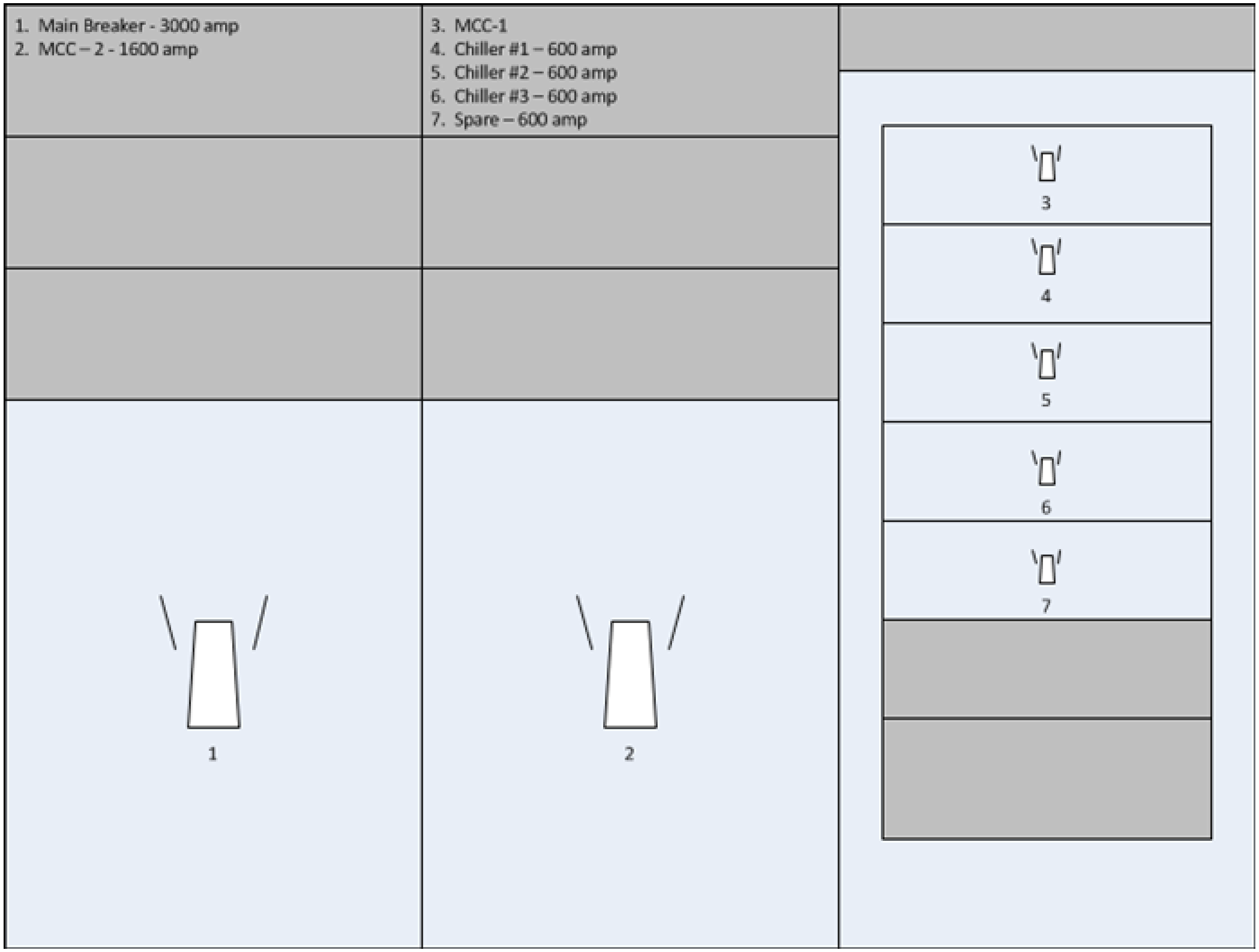


Figure 3-12 Main Service Entrance #2 digital map.

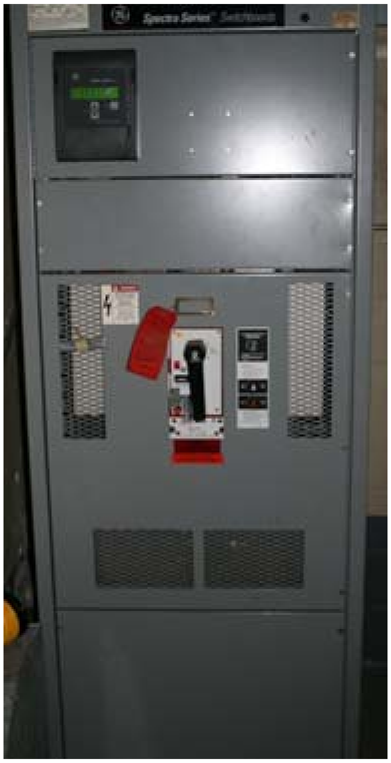


Figure 3-13 – Main Service Entrance #3 (480V, 1200amp).

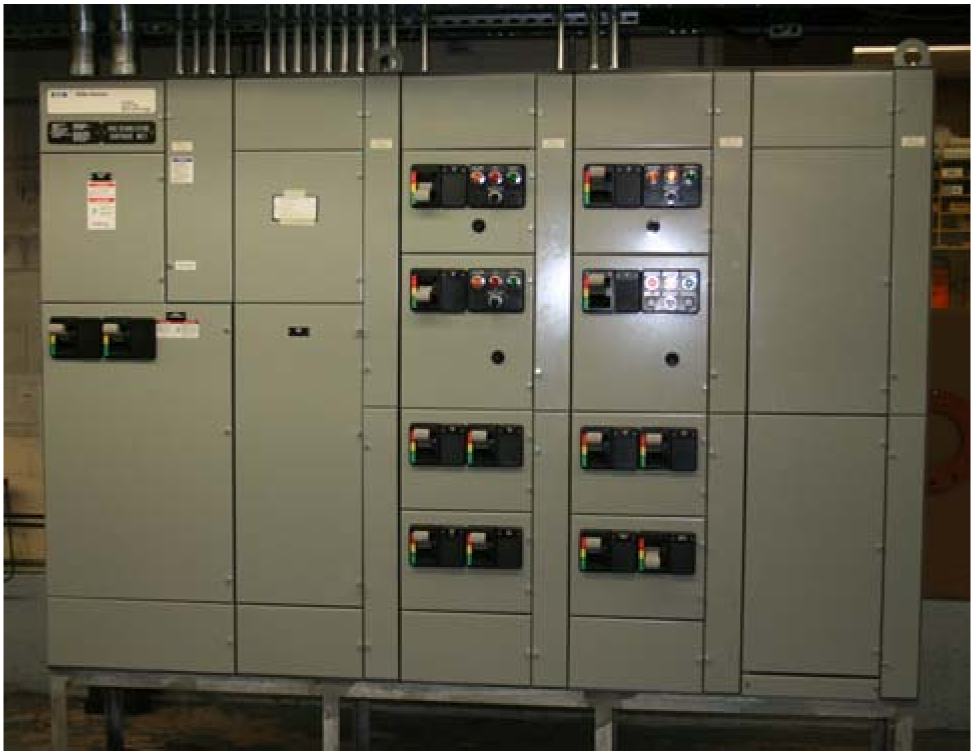


Figure 3-14 – Motor Control Center #1 (480V, 600amp).

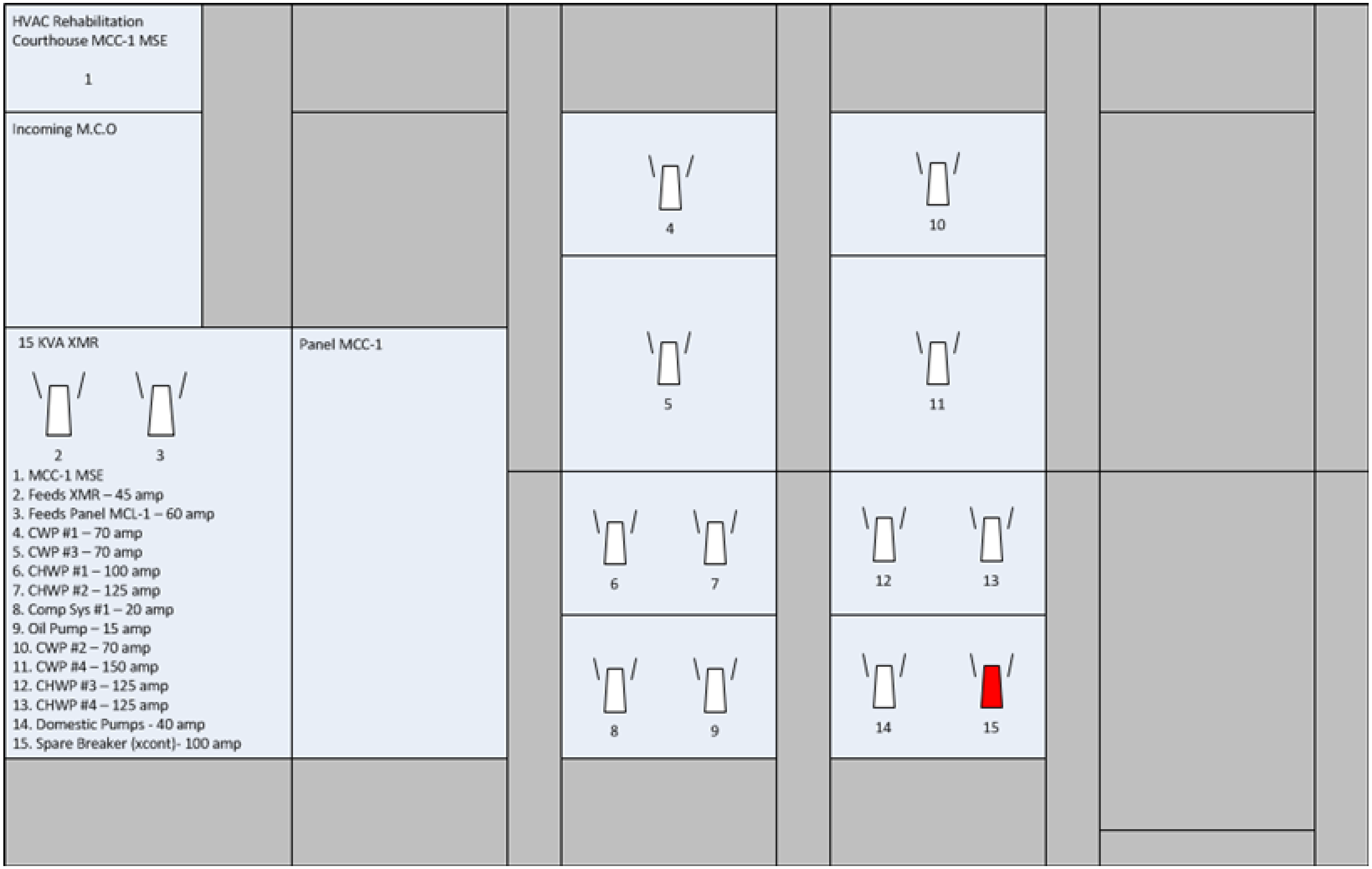


Figure 3-15 Motor Control Center #1 digital map (disconnected circuits shown in red).

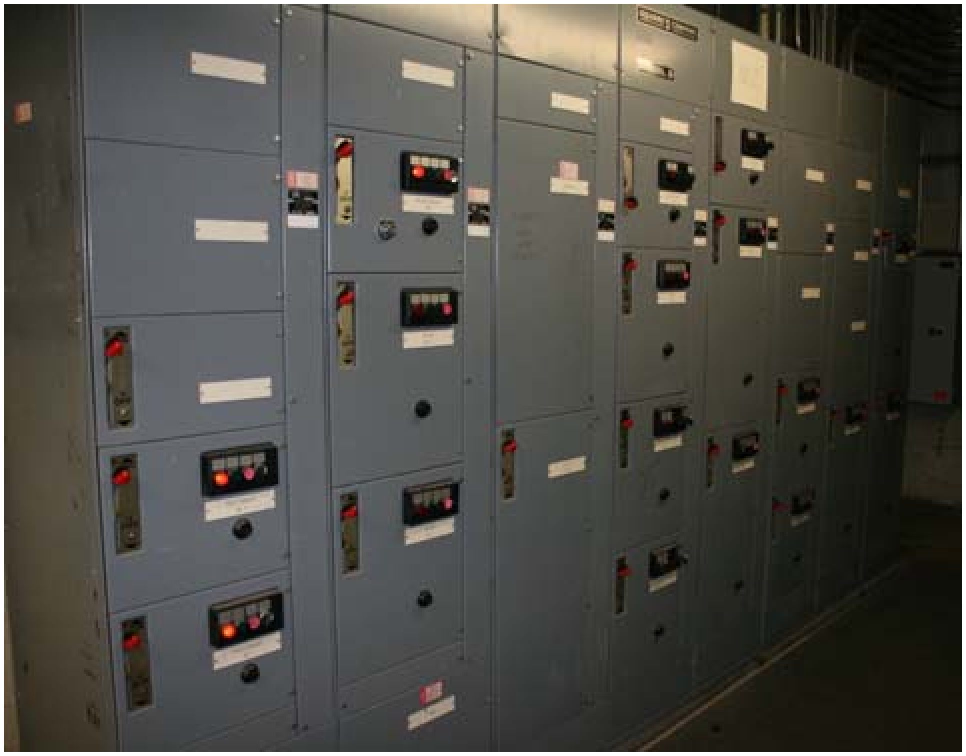


Figure 3-16 – Motor Control Center #2 (480V, 1600amp).

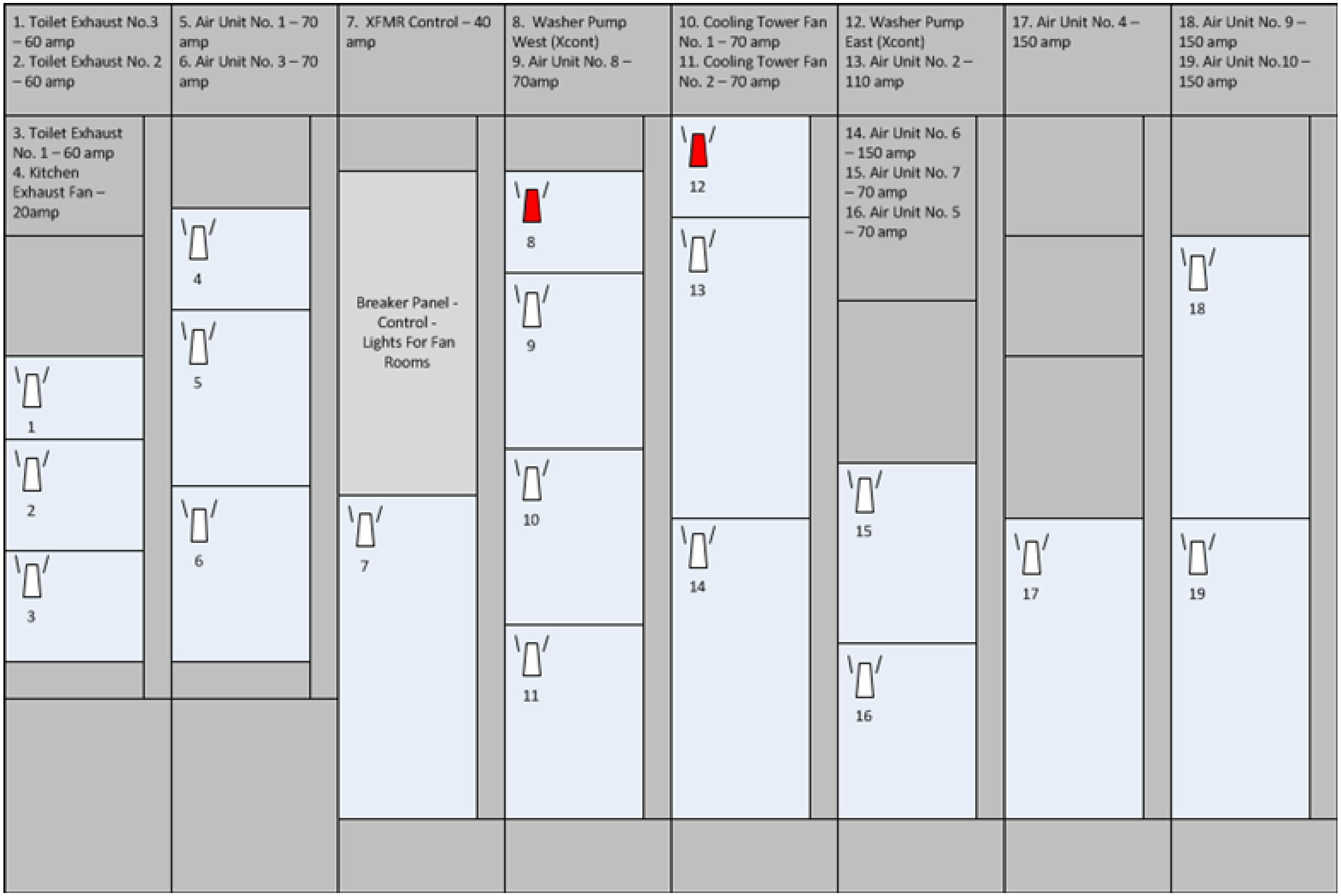

1. Toilet Exhaust No.3 – 60 amp
2. Toilet Exhaust No. 2 – 60 amp
5. Air Unit No. 1 – 70 amp
6. Air Unit No. 3 – 70 amp
7. XFMR Control – 40 amp
8. Washer Pump West (Xcont)
9. Air Unit No. 8 – 70amp
10. Cooling Tower Fan No. 1 – 70 amp
11. Cooling Tower Fan No. 2 – 70 amp
12. Washer Pump East (Xcont)
13. Air Unit No. 2 – 110 amp
17. Air Unit No. 4 – 150 amp
18. Air Unit No. 9 – 150 amp
19. Air Unit No.10 – 150 amp
3. Toilet Exhaust No. 1 – 60 amp
4. Kitchen Exhaust Fan – 20amp
1
2
3
4
5
6
Breaker Panel - Control - Lights For Fan Rooms
7
8
9
10
11
12
13
14
14. Air Unit No. 6 – 150 amp
15. Air Unit No. 7 – 70 amp
16. Air Unit No. 5 – 70 amp
15
16
17
18
19

Figure 3-17 Motor Control Center #2 digital map (disconnected circuits shown in red).

Table 3-2 shows all circuits for MSE1. Circuits highlighted in red have been disconnected and are not in use. Circuit descriptions are as labeled on site.

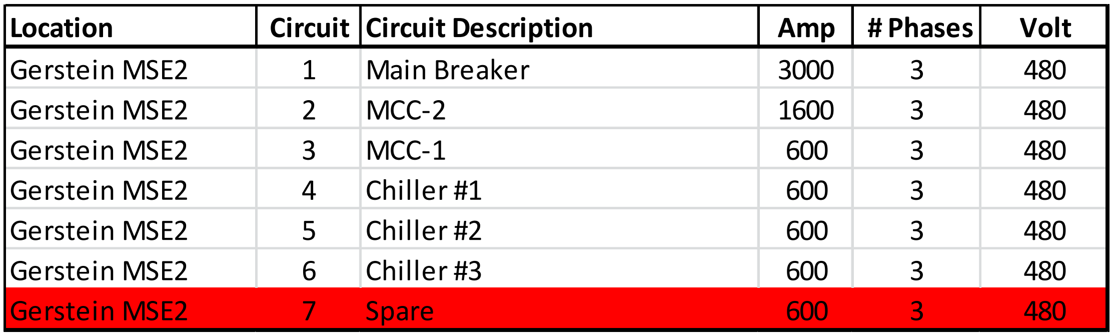
Table 3-2 – MSE1 Circuits.

Location	Circuit	Circuit Description	Amp	# Phases	Volt
Gerstein MSE1	1	Breaker E4 / Panel E1 / First Floor East	100	3	208/120
Gerstein MSE1	2	Breaker E6 / Panel CR / CR Snack Bar	50	3	208/120
Gerstein MSE1	3	Breaker E7 / Fan Coil Unit This Room (Xcont)	15	3	208/120
Gerstein MSE1	4	Breaker E1 / Mech.RM.Exh.Fans / MC1-1 NO.23 24	50	3	208/120
Gerstein MSE1	5	Breaker E2 / Panel E2C	50	3	208/120
Gerstein MSE1	6	Breaker E3 / Panel E2 / First Floor West	125	3	208/120
Gerstein MSE1	7	Breaker E5 / Panel BE / Mech Rm Entrance	70	3	208/120
Gerstein MSE1	8	Breaker EH2 / Primary T3 High Side	200	3	480
Gerstein MSE1	9	Breaker EH4 / Panel EP / Elevator 1-2-3	225	3	480
Gerstein MSE1	10	Compressor Big	50	3	480
Gerstein MSE1	11	Breaker EH3 / Panel 1HE / First Flr East	100	3	480
Gerstein MSE1	12	Breaker EH5 Fire Pump	70	3	480
Gerstein MSE1	13	Breaker EH6 Sump Pumps	50	3	480
Gerstein MSE1	14	Main 10th FL UPS / Feed PNL "RC" / 10th Floor	50	3	480
Gerstein MSE1	15	Emergency Main #1 / EH1 Main / Feeds ATS No.1	600	3	480
Gerstein MSE1	16	Breaker H10 / West Escalator	90	3	480
Gerstein MSE1	17	Breaker H11 / East Escalator	90	3	480
Gerstein MSE1	18	Main #1 / Breaker H1 / Lighting & Reop.Main	3000	3	480
Gerstein MSE1	19	Breaker H6 / Buss Riser	600	3	480
Gerstein MSE1	20	Breaker H7 / Panel 1HB	70	3	480
Gerstein MSE1	21	Breaker H5 / Panel 1HA	100	3	480
Gerstein MSE1	22	Breaker H3 / Primary T2 High Side	600	3	480
Gerstein MSE1	23	Breaker H8 / East Freight Elevator / Jail Elev. #8 CKT8	30	3	480
Gerstein MSE1	24	Breaker H9 / West Freight Elevator / Mail Elev. Room / Elev #5 CKT9	30	3	480
Gerstein MSE1	25	Breaker H4 / Panel BHA	200	3	480
Gerstein MSE1	26	Breaker H2 / Primary T1 High Side	1000	3	480
Gerstein MSE1	27	5 Floor Main / Breaker I3 / Panel 5HA	400	3	480
Gerstein MSE1	28	3 Floor Main / Breaker I2 / Panel 5HA	400	3	480
Gerstein MSE1	29	No Label	225	3	208/120
Gerstein MSE1	30	Breaker L-11 / Buss Riser / T1 Low Side	1600	3	208/120
Gerstein MSE1	31	Breaker L-1 / Spare	100	3	208/120
Gerstein MSE1	32	Breaker L-2 / Spare	100	3	208/120
Gerstein MSE1	33	Breaker L-3	100	3	208/120
Gerstein MSE1	34	Breaker L-4 / Panel KC / Kitchen P	100	3	208/120
Gerstein MSE1	35	Breaker L-5 / Panel BA / Mech RM Entrance	100	3	208/120
Gerstein MSE1	36	Breaker L-6 / Kitchen / Kit Dish Washer Booster (Do Not Operate)		3	208/120
Gerstein MSE1	37	Breaker L-7 / Panel BF / Pool Lights / Motors / Southwall / Basement	100	3	208/120
Gerstein MSE1	38	Breaker L-8 / Panel 1A	175	3	208/120
Gerstein MSE1	39	Breaker L-10 / Panel KP & KL	400	3	208/120
Gerstein MSE1	40	Breaker L-9 / Panel 1B	200	3	208/120
Gerstein MSE1	41	T2 Low Side	1000	3	208/120
Gerstein MSE1	42	T3 Low Side	1000	3	208/120
Gerstein MSE1	43	Plant T	400	3	480
Gerstein MSE1	44	9B - #1 Condenser Water Pump	400	3	480

Table 3-3 shows all circuits for MSE2. Circuits highlighted in red have been disconnected and are not in use. Circuit descriptions are as labeled on site. Table 3-4 shows MSE3's circuit. Table 3-5 shows circuits for MCC1. Circuits highlighted in red have been disconnected and are not in use. Circuit descriptions are as labeled on site

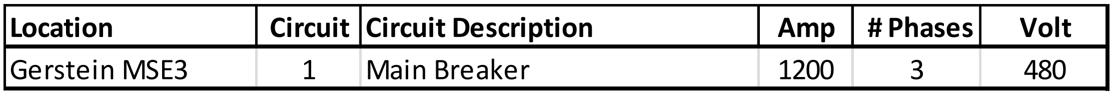
Table 3-6 shows circuits for MCC2. Circuits highlighted in red have been disconnected and are not in use. Circuit descriptions are as labeled on site

Table 3-3 – MSE2 Circuits.



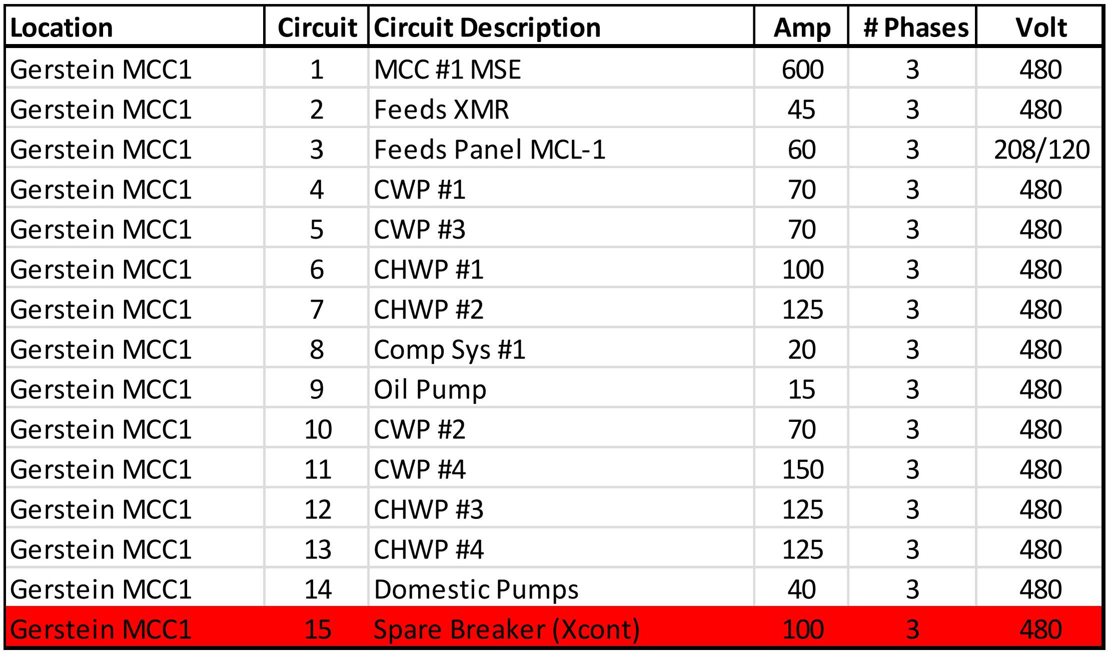
Location	Circuit	Circuit Description	Amp	# Phases	Volt
Gerstein MSE2	1	Main Breaker	3000	3	480
Gerstein MSE2	2	MCC-2	1600	3	480
Gerstein MSE2	3	MCC-1	600	3	480
Gerstein MSE2	4	Chiller #1	600	3	480
Gerstein MSE2	5	Chiller #2	600	3	480
Gerstein MSE2	6	Chiller #3	600	3	480
Gerstein MSE2	7	Spare	600	3	480

Table 3-4 – MSE3 Circuit.



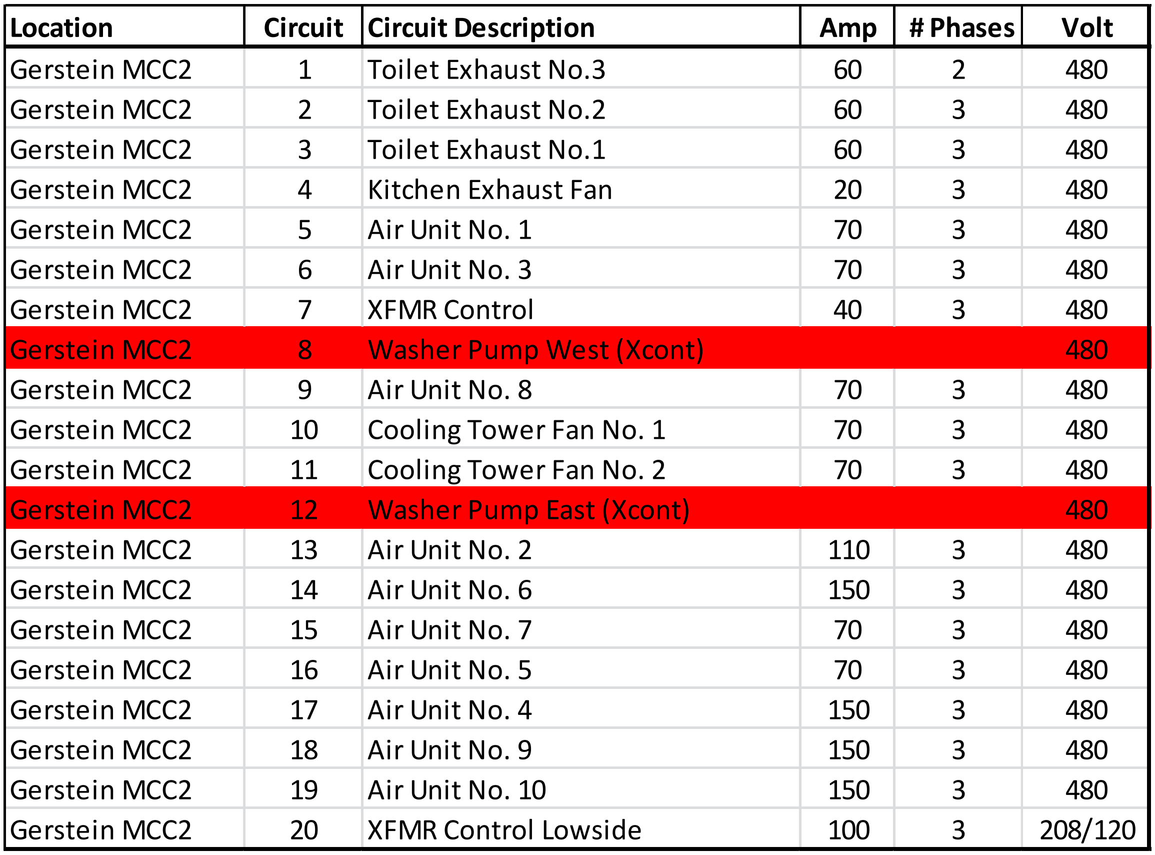
Location	Circuit	Circuit Description	Amp	# Phases	Volt
Gerstein MSE3	1	Main Breaker	1200	3	480

Table 3-5 – MCC1 Circuits.



Location	Circuit	Circuit Description	Amp	# Phases	Volt
Gerstein MCC1	1	MCC #1 MSE	600	3	480
Gerstein MCC1	2	Feeds XMR	45	3	480
Gerstein MCC1	3	Feeds Panel MCL-1	60	3	208/120
Gerstein MCC1	4	CWP #1	70	3	480
Gerstein MCC1	5	CWP #3	70	3	480
Gerstein MCC1	6	CHWP #1	100	3	480
Gerstein MCC1	7	CHWP #2	125	3	480
Gerstein MCC1	8	Comp Sys #1	20	3	480
Gerstein MCC1	9	Oil Pump	15	3	480
Gerstein MCC1	10	CWP #2	70	3	480
Gerstein MCC1	11	CWP #4	150	3	480
Gerstein MCC1	12	CHWP #3	125	3	480
Gerstein MCC1	13	CHWP #4	125	3	480
Gerstein MCC1	14	Domestic Pumps	40	3	480
Gerstein MCC1	15	Spare Breaker (Xcont)	100	3	480

Table 3-6 – MCC2 Circuits.



Location	Circuit	Circuit Description	Amp	# Phases	Volt
Gerstein MCC2	1	Toilet Exhaust No.3	60	2	480
Gerstein MCC2	2	Toilet Exhaust No.2	60	3	480
Gerstein MCC2	3	Toilet Exhaust No.1	60	3	480
Gerstein MCC2	4	Kitchen Exhaust Fan	20	3	480
Gerstein MCC2	5	Air Unit No. 1	70	3	480
Gerstein MCC2	6	Air Unit No. 3	70	3	480
Gerstein MCC2	7	XFMR Control	40	3	480
Gerstein MCC2	8	Washer Pump West (Xcont)			480
Gerstein MCC2	9	Air Unit No. 8	70	3	480
Gerstein MCC2	10	Cooling Tower Fan No. 1	70	3	480
Gerstein MCC2	11	Cooling Tower Fan No. 2	70	3	480
Gerstein MCC2	12	Washer Pump East (Xcont)			480
Gerstein MCC2	13	Air Unit No. 2	110	3	480
Gerstein MCC2	14	Air Unit No. 6	150	3	480
Gerstein MCC2	15	Air Unit No. 7	70	3	480
Gerstein MCC2	16	Air Unit No. 5	70	3	480
Gerstein MCC2	17	Air Unit No. 4	150	3	480
Gerstein MCC2	18	Air Unit No. 9	150	3	480
Gerstein MCC2	19	Air Unit No. 10	150	3	480
Gerstein MCC2	20	XFMR Control Lowside	100	3	208/120

Hardware Sensors and Design

Electrical consumption data was captured for every building circuit using National Instruments data acquisition hardware and Magnelab current sensor transformers. At each circuit breaker a current sensor transformer was installed around each phase of power. A current transformer (CT) has a primary winding, a magnetic core, and a secondary winding. A magnetic field in the core is produced with alternating current flows through the primary winding thereby inducing a current in the secondary winding circuit. Transformers are designed to ensure that primary and secondary circuits are efficiently coupled, so that there is an accurate relationship between the primary and secondary current.

The most common design of CT consists of a length of wire wrapped many times around a silicon steel ring passed over the circuit being measured. The CT's primary circuit therefore consists of a single 'turn' of conductor, with a secondary of many tens or hundreds of turns. The primary winding may be a permanent part of the current transformer, with a heavy copper bar to carry current through the magnetic core. Window-type current transformers (aka zero sequence current transformers, or ZSCT) are also common, which can have circuit cables run through the middle of an opening in the core to provide a single-turn primary winding. When conductors passing through a CT are not centered in the circular (or oval) opening, slight inaccuracies may occur. Shapes and sizes can vary depending on the end user or switchgear manufacturer. Typical examples of low voltage single ratio metering current transformers are either ring type or plastic moulded case. High-voltage current transformers are mounted on porcelain bushings to insulate them from ground. Some CT configurations slip around the bushing of a high-voltage transformer or circuit breaker, which automatically centers the conductor inside the CT window.

The primary circuit is largely unaffected by the insertion of the CT. The rated secondary current is commonly standardized at 1 or 5 amperes. For example, a 4000:5 CT would provide an output current of 5 amperes when the primary was passing 4000 amperes. The secondary winding can be single ratio or multi ratio, with five taps being common for multi ratio CTs. The load, or burden, of the CT should be of low resistance.

If the voltage time integral area is higher than the core's design rating, the core goes into saturation towards the end of each cycle, distorting the waveform and affecting accuracy.

The MagNelab current sensor transformers used for this research are single turn CTs with burden resistors to product a low-voltage output. Figure 3-9 shows a basic current sensor transformers diagram together with input current vs. output voltage.

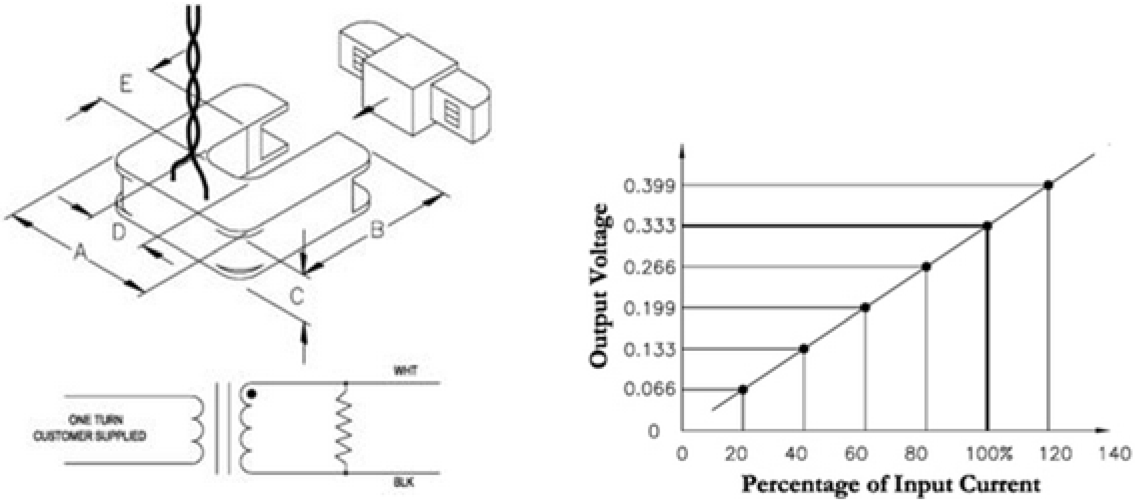


Figure 3-18 current sensor transformers diagram and input current vs. output voltage.

For example, a 400 amp rated current sensor transformer would output 0.333 Volt across its two differential lead wires (black and white) when the 400amp current is detected through the sensor. On the same 400 amp sensor, if 160 amps of power were being sensed, the output voltage would be 0.133 Volts. For each circuit, an amp sensor rated at or close to the breaker's amp rating was used. Figure 3-9 illustrates an example of how the current sensor transformers were installed on one particular circuit.

By employing National Instruments data acquisition hardware, the voltages produced by the current sensor transformers were accurately metered. Specifically, National Instrument's Compact Real Time Input & Output (cRIO) 9022 and 9014 devices, configurable embedded control and acquisition systems, together with National Instruments C-Series 9206 16 channel analog input module and 9211 4 channel thermocouple input module, were used for data capture. A four slot module chassis was used with the cRIO 9022 while a 9144 EtherCAT slave chassis connected to the cRIO 9022 served to connect additional input modules. The cRIO 9014 was connected to an eight slot chassis. Figures 3-16 through 3-20 show the cRIOs, cRIO chassis, cRIO EtherCAT slave chassis, 16 channel voltage input module, and the 4 channel thermocouple differential analog input module. The thermocouple module was connected to two thermocouples that metered interior and exterior building temperatures.

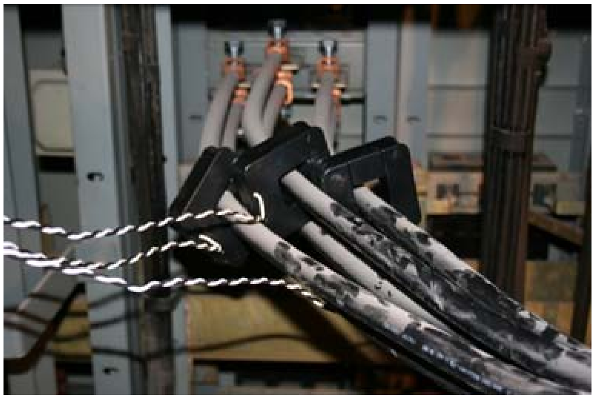


Figure 3-19 400 amp current sensor transformers phases a,b,& c..

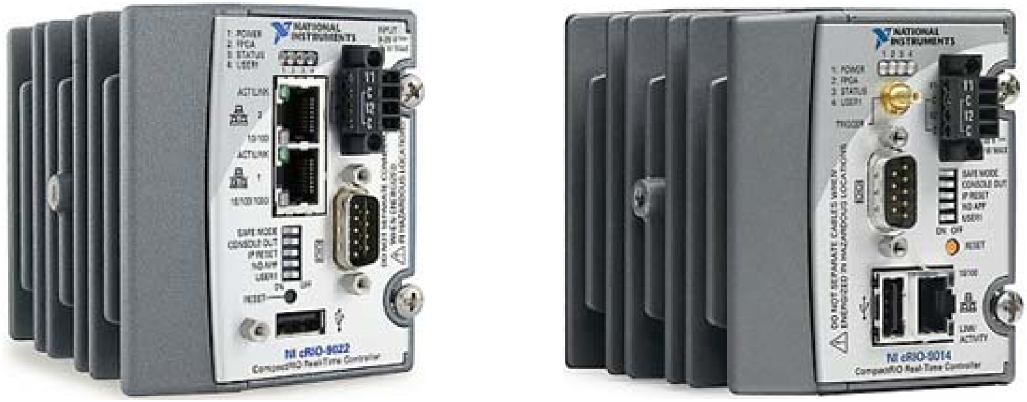


Figure 3-20 National Instruments cRIO 9022 & cRIO 9014.

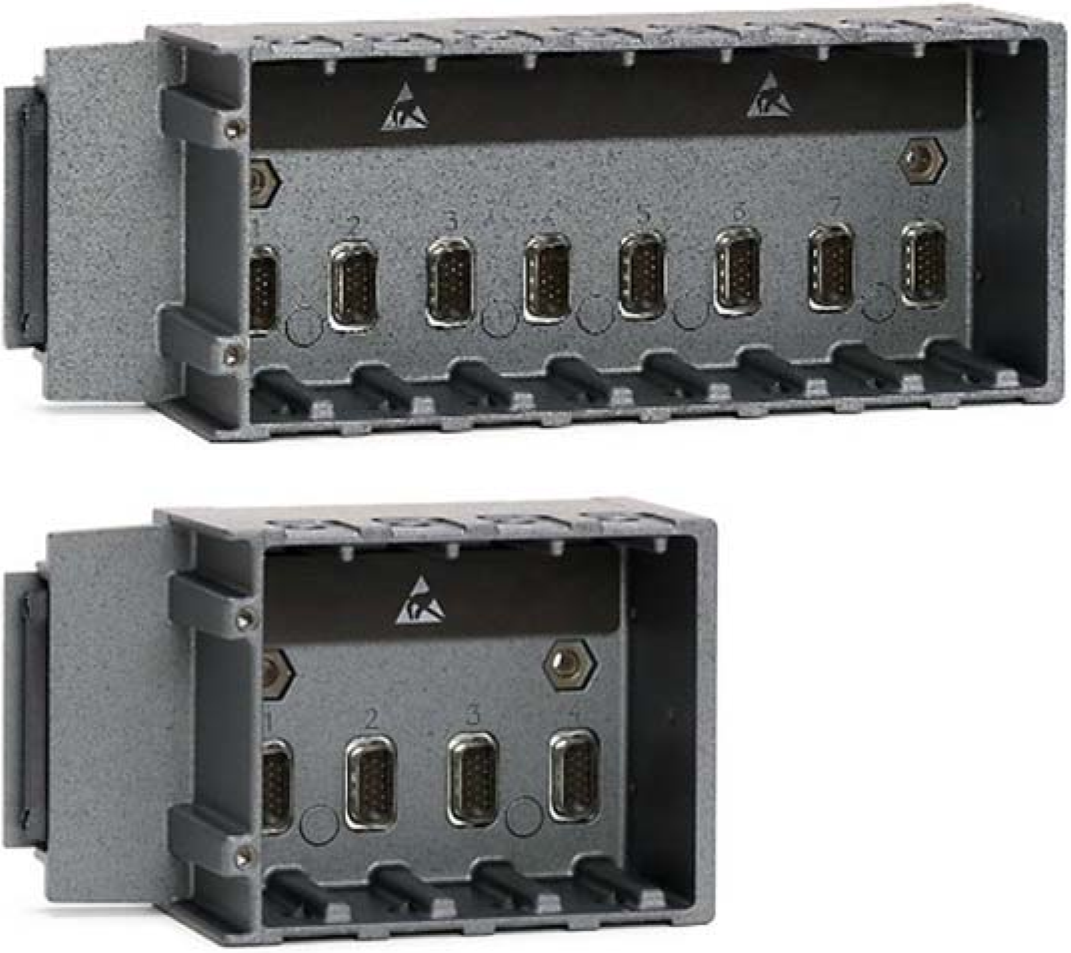


Figure 3-21 National Instruments cRIO 8-slot & 4-slot module chassis.

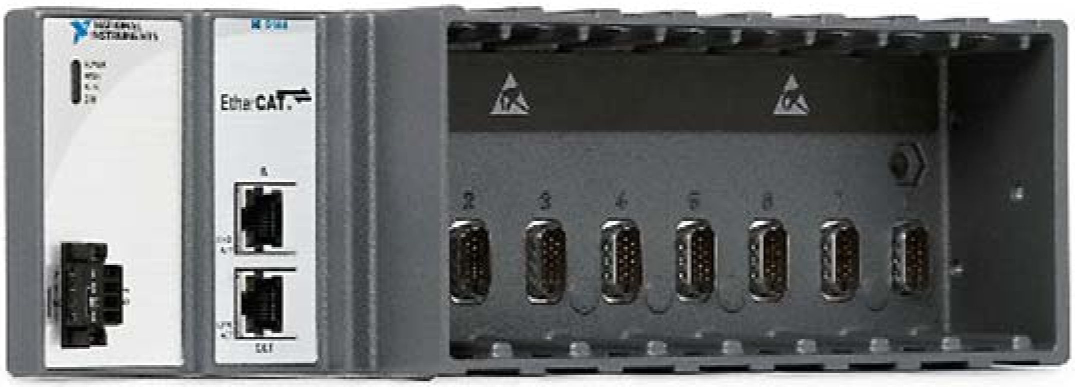


Figure 3-22 National Instruments 9144 EtherCAT slave chassis for cRIO 9022.

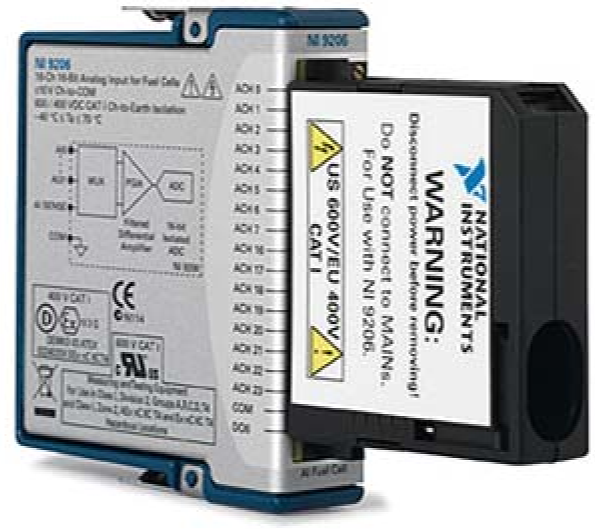


Figure 3-23 National Instruments 9206 16 channel analog voltage input module.

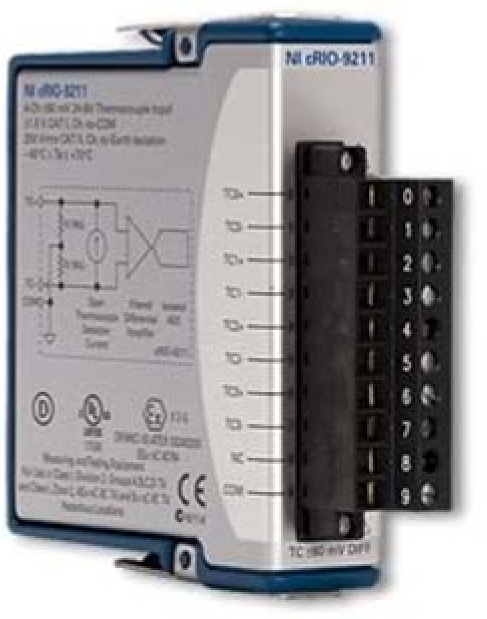


Figure 3-24 National Instruments 9144 4 channel thermocouple differential analog input module.

Figure 3-21 depicts the control box for the National Instruments hardware and physical wire connections from the Magnelab current transformer sensors and communications cables that connect to the workstation running National Instruments software (detailed in next section).

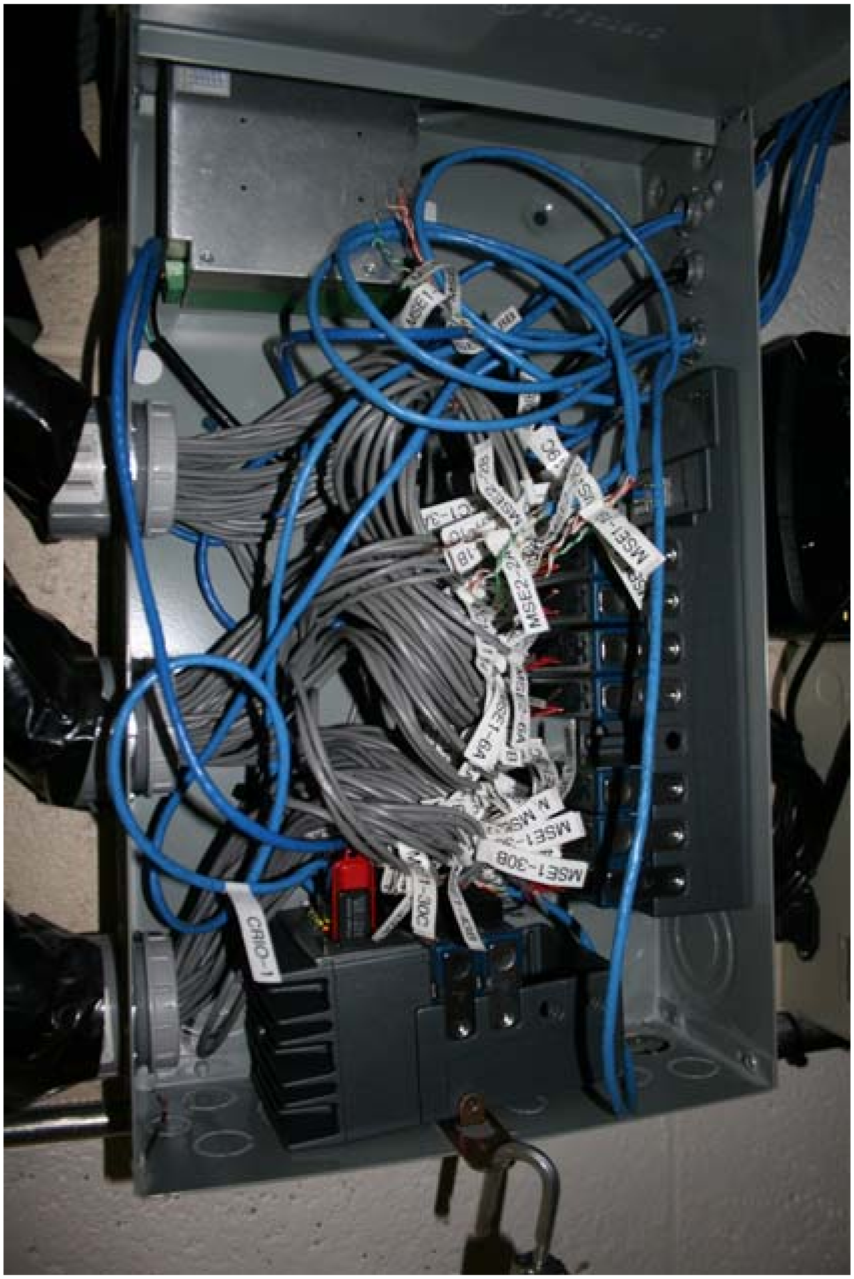


Figure 3-25 National Instruments hardware control box with cRIO and EtherCAT.

Approximately 5,000 feet of 18 gauge insulated and shielded wire was used to connect the physical leads from the Magnelab current transformer sensors to the National Instruments data acquisition hardware. Additionally, approximately 1,000 feet of Cat 6 Ethernet cable was used to connect the National Instruments data acquisition hardware to the main computer workstation running the National Instruments software (see next section). By metering every single circuit in the building, a high resolution of electrical consumption data is achieved. Figure 3-27 demonstrates the NI hardware configuration.

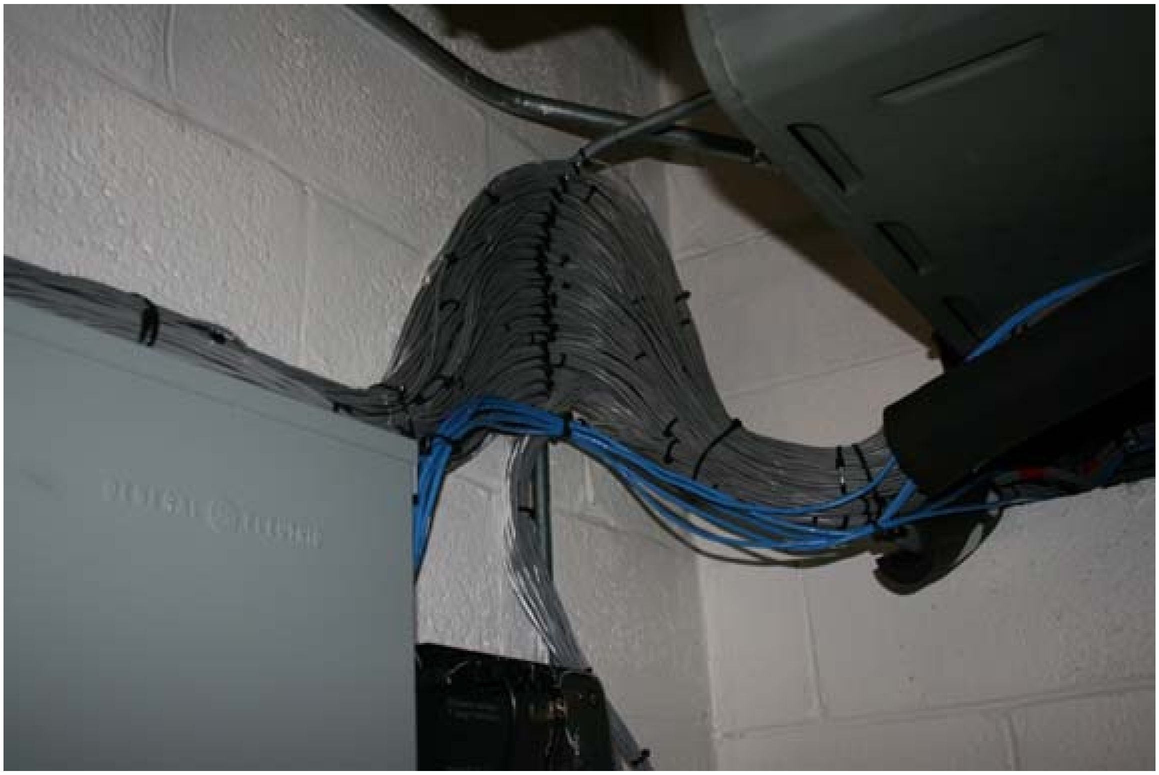


Figure 3-26 Sensors and communications cabling.

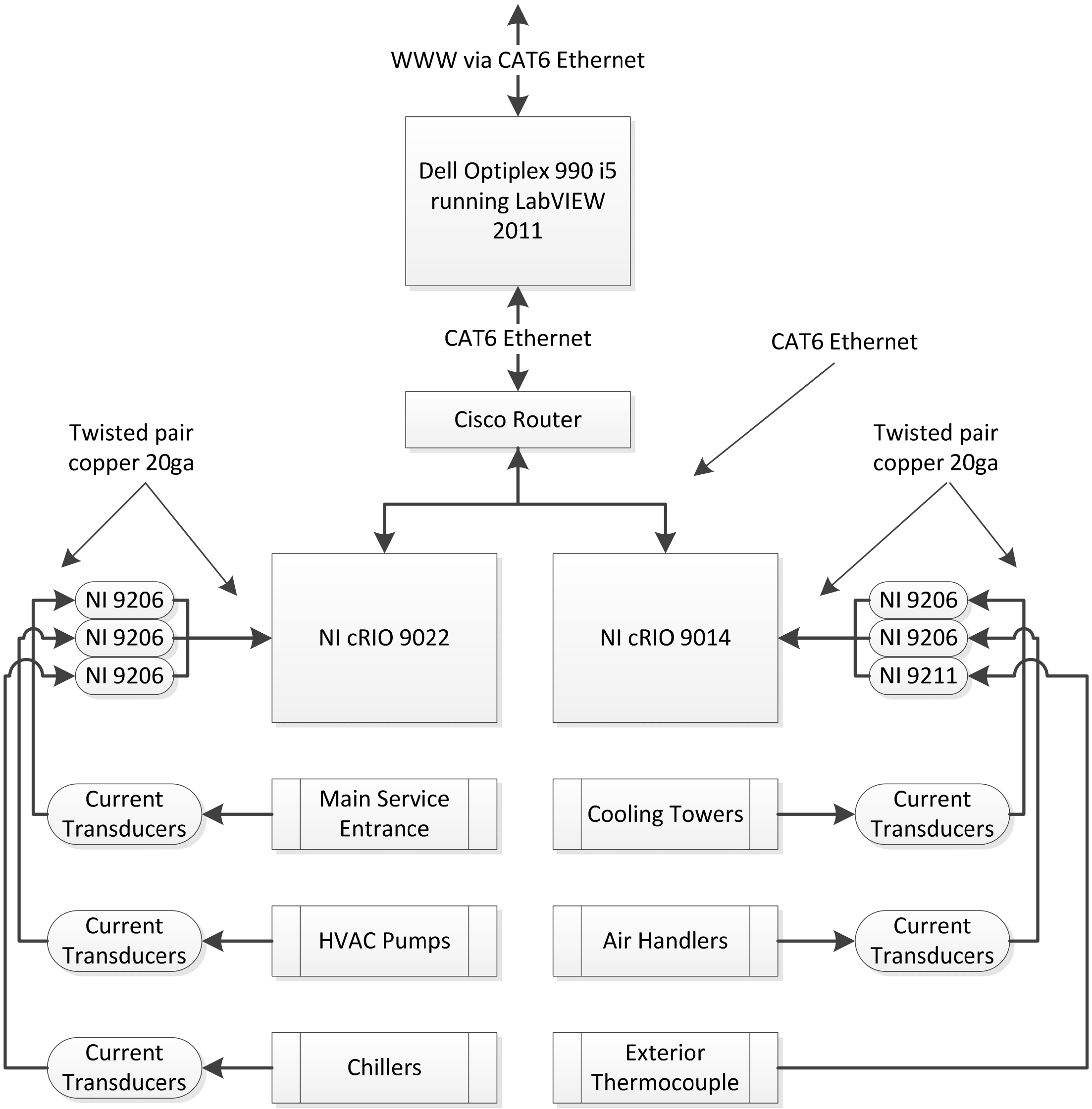


Figure 3-27 National Instruments hardware configuration.

Software Design Programming

The software employed to meter and record data emanating from the Magnelab current transformer sensors and National Instruments data acquisition hardware was National Instruments LabVIEW 2011 SP1. LabVIEW, through its comprehensive system design environment, unique graphical programming language, built-in engineering-specific libraries of software functions and hardware interfaces; and data analysis, visualization, and sharing features, facilitated the data capture for this research. For this application, LabVIEW was installed on a Dell OptiPlex 990 64-bit Intel Core i5-2400 CPU @ 3.10 GHz, 8 GB RAM, running Windows 7 64-bit operating system.

The first step with LabVIEW begins by setting up the data acquisition hardware. All data acquisition hardware devices were connected to the Dell workstation via Ethernet over a local area network. Within the National Instruments Measurement & Automation Explorer, each cRIO device is configured with the correct software, static local area network IP addresses, and their internal clocks are set to the correct day and time. Figure 3-23 shows the hardware communication connections for the local area network; 192.168.1.XXX.

The second step in LabVIEW involves software programming which allows the user to program precise control over the data acquisition hardware. This begins by creating a LabVIEW project. Once a project has been created, the user then adds the data acquisition hardware to the newly created project. For this project, the cRIO 9022, cRIO 9014, 8 slot cRIO chassis, 4 cRIO slot chassis, EtherCAT chassis, and input modules were added. Figure 3-28 shows the project with added hardware in LabVIEW.

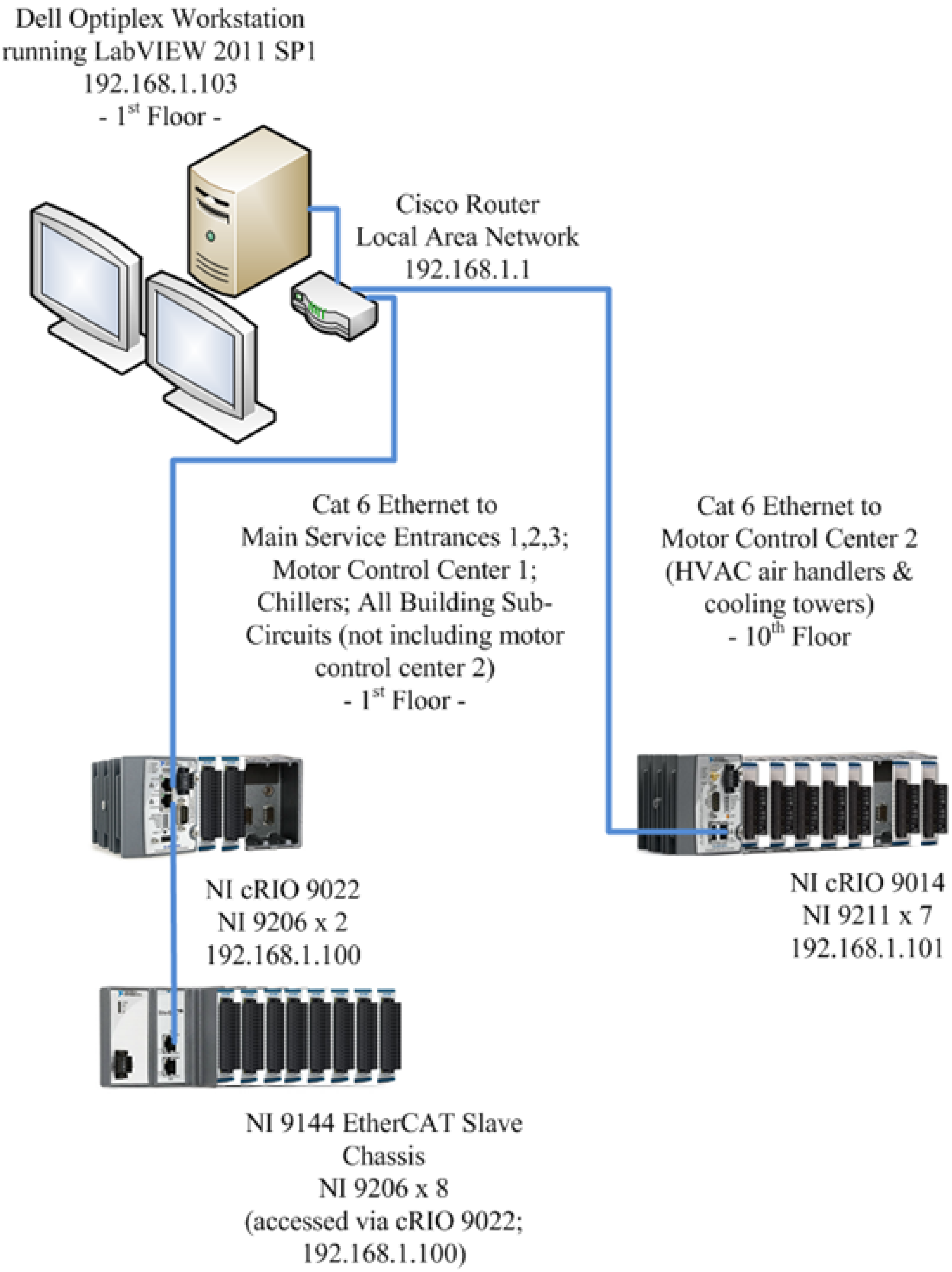
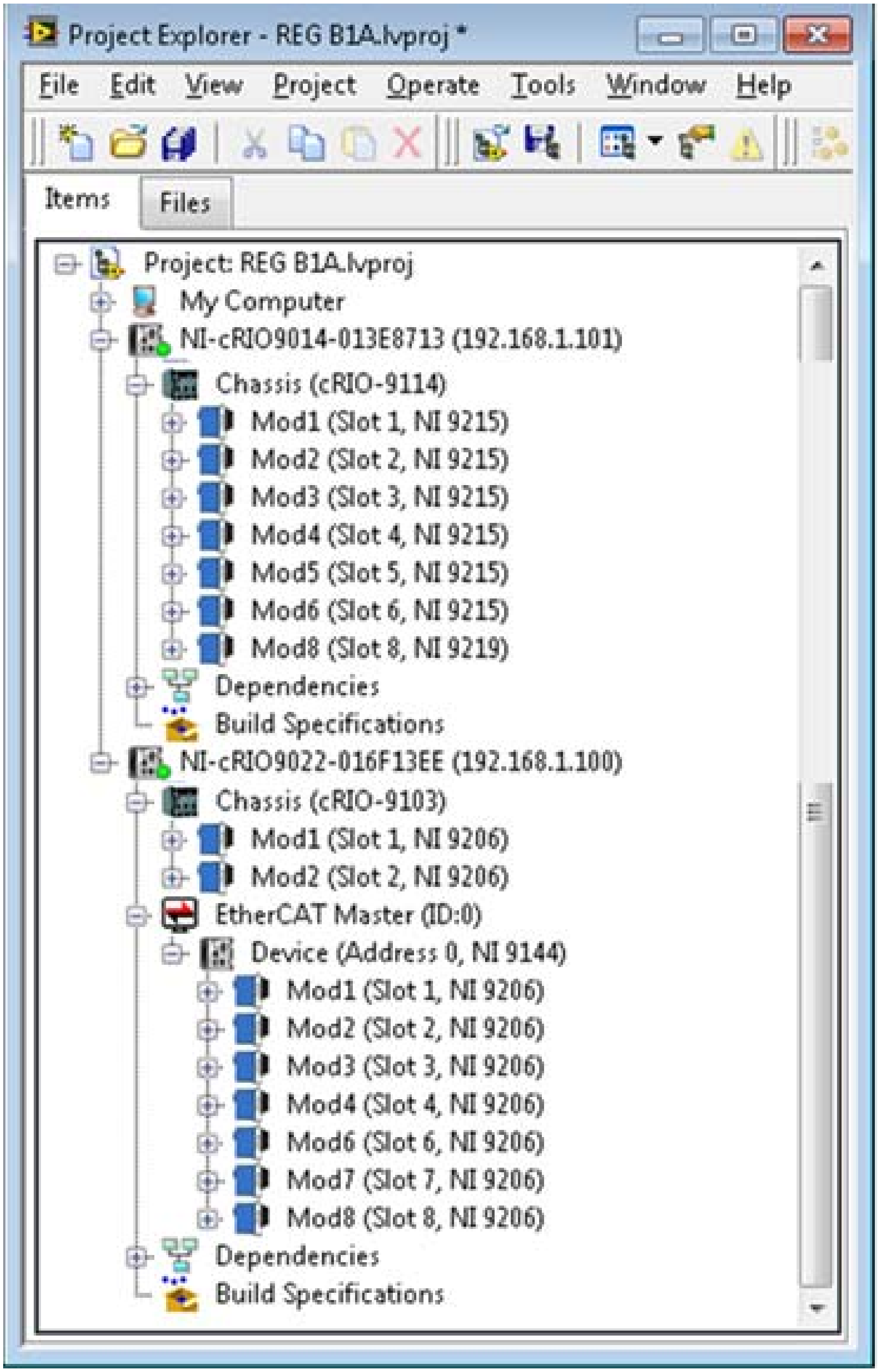


Figure 3-28 Local area network hardware communication connections.



```
Project: REG B1A.lvproj
    My Computer
    NI-cRIO9014-013E8713 (192.168.1.101)
        Chassis (cRIO-9114)
            Mod1 (Slot 1, NI 9215)
            Mod2 (Slot 2, NI 9215)
            Mod3 (Slot 3, NI 9215)
            Mod4 (Slot 4, NI 9215)
            Mod5 (Slot 5, NI 9215)
            Mod6 (Slot 6, NI 9215)
            Mod8 (Slot 8, NI 9219)
        Dependencies
        Build Specifications
    NI-cRIO9022-016F13EE (192.168.1.100)
        Chassis (cRIO-9103)
            Mod1 (Slot 1, NI 9206)
            Mod2 (Slot 2, NI 9206)
        EtherCAT Master (ID:0)
            Device (Address 0, NI 9144)
                Mod1 (Slot 1, NI 9206)
                Mod2 (Slot 2, NI 9206)
                Mod3 (Slot 3, NI 9206)
                Mod4 (Slot 4, NI 9206)
                Mod6 (Slot 6, NI 9206)
                Mod7 (Slot 7, NI 9206)
                Mod8 (Slot 8, NI 9206)
        Dependencies
        Build Specifications
```

Figure 3-29 LabVIEW project with added Hardware.

The final step in LabVIEW is the creation of virtual instruments (VIs). Virtual instruments are program files using National Instruments LabVIEW's graphical programming interface. Multiple VIs serve to precisely capture and record data. Figure 3-29 depicts the data capture flow chart methodology. All graphical LabVIEW VIs programed to capture and record data are detailed in Appendix A. All VIs' display interfaces are also depicted in Appendix A. The entire project with all VIs is shown in Figure 3-30.

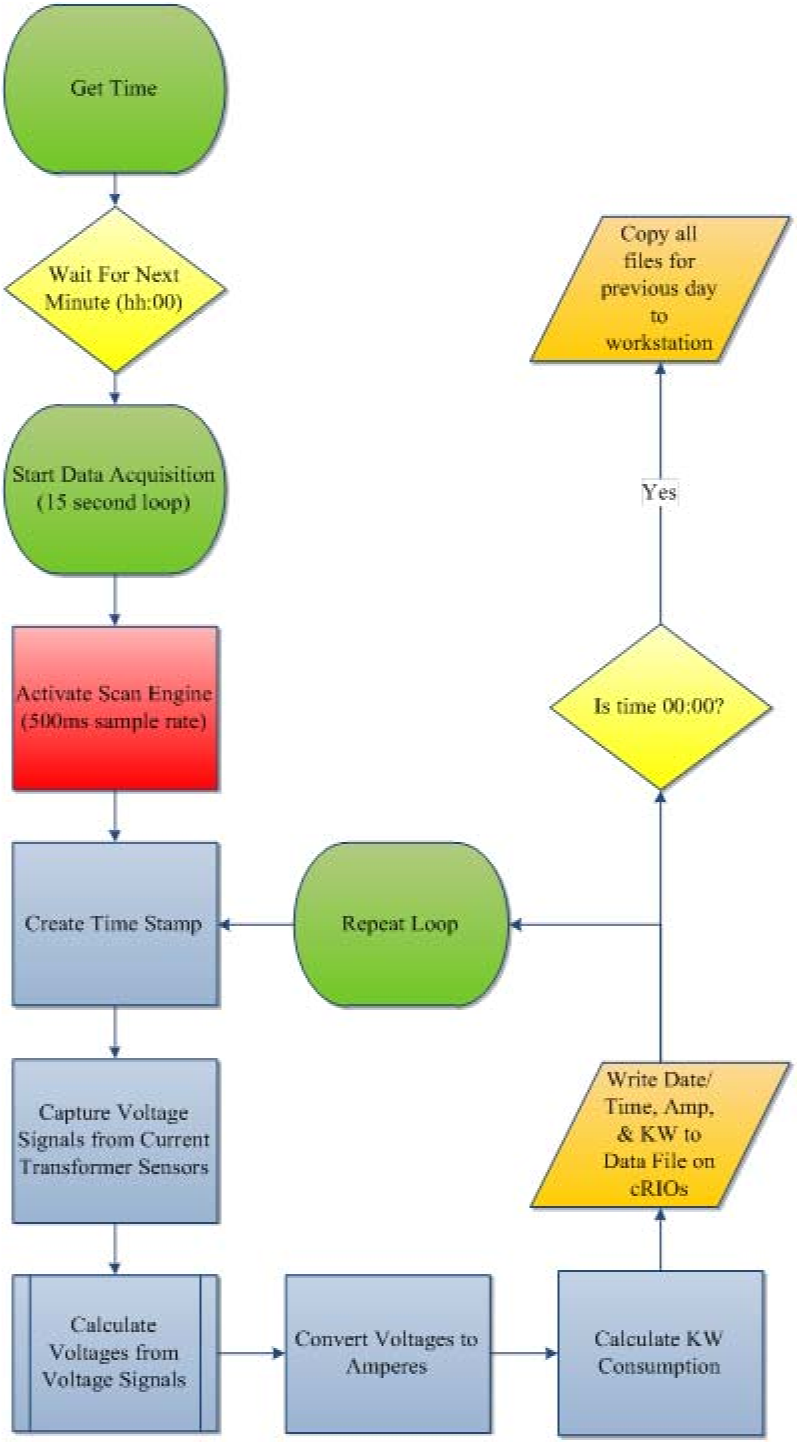
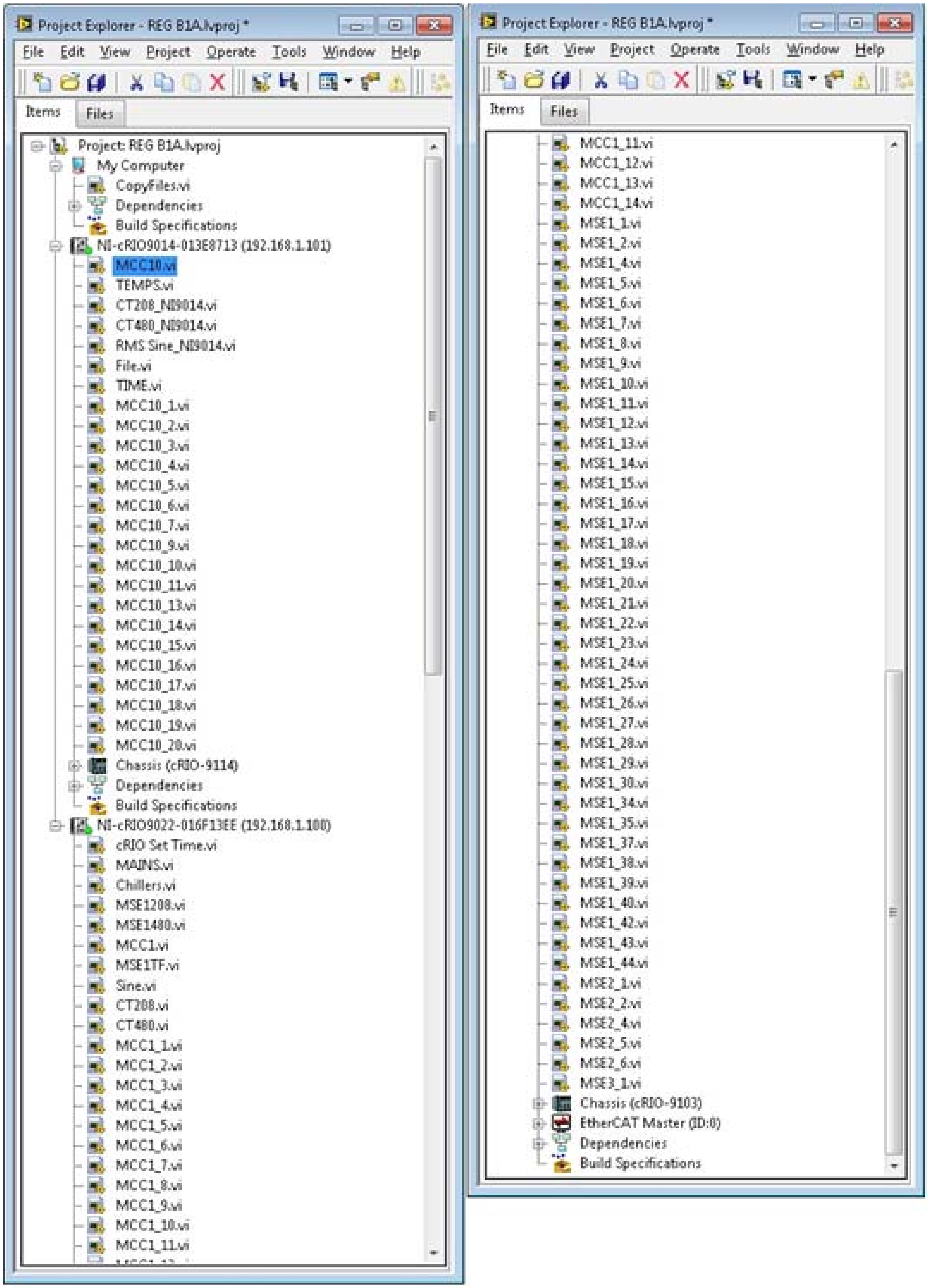

Get Time
Wait For Next Minute (hh:00)
Start Data Acquisition (15 second loop)
Activate Scan Engine (500ms sample rate)
Create Time Stamp
Capture Voltage Signals from Current Transformer Sensors
Calculate Voltages from Voltage Signals
Convert Voltages to Amperes
Calculate KW Consumption
Write Date/ Time, Amp, & KW to Data File on cRIOs
Is time 00:00?
Yes
Copy all files for previous day to workstation
Repeat Loop

Figure 3-29 Data capture flow chart.


Project Explorer - REG B1A.lvproj *
File Edit View Project Operate Tools Window Help
Items
Files

```
Project: REG B1A.lvproj
  My Computer
    CopyFiles.vi
    Dependencies
    Build Specifications
  NI-cRIO9014-013E8713 (192.168.1.101)
    MCC10.vi
    TEMPS.vi
    CT208_NI9014.vi
    CT480_NI9014.vi
    RMS Sine_NI9014.vi
    File.vi
    TIME.vi
    MCC10_1.vi
    MCC10_2.vi
    MCC10_3.vi
    MCC10_4.vi
    MCC10_5.vi
    MCC10_6.vi
    MCC10_7.vi
    MCC10_9.vi
    MCC10_10.vi
    MCC10_11.vi
    MCC10_13.vi
    MCC10_14.vi
    MCC10_15.vi
    MCC10_16.vi
    MCC10_17.vi
    MCC10_18.vi
    MCC10_19.vi
    MCC10_20.vi
    Chassis (cRIO-9114)
    Dependencies
    Build Specifications
  NI-cRIO9022-016F13EE (192.168.1.100)
    cRIO Set Time.vi
    MAINS.vi
    Chillers.vi
    MSE1208.vi
    MSE1480.vi
    MCC1.vi
    MSE1TF.vi
    Sine.vi
    CT208.vi
    CT480.vi
    MCC1_1.vi
    MCC1_2.vi
    MCC1_3.vi
    MCC1_4.vi
    MCC1_5.vi
    MCC1_6.vi
    MCC1_7.vi
    MCC1_8.vi
    MCC1_9.vi
    MCC1_10.vi
    MCC1_11.vi
```

```
    MCC1_11.vi
    MCC1_12.vi
    MCC1_13.vi
    MCC1_14.vi
    MSE1_1.vi
    MSE1_2.vi
    MSE1_4.vi
    MSE1_5.vi
    MSE1_6.vi
    MSE1_7.vi
    MSE1_8.vi
    MSE1_9.vi
    MSE1_10.vi
    MSE1_11.vi
    MSE1_12.vi
    MSE1_13.vi
    MSE1_14.vi
    MSE1_15.vi
    MSE1_16.vi
    MSE1_17.vi
    MSE1_18.vi
    MSE1_19.vi
    MSE1_20.vi
    MSE1_21.vi
    MSE1_22.vi
    MSE1_23.vi
    MSE1_24.vi
    MSE1_25.vi
    MSE1_26.vi
    MSE1_27.vi
    MSE1_28.vi
    MSE1_29.vi
    MSE1_30.vi
    MSE1_34.vi
    MSE1_35.vi
    MSE1_37.vi
    MSE1_38.vi
    MSE1_39.vi
    MSE1_40.vi
    MSE1_42.vi
    MSE1_43.vi
    MSE1_44.vi
    MSE2_1.vi
    MSE2_2.vi
    MSE2_4.vi
    MSE2_5.vi
    MSE2_6.vi
    MSE3_1.vi
    Chassis (cRIO-9103)
    EtherCAT Master (ID:0)
    Dependencies
    Build Specifications
```

Figure 3-30 Entire LabVIEW project will all Vis.

Measurement & Verification

Measurement and & Verification took place upon hardware installation of the system. A Fluke 434 Power Quality Analyzer was employed to measure and verify each individual current transformer sensor against the observed sensor reading within the LabVIEW environment. A few sensors and/or lead wires had to be adjusted and/or replaced. Additionally, throughout the data capture, a random sampling of circuits was periodically conducted to verify that the sensor readings being captured in the LabVIEW environment were accurate.

CHAPTER 4: METHODOLOGY

Introduction to Neural and Adaptive Systems

The study of neural and adaptive systems is a unique and growing interdisciplinary field that considers adaptive, distributed, and mostly nonlinear systems. They will complement effectively present engineering design principles and help build the preprocessors to interface with the real world and ensure the optimality needed in complex systems. When applied correctly, a neural or adaptive system may considerably outperform other methods. Neural and adaptive systems are used in many important engineering applications, such as signal enhancement, noise cancellation, and classification of input patterns, system identification, prediction, and control. They are used in many commercial products such as modems, image-processing and recognition systems, speech recognition, front-end signal processors and biomedical instrumentation (Principe & Euliano, 1999).

The leading characteristic of neural and adaptive systems is their adaptability, which brings a totally new system design style, Figure 4-1. Instead of being built a priori from specification, neural and adaptive systems use external data to automatically set their parameters. This means that neural systems are parametric. It also means that they are made "aware" of their output through a performance feedback loop that includes a cost function. The performance feedback is utilized directly to change the parameters through systematic procedures called learning or training rules, so that the system output improves with respect to the desired goal (Principe et al., 1999).

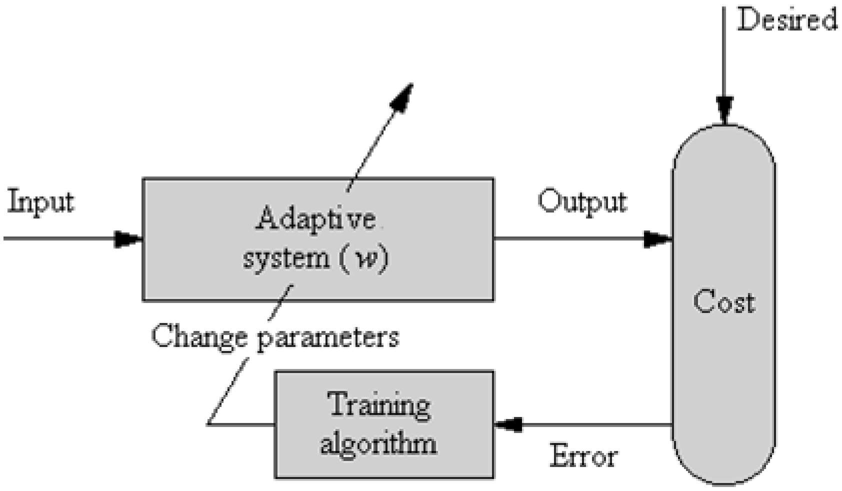


Figure 4-1 Adaptive system's design (Principe et al., 1999).

This chapter describes the application of neural network for the prediction of the peak demand and represents the results obtained from that network comparing it with the experimental data measured from the site of interest.

The system designer has to specify just a few crucial steps in the overall process: he has to decide the system topology, choose a performance criterion, and design the adaptive algorithms. In neural systems, the parameters are often modified in a selected set of data called the training set and are fixed during operation. The designer thus has to know how to specify the input and desired response data and when to stop the training phase. In adaptive systems, the system parameters are continuously adapted during operation with the current data.

The problem of data fitting is one of the oldest in experimental science. The real world tends to be very complex and unpredictable, and the exact mechanisms that generate the data are often unknown. Moreover, when we collect physical variables, the sensors are not ideal (of finite precision, noisy, with constrained bandwidth, etc.), so the measurements do not represent the real phenomena exactly. One of the quests in science is to estimate the underlying data model.

The importance of inferring a model from the data is to apply mathematical reasoning to the problem. The major advantage of a mathematical model is the ability to understand, explain, predict, and control outcomes in the natural system (Principe et al., 1999).

Figure 4-2 illustrates the data-modeling process. The most important advantage of the existence of a formal equivalent model is the ability to predict the natural system's behavior at a future time and to control its outputs by applying appropriate inputs.

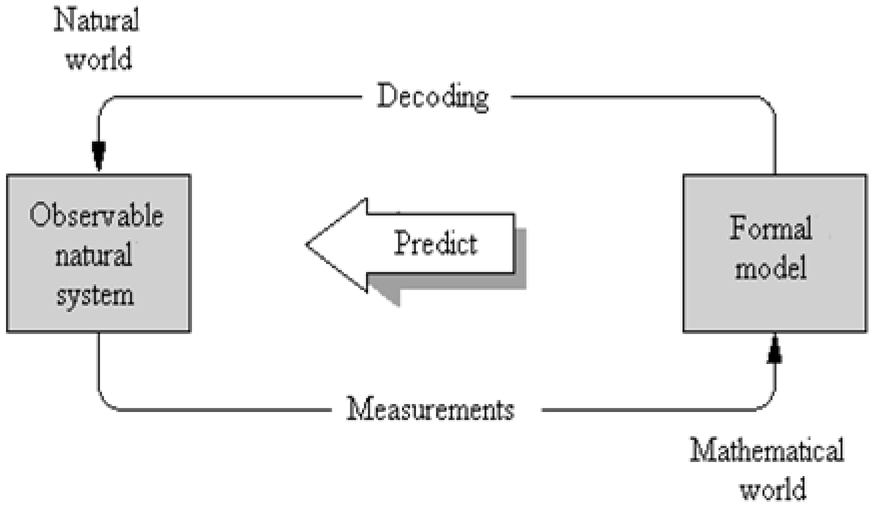


Figure 4-2 Natural Systems and formal models (Principe et al., 1999).

What is a Neural Network?

A neural network is an adaptable system that can learn relationships through repeated presentation of data, and is capable of generalizing to new, previously unseen data. Some networks are supervised, in that a human must determine what the network should learn from the data. Other networks are unsupervised, in that the way they organize information is hard-coded into their architecture .

Neural networks are used for both regression and classification. In regression, the outputs represent some desired, continuously valued transformation of the input patterns. In classification, the objective is to assign the input patterns to one of several categories or classes, usually represented by outputs restricted to lie in the range from 0 to 1, so that they represent the probability of class membership. For regression, it can be shown that neural networks can learn any desired input-output mapping if they have sufficient numbers of processing elements in the hidden layer(s). For classification, neural networks can learn the Bayesian posterior probability of correct classification .

Multilayer Perceptron

The multilayer perceptron (MLP) is one of the most widely implemented neural network topologies. The MLP is capable of approximating arbitrary functions which has been important in the study of nonlinear dynamics, and other function mapping problems.

The multilayer perceptron is trained with error correction learning, which means that the desired response for the system must be known. In pattern recognition, this is normally the case, since we have our input data labeled, i.e. we know which data belongs to which experiment.

Error correction learning works in the following way: From the system response at the nonlinear processing element, PE i at iteration n, yi(n), and the desired response di(n) for a given input pattern an instantaneous error ei (n) is defined by

$$e_i(n)= d_i(n)- y_i(n) \tag{1}$$

Using the theory of gradient descent learning, each weight in the network can be adapted by correcting the present value of the weight with a term that is proportional to the present input and error at the weight, i.e.

$$w_{ij}(n+1)= w_{ij}(n)+\eta\delta_i(n)x_j(n) \tag{2}$$

The local error δi(n) can be directly computed from ei(n) at the output PE or can be computed as a weighted sum of errors at the internal PEs. The constant η is called the step size. This procedure is called the back propagation algorithm. Back propagation computes the sensitivity of a cost functional with respect to each weight in the network, and updates each weight proportional to the sensitivity.

The advantage of the procedure is that it can be efficiently implemented with local information and requires just a few multiplications per weight. Momentum learning is an improvement to the straight gradient descent in the sense that a memory term (the past increment to the weight) is used to speed up and stabilize convergence.

In momentum learning the equation to update the weights becomes:

$$w_{ij}(n+1)= w_{ij}(n)+ \eta\delta_i(n)x_j(n)+ \alpha(w_{ij}(n)- w_{ij}(n-1)) \tag{3}$$

where α is the momentum. Normally α should be set between 0.1 and 0.9.
Training can be implemented by presenting all the patterns in the input file (an epoch), accumulate the weight updates, and then update the weights with the average weight update. This is called batch learning.

We start by loading an initial value for each weight (normally a small random value) to start back propagation and proceed until one of these three stopping criterion is met which are: to cap the number of iterations, to threshold the output mean square error or to use cross validation. Cross validation is the most powerful of the three since it stops the training at the point of best generalization (i.e. the performance in the test set) is obtained. In our study, cross validation is used; thus, a small part of the training data is used to see how the trained network is doing. Cross validation computes the error in a test set at the same time that the network is being trained with the training set. When the performance starts to degrade in the validation set, training should be stopped.

A learning curve is developed during the training procedure to show how the mean square error evolves with the training iteration. When the learning curve is flat, the step size is to be increased to speed up learning. When the learning curve moves up and down the step size should be decreased. An important point that should be considered in order to decrease the training time and provide better performance is the normalization of the training data.

The performance of the MLP in the test set to be limited by the relation N>W/e , where N is the number of training epochs, W the number of weights and e the performance error. Training continues until the mean square error is less than e /2 (Youssef, Al-Makky, & Abd-Elwahab, 2003).

Learning Curve

This plot shows the mean squared error (MSE) of the network after each epoch of data. The epoch number is shown on the X-axis and the MSE is shown on the Y-axis. The MSE of the training set is shown in red and the MSE of the cross-validation set (if included) is shown in blue. A network that is trained well should have a constantly decreasing slope of the training MSE (typically an exponential decay). As long as the training set learning curve is decreasing, the network is still training. If the training set learning curve is increasing or bouncing up and down, the network is probably not training well (try decreasing the learning rates) (Principe et al., 1999).

The specified data sets are Training, Cross Validation and Testing. Training data is the portion of the data used to actually train the network. This is normally the largest portion of data.

Cross Validation data is used to intermittently validate the training. Periodically testing the network (no weight changes during cross validation) during training can help avoid overspecializing on the training data. Cross validation data is data set aside to test the network during training (the network parameters are not directly updated/trained with this data). It is used to stop the network training when the network starts to specialize too much on the training data.

Testing data is used to further validate the results of a trained network. Although the network was not trained with the cross validation data, the training may have been stopped using it. Therefore, the cross validation data is not truly “out-of-sample”. The testing data is data set aside to test the network after it has been trained and is truly “out-of-sample”.

The developed neural network in Figure 4-3 is called a Multi-Layer Perceptron (MLP). It is the most common supervised neural network. It consists of multiple layers of processing elements (PEs) connected in a feed forward fashion. The PEs in the developed network are the orange circular icons and are called axons. The connections between the PEs are the icons with horizontal and diagonal lines between the axons and are called synapses. Back propagation of errors is used to train the MLP. The smaller icons on top of the axons and synapses are called back propagation components and pass the error backwards from the end of the network to the beginning. The green axons on top of the back-propagation components are called gradient search components and adjust the weights contained in the synapses and axons.

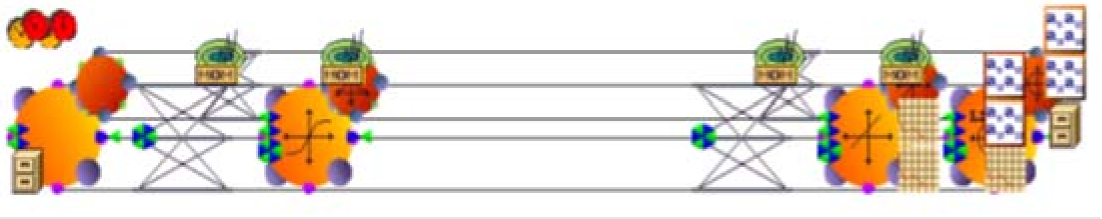


Figure 4-3 Layout of peak demand neural network.

The 2nd axon from the right is the output axon and generates the actual network outputs. The 1st axon on the left is called the input axon and it does nothing but accept the input from the file component.

Each neural component encapsulates the functionality of a particular piece of a neural network. A working neural network simulation requires the interaction of many different components.

As mentioned before, adaptive learning using gradient descent focuses on using the error between the system output and the desired system output to train the system. The learning algorithm adapts the weights of the system based on the error until the system produces the desired output.

The goal of the developed network is for the system output to be the same as the desired output, so we want to minimize the mean squared error. The method used to do this is called error back propagation. This is done through three main steps; first, the input data is propagated forward through the network to compute the system output. Next, the error is computed and propagated backward through the network. And lastly, the error is used to modify the weights.

Figure 4-4 represents the structure of the developed neural network, in which the inputs for the developed neural networks are the different energy channels, such as Chiller, AC units, lighting, motors, etc. The output of the network is the peak demand.

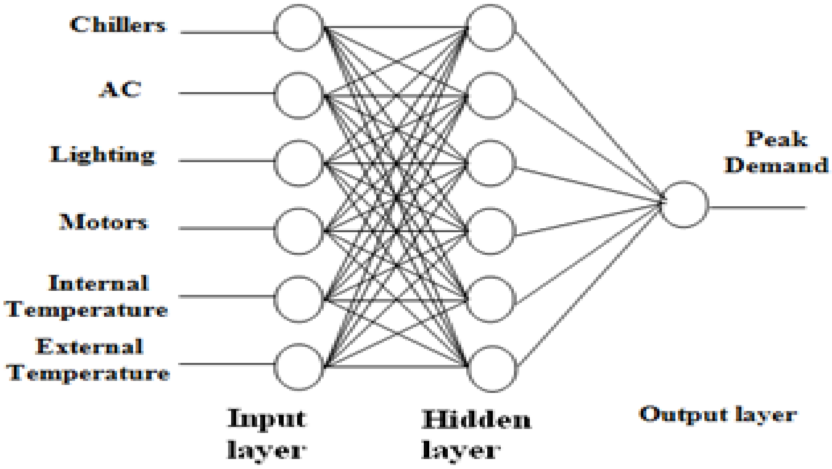


Figure 4-4 Artificial neural network structure.

The Main Parts of the Developed Neural Network

Input Axon

This component is a place holder axon that simply accepts the inputs to the system and passes it on to the rest of the neural network. The file component on the bottom-left is used to read the data from the file system. These components make up the first hidden layer of the neural network, Figure 4-5.

First Hidden Layer

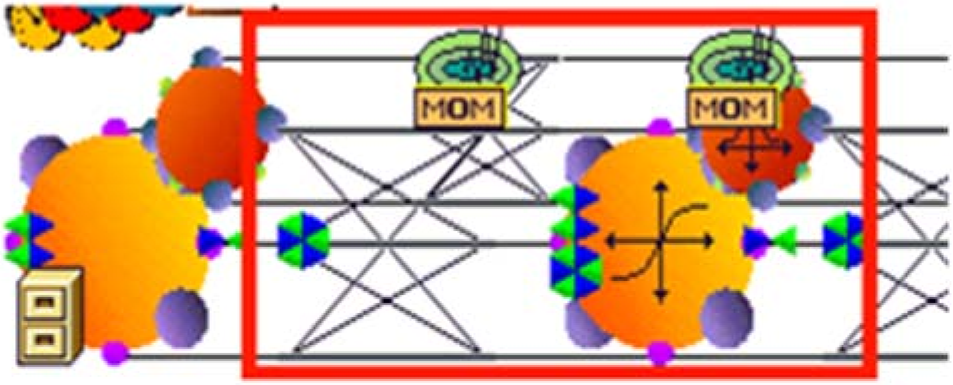


Figure 4-5 First Hidden Layer.

The synapse, is the first component in the box and makes the connection between each input and each processing element (PE) in the hidden layer. The synapse, Figure 4-6, contains the connections and the trainable weights for each connection.

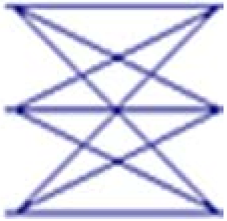


Figure 4-6 The Synapse.

The second component is the tanh axon. This component has the processing elements for the hidden layer, each of which sums the weighted connections from the inputs, Figure 4-7.

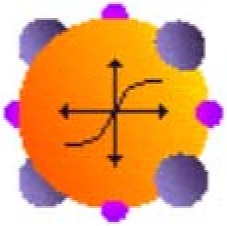


Figure 4-7 The tanh axon.

The green components on top of the axon and synapse are the momentum gradient search components, Figure 4- 8.

Figure 4-8 The momentum.

Second Hidden Layer

These components in Figure 4-9 make up the second hidden layer of the neural network.

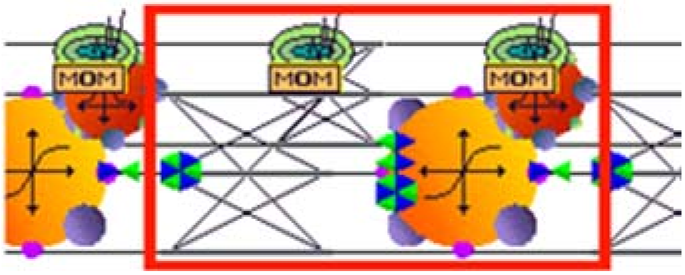


Figure 4-9 The second hidden layer

Output Layer (Classification)

The output layer is also made up of the same two components, the synapse and the tanh axon, Figure 4-10.

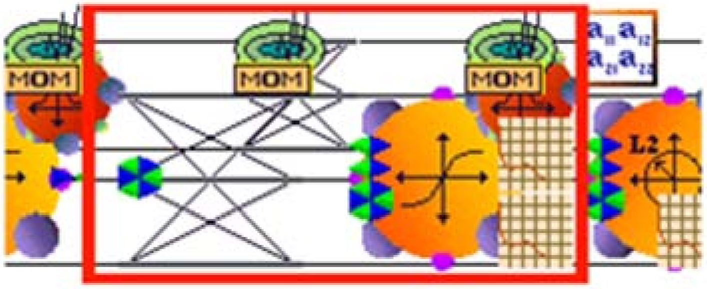


Figure 4-10 The output layer.

There is one output PE for each desired output specified in the desired file. The desired file is shown in the file component on the right side of the criterion.

Criterion

The criterion accepts the output(s) of the network and the desired output(s) and compares them, Figure 4-11. It computes the error and passes this error to the back propagation components which adjust the weights of the network for training. Because the criterion has the information about the performance of the network, it contains many access points for network performance.

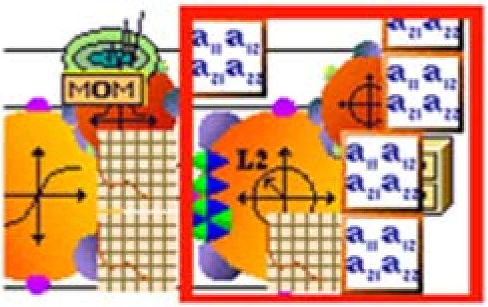


Figure 4-11 The criterion.

Controllers

These controllers contain the global control parameters for the network. The yellow dials are the static controllers. They contain parameters such as the number of epochs per run, the number of exemplars per epoch, the data sets to use, etc.

The red dials are the back prop controllers. It contains learning parameters like the learning mode (batch, online, etc.), Figure 4-12.

Figure 4-12 The controllers.

Bias Axon

The Bias Axon simply provides a bias term, which may be adapted, Figure 4-13.

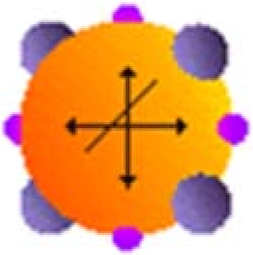


Figure 4-13 The bias axon.

Gaussian Axon

The Gaussian Axon implements a radial basis function layer. The Gaussian Axon only responds significantly to a local area of the input space (where the peak of the Gaussian is located), Figure 4-14. It is therefore considered to be a local function approximator.

The center of the Gaussian is controlled using the bias weight inherited from the Bias Axon, and its width using the β parameter inherited from the Linear Axon.

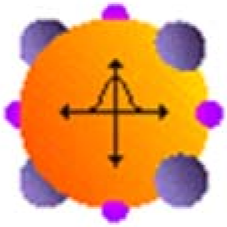


Figure 4-14 The Gaussian axon.

The Tanh axon.

The Tanh Axon applies a bias and tanh function to each neuron in the layer. This will squash the range of each neuron in the layer to between -1 and 1. Such nonlinear elements provide a network with the ability to make soft decisions, Figure 4-15.

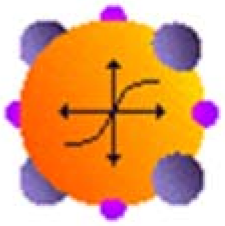


Figure 4-15 The tanh axon.

Artificial Neural Network Modeling, Training, Learning, & Initial Prediction

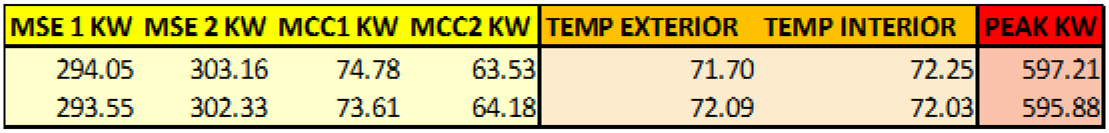
The software used for analysis is Neural Solutions from NeuralDimension, Inc. Established in 1995, NeuroSolutions is a leading edge neural network development software that combines a modular, icon-based network design interface with an implementation of advanced learning procedures, such as Levenberg-Marquardt and back propagation through time.

For the initial proof of concept analysis, one work week of data were captured. Every circuit detailed in chapter three was individually captured. Data sampling occurred every 15 seconds for a time period of one work week (five like business days Monday through Friday). For this proof of concept, a total of 28,800 data points for each circuit was recorded.

Two tests were conducted to prove the concept. First, a summation of all circuits' data for each data instance was computed to generate a cumulative output for the main service entrance #1, main service entrance #2, motor control center #1, and motor control center #2. From this data peak Kilowatt consumption was computed. This data together with internal and external temperature data was used for the first neural network model.

Table 4-1 depicts an example of two fifteen second instances of recorded data.

Table 4-1 Example of recorded data for proof of concept.



MSE 1 KW	MSE 2 KW	MCC1 KW	MCC2 KW	TEMP EXTERIOR	TEMP INTERIOR	PEAK KW
294.05	303.16	74.78	63.53	71.70	72.25	597.21
293.55	302.33	73.61	64.18	72.09	72.03	595.88

During the training phase of the neural network, the mean squared error (MSE) was cross validated against the run of test data or Epoch, Figure 4-16. The data used for testing was the last 1,000 iterations of the initial one work week data capture. The predicted results were then compared against actual as seen in Figure 4-17. Performance output is detailed in Table 4-3. The average calculated error for this first neural network model resulted at 0.25%. The low error rate is attributed to the discrete nature of this particular analysis. Since peak demand is the summation of all circuits, by analyzing the main service entrances and motor control centers which are essentially a summation of all circuits, the neural network prediction error is low. This model essentially serves as the test validation for the artificial neural network methodology.

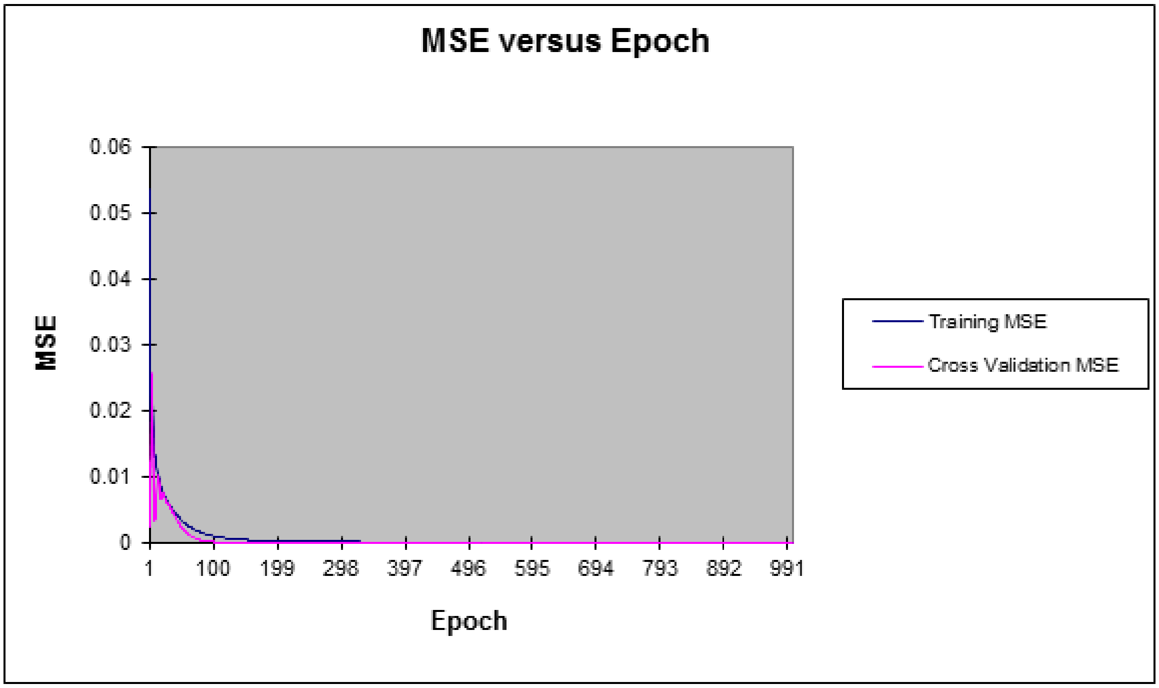


Figure 4-16 Model 1 training mean squared error vs. cross validation epoch

Table 4-2 Model 1 training mean squared error vs. cross validation epoch

Best Networks	***Training***	***Cross Validation***
Epoch #	1000	441
Minimum MSE	0.000102985	2.16631E-05
Final MSE	0.000102985	2.20178E-05

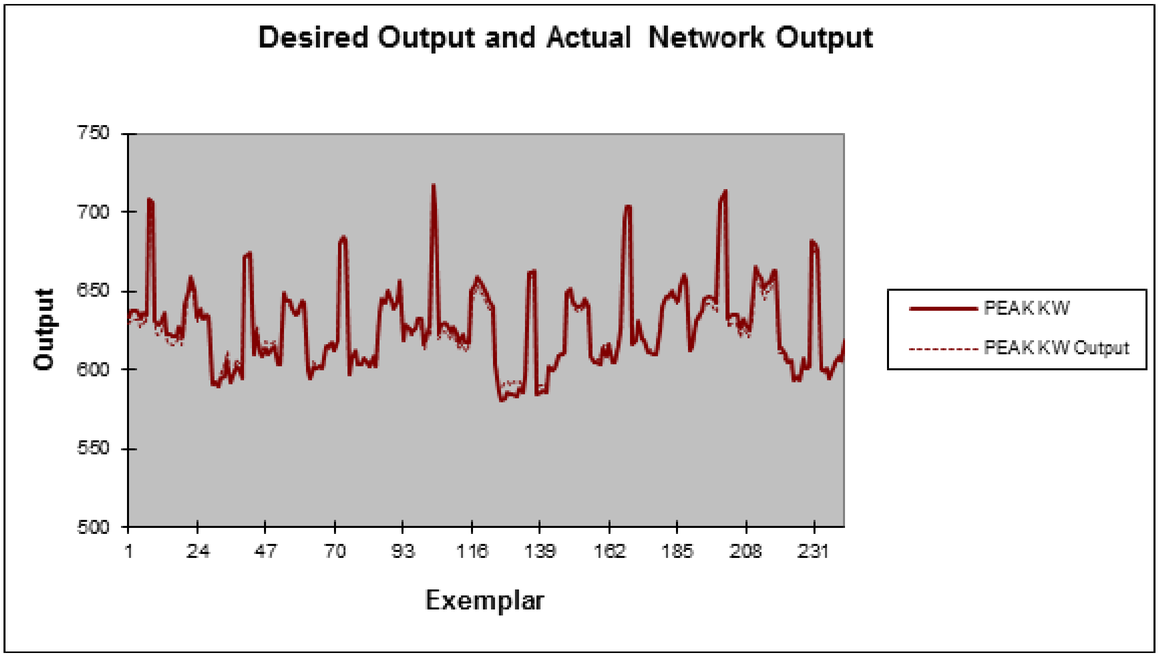


Figure 4-17 Model 1predicted output vs. actual recorded

Table 4-3 Model 1 performance output: mean square error, nominal mean square error, mean average error, minimum absolute error, maximum absolute error, and r

Performance	*PEAK KW*
MSE	14.29485496
NMSE	0.018011282
MAE	3.155394438
Min Abs Error	0.002136219
Max Abs Error	8.591848246
r	0.99401406

The second neural network model, the true practical approach, included a subset of all data captured including data captured for the three chillers, all circuits of motor control center # 1, and all circuits of motor control center #2. This subset represents large variable loads within the building environment while ignoring a large portion of static loads within the building environment. Internal and external temperatures and peak Kilowatt consumption data (summation of all building circuits) were also used.

Figure 4-18 depicts the training mean squared error vs. cross validation epoch for this second model. The average calculated error for this second neural network model resulted at 3.2%

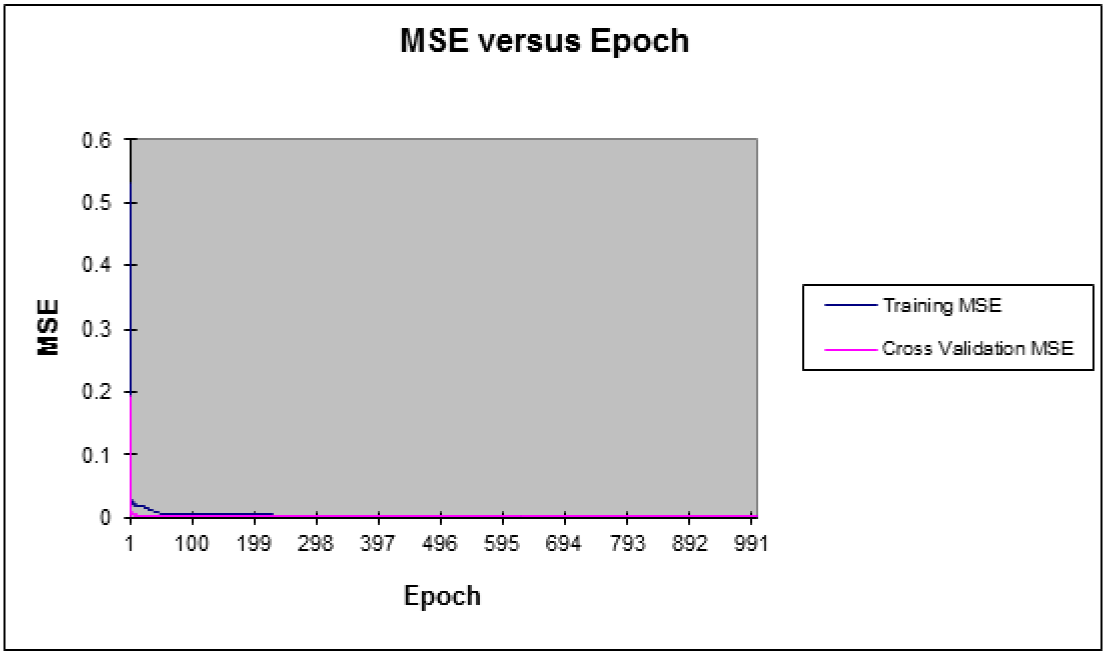
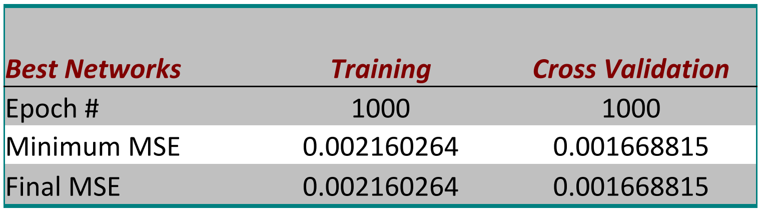


Figure 4-18 Model 2 training mean squared error vs. cross validation epoch

Table 4-4 Model 2 training mean squared error vs. cross validation epoch



Best Networks	Training	Cross Validation
Epoch #	1000	1000
Minimum MSE	0.002160264	0.001668815
Final MSE	0.002160264	0.001668815

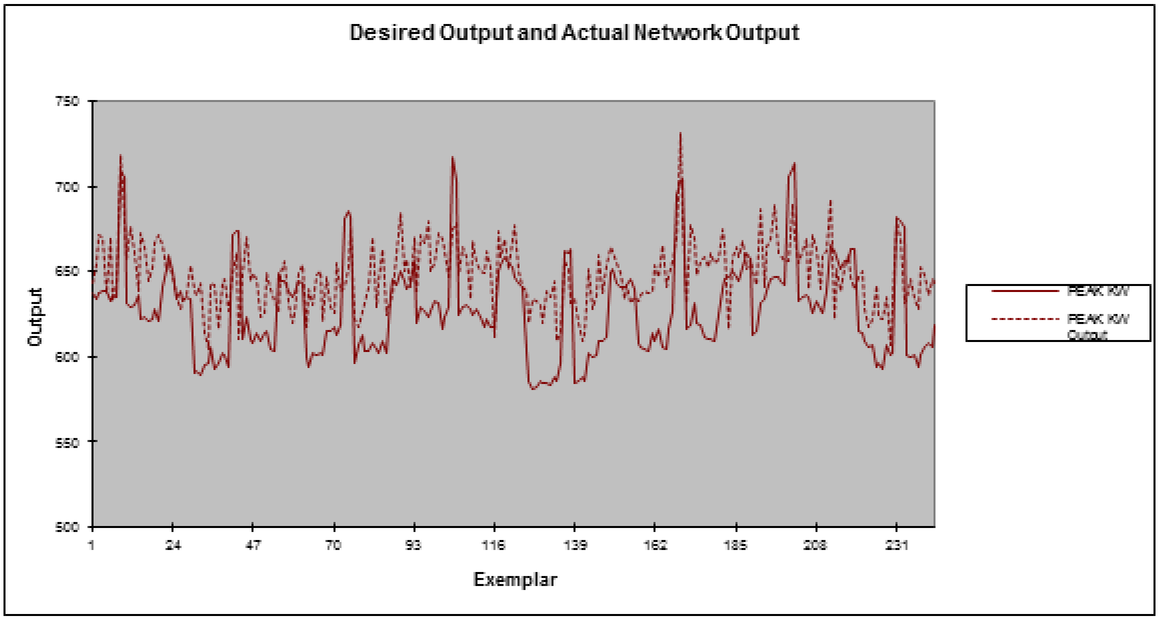
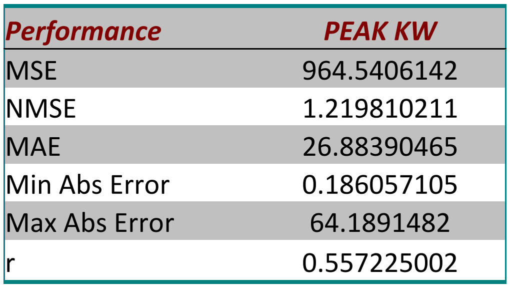


Figure 4-19 Model 2 predicted output vs. actual recorded

Table 4-5 Model 2 performance output: mean square error, nominal mean square error, mean average error, minimum absolute error, maximum absolute error, and r



Performance	*PEAK KW*
MSE	964.5406142
NMSE	1.219810211
MAE	26.88390465
Min Abs Error	0.186057105
Max Abs Error	64.1891482
r	0.557225002

Full Experiment

Lessons learned from the initial proof of concept experiment were incorporated into the full experiment. Following full data capture, statistical analysis was conducted on all model inputs and only those inputs demonstrating statistical significance ($p < 0.05$) during cross-validation were incorporated into the final model. Due to the local climate of the building location, humidity data from the U.S. National Climate Data Center were also added as an input to the model in an effort to increase accuracy. Lastly, two weeks of data instead of one were used for the testing period.

Data Collection

Using the same methodology described in the proof of concept experiment, the full experiment captured approximately four months of continuous uninterrupted data at the REGJB. In the initial proof of concept phase, each main service entrance electrical consumption was calculated as a sum of its respective sub circuits. For the full experiment, however, additional hardware was installed allowing for the direct data capture of main service entrance electrical consumption. All new hardware additions to the system were calibrated and validated to ensure proper function. The main service entrance data was used for output data fed to the ANN during the training and cross-validation stage.

Advanced Artificial Neural Network Analysis

For the main research analysis, several subsets of data were tested. Variable vs. fixed electrical loads were analyzed to determine which inputs were statistically significant enough to incorporate into the ANN forecasting model. Input variables chosen as having a direct impact on building electrical peak load are detailed as follows:

Heating, Ventilation and Air-Conditioning KW: The large government building under study has two basic building loads. First, the base load consists of all lighting and receptacle power throughout the building. Given the building's purpose and usage schedule, the base load was observed as being mostly static. The second load, or variable load, is comprised entirely of the building's HVAC system. The HVAC load with its inherent variability, coupled with additional factor inputs listed below, resulted in accurate forecasting using the ANN strategy.

Day Type: Intuitively, an electrical load forecast is contingent on the time and the day for which the prediction is being made. In the proposed model, type of day served as an input. Day type was established as a number input of 1-7 where 1 represents Monday and 7 represents Sunday. By doing so, input data were thus classified according to day type. In the building being studied, all Mondays were mostly similar, as were Tuesdays, Wednesdays, and so on. Day types also help classify weekends separate from weekdays, and holidays which fall on a weekday are classified as a weekend day type. Also, any day types that were entirely abnormal (i.e. power blackout, major HVAC failure, etc.) were filtered out.

Time of Day: As with type of day, time of day also served as a unique classifier. Since electrical building loading varies throughout the day, time of day is an effective input.

Exterior Temperature: There are several possible inputs related to weather which could be used as predictive indicators in the ANN architecture. Exterior weather conditions including but not limited to solar intensity, cloudiness, temperature, humidity, precipitation, wind speed, and barometric pressure are possibilities. Of these, however, exterior temperature is of most interest since it affects power consumption throughout the HVAC system [18]. The proposed ANN structure therefore only considers exterior temperature.

Humidity: Due to the specific local climate of the test building, humidity was also included as an input. Humidity data was acquired from the U.S. National Climate Data Center.

The output variable is detailed as follows:

Total Building Electrical KW Demand: Comprised of total electrical load, the building's KW demand is the ultimate forecast goal.

Figure 4-20 represents the final structure of the developed neural network, in which the inputs for the developed neural networks are: HVAC KW, day type, time of day, exterior temperature, and humidity. The output of the network is the total building electrical demand KW. The second hidden layer was added during initial testing of the ANN in order to reduce the output error during training.

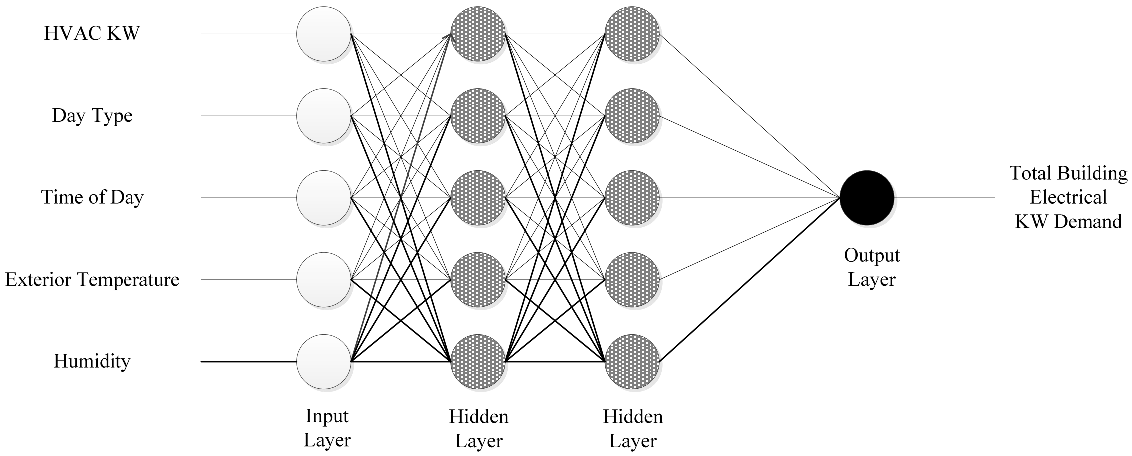


Figure 4-20 Final ANN Design.

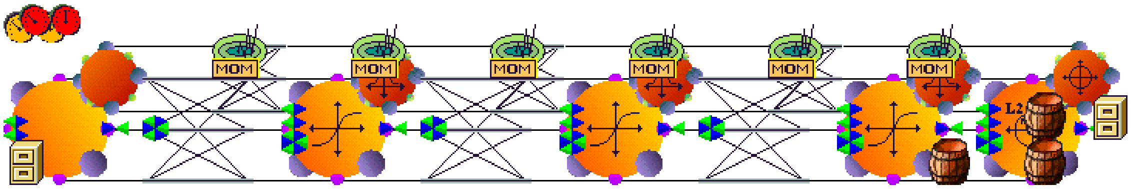


Figure 4-21 NeuroSolutions Multi-Layer Perceptron Supervised Neural Network.

Final ANN model changes:

- Aggregate HVAC KW (Chillers, Air-Handlers, Cooling Towers, HVAC motors & pumps) consumption used as one input.
- Interior Temperature eliminated (statistical significance)
- Lighting eliminated (statistical significance)
- Non-HVAC motors eliminated (statistical significance)
- Humidity added as input
- Second Hidden Layer added (reduced forecasting error)
- Two week testing period

Up to this point, the developed neural network was designed to forecast total building electrical KW demand at time ***t*** given time ***t***'s day type, time period, HVAC KW, and exterior temperature. In order to avoid a peak electrical demand occurrence through preemptive building management action, however, sixty-minute forecasting had to be built into the ANN. This was accomplished by shifting the recorded total electrical KW demand data in the 90-day ANN data set by one hour. More specifically, for each HVAC KW, day type, time of day, and exterior temperature input given to the ANN, the corresponding electrical demand output fed to the ANN training and cross-validation procedures was the total electrical KW demand of the building one hour into the future. Or put another way, for every KW demand output fed to the ANN training process, its corresponding HVAC KW, day type, time of day, exterior temperature, and humidity inputs are from one hour prior. By doing so, sixty-minute forward prediction is built into the developed ANN model.

Alternate Prediction Methodology vs. ANNs

In order to measure the performance of the developed ANN, several other statistical methods were employed. Specifically, linear regression, multi-adaptive regression splines (MARSplines), and simple moving average (SMA) were used to analyze the same data in an effort to predict total building electrical KW Demand.

Controlled Peak Demand Through Simulated Intelligent Electrical Loading

In an effort to smooth out the experienced peak demand during each billing cycle, simulated intelligent electrical loading using an electrical loading policy was adopted.

During periods of experienced peak demand, those electrical loads that could be mitigated or reduced were targeted.

CHAPTER 5: RESULTS AND DISCUSSION

The performance of the ANN forecast method for predicting total building KW demand in a large government building is presented in this chapter. Additionally comparative analysis between the developed ANN model and other forecasting methods including linear regression, MARSplines, and SMA is also detailed in this chapter. Approximately ninety days of uninterrupted data, at a sample rate of 15 seconds, were captured at the large government building. Due to nature of the building's daily operations schedule, peak demand events were only observed occurring during the work week; specifically Monday through Friday. Since there was never a chance of a peak event occurring on a weekend day, it was only necessary to test the ANN for peak demand on working weekdays. The data was tested for acceptable randomness and the residual plots for KW are shown in Figure 5-1.

Using the testing data set, a sixty-minute forward forecast using the ANN model was tested for each workday day type in a given work week (Monday – Friday) for two weeks. In an effort to validate the neural network performance during unique periods of experienced demand of each workday day type, testing was conducted every hour (240 data points, 15 sec sampling) beginning Monday 12:00am of week one through Friday 11:59pm of week two. During the training phase of the final neural network, the mean squared error (MSE) was cross validated against the run of test data or Epoch as can be seen in Figure 5-2. The gradient slope and convergence towards zero between the training MSE and cross validation demonstrates effective training and cross validation of the ANN.

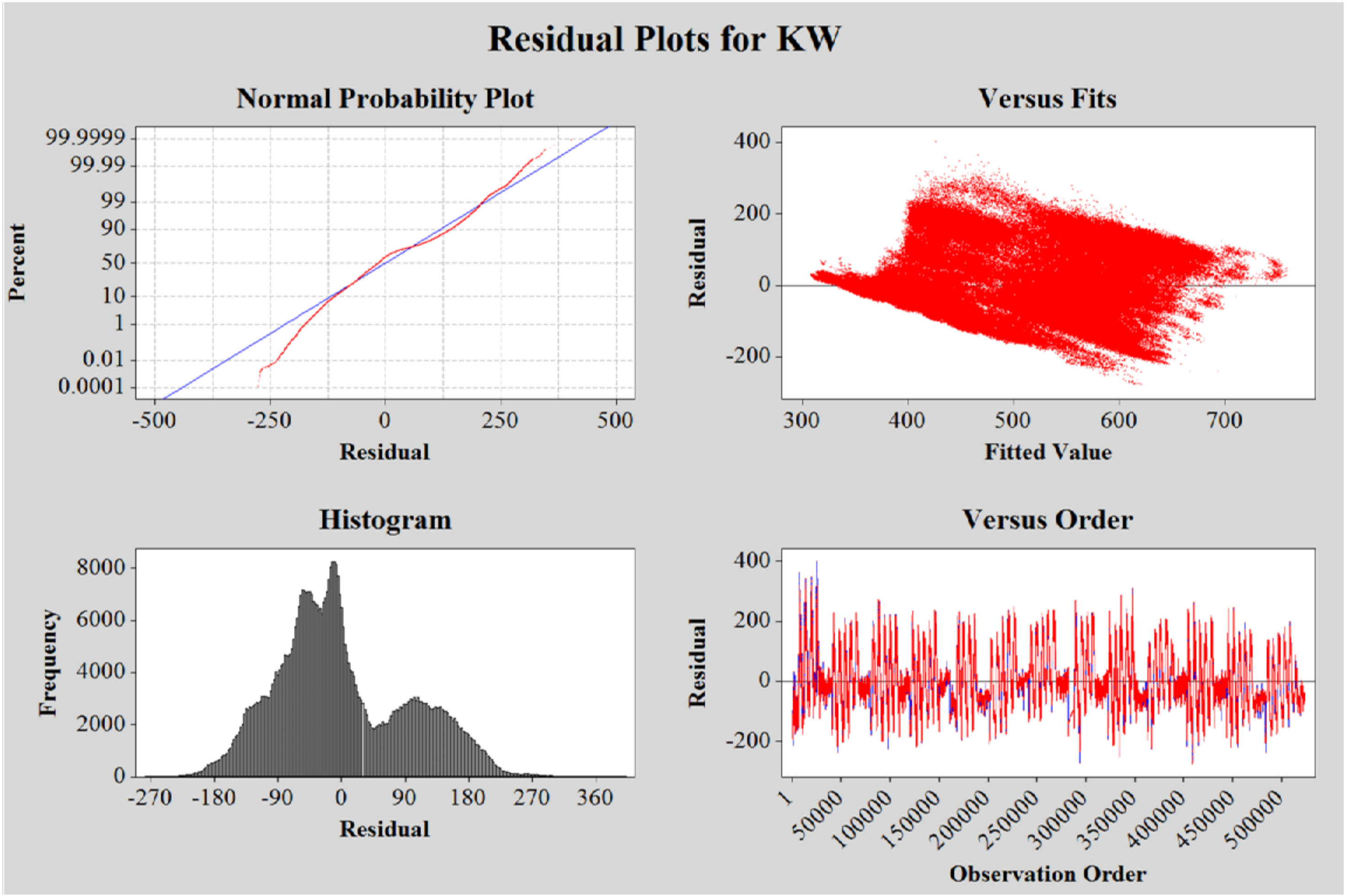


Figure 5-1 Residual Plots for KW.

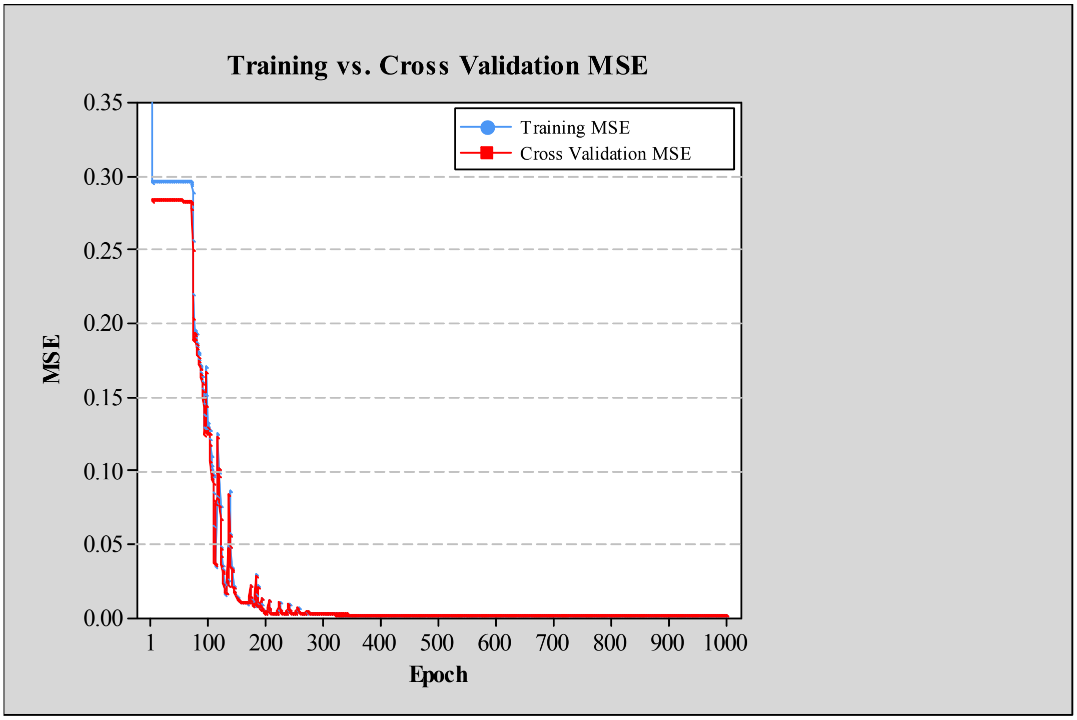


Figure 5-2 Final model training mean squared error vs. cross validation epoch.

Two work weeks of data were tested using the developed ANN. Figures 5-3 through 5-7 show the desired output versus actual network output for each workday type. For each tested workday, the desired output and actual network output were averaged between the two tested weekdays. Table 5-1 depicts the average sixty-minute ANN forecasting errors realized for both work weeks.

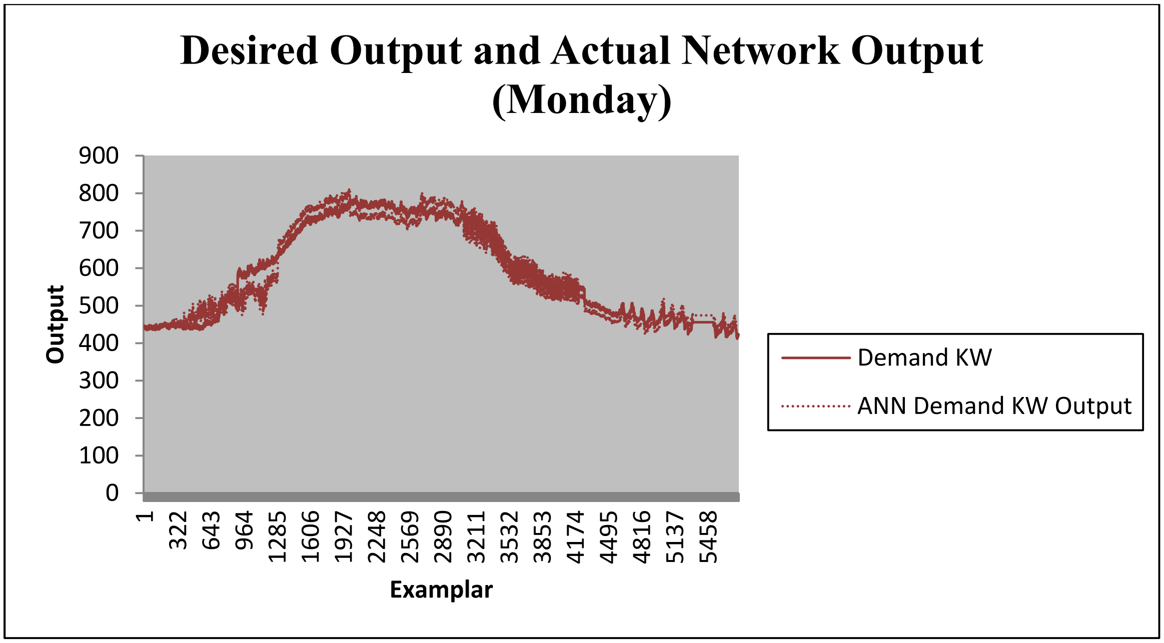


Figure 5-3 Desired output and actual network output (Monday).

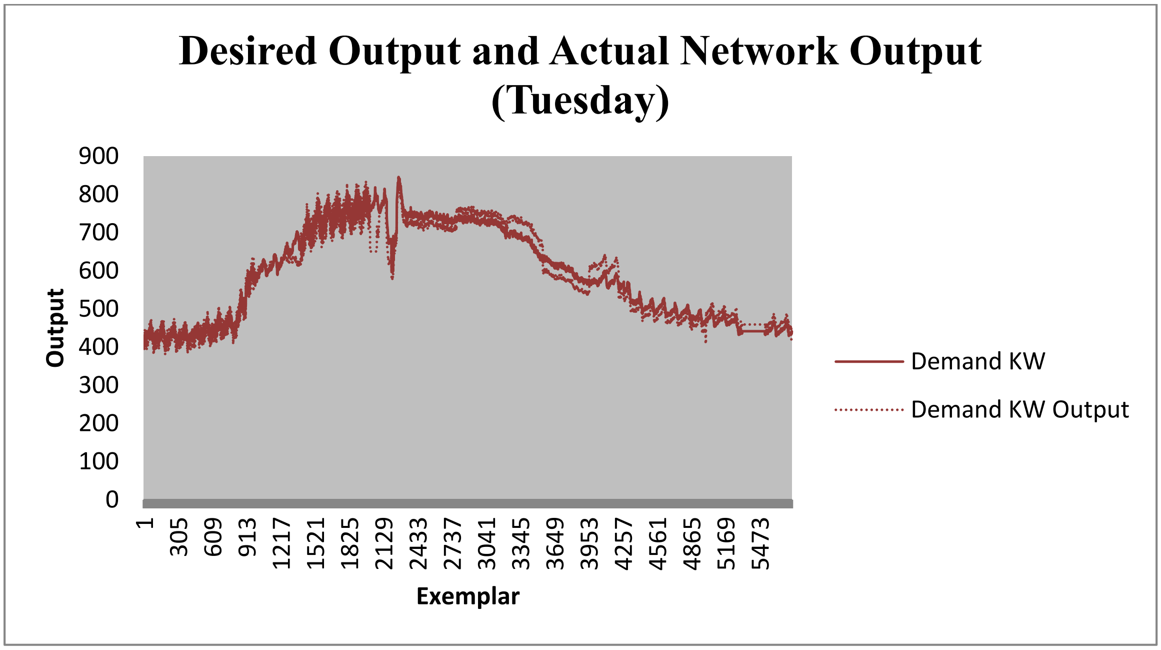


Figure 5-4 Desired output and actual network output (Tuesday).

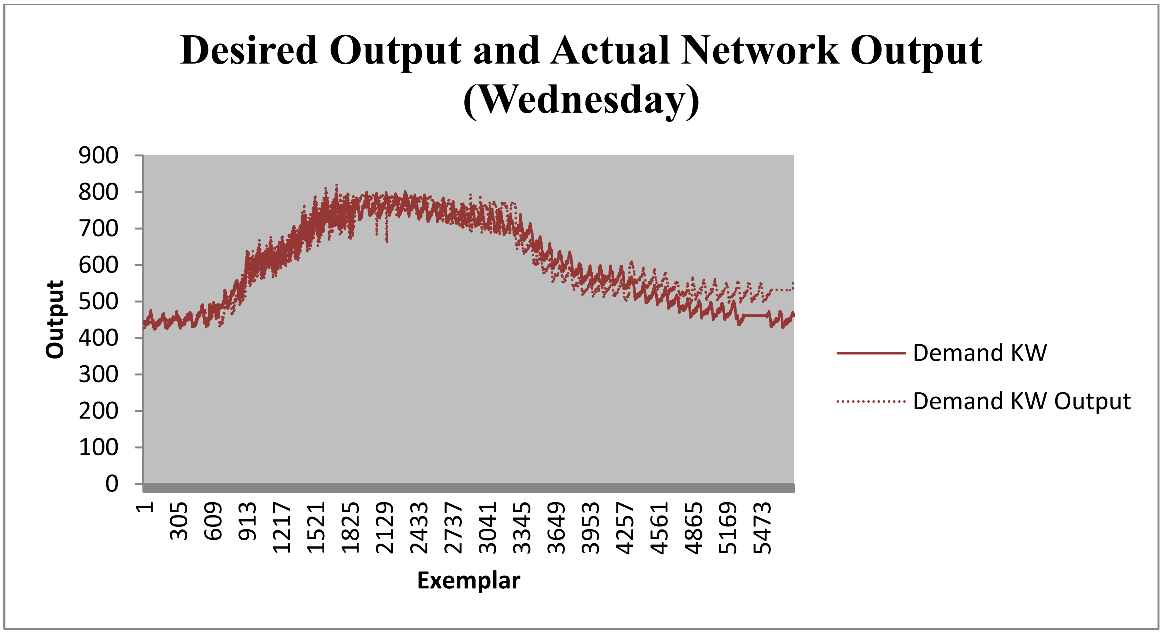


Figure 5-5 Desired output and actual network output (Wednesday).

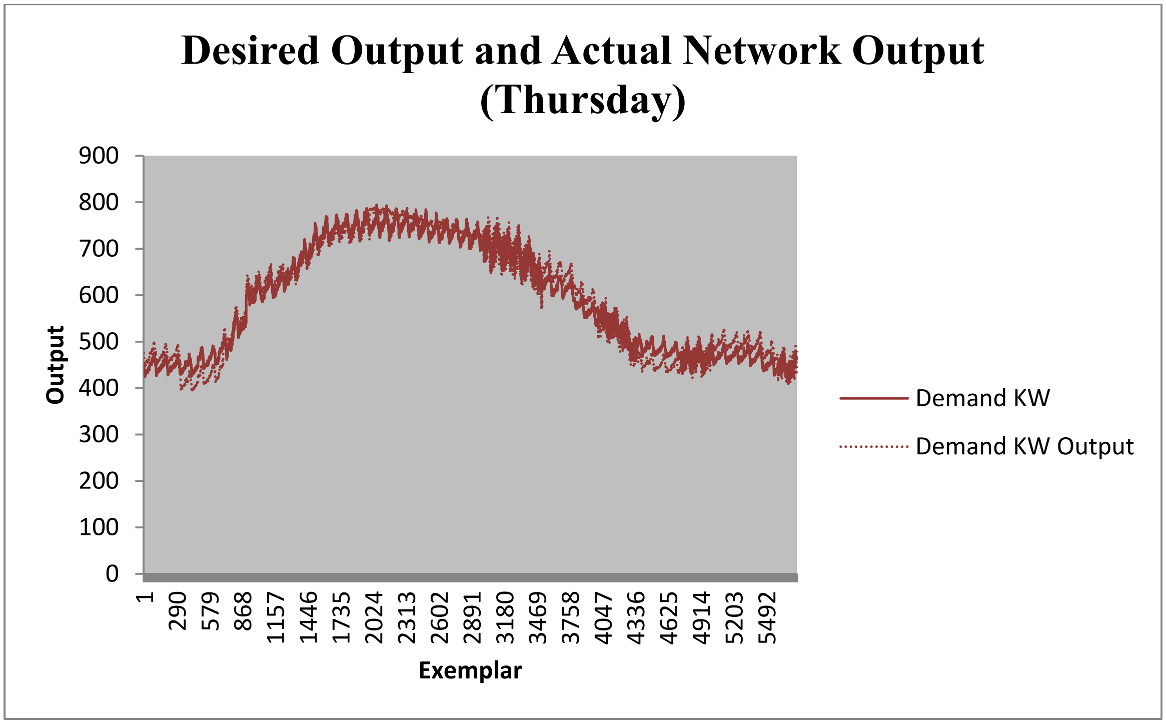


Figure 5-6 Desired output and actual network output (Thursday).

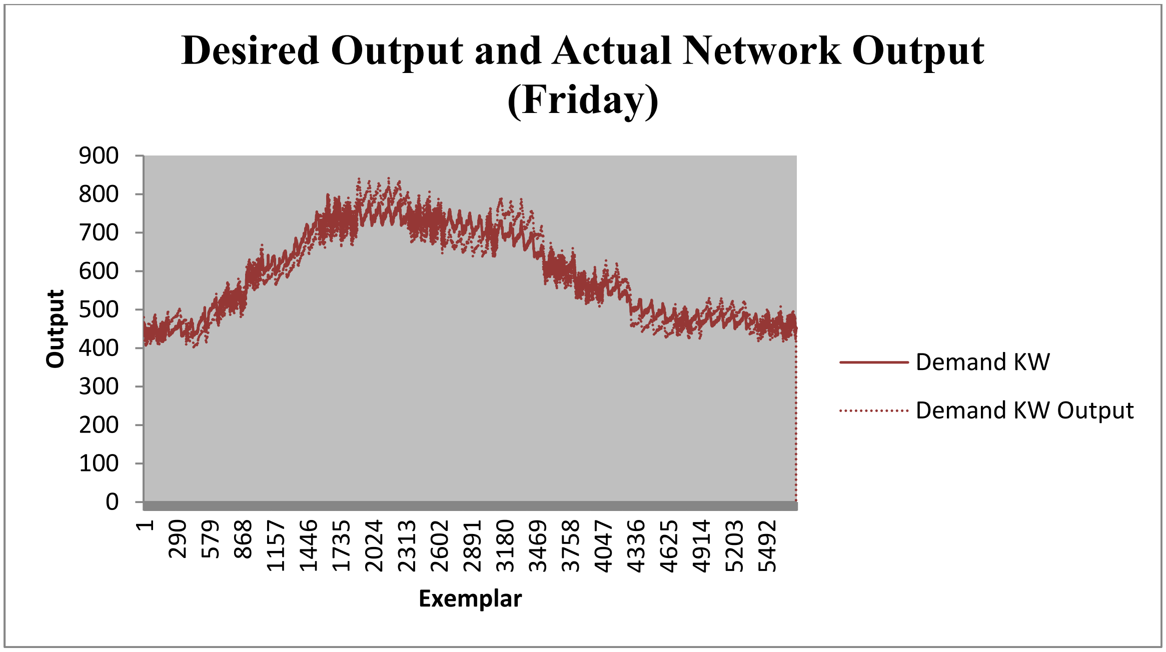


Figure 5-7 Desired output and actual network output (Friday).

Also using the same test data, the ANN model's performance was compared against three other benchmark models: SMA, linear regression, and MARSplines. Table 2 also depicts the entire work week's sixty-minute forecasting errors for the SMA, linear regression, and MARSplines benchmark models for both weeks. The average of both testing work week's sixty-minute forecasting errors for all models are also plotted in Figure 6. Table 3 depicts twenty-four hour period MAPEs amd AMEs for the ANN and benchmarking models during both testing weeks. For the developed ANN model, the MAPE and AME for all tested day types was 3.9% and 18.2% respectively. The benchmark models and their respective forecast errors are detailed next.

Simple Moving Average (SMA)

A SMA benchmark model was created for performance comparison to the developed ANN. The SMA was limited to sixty minutes of past data to predict sixty minutes into the future. The SMA model was tested using the same two week testing period and the average and maximum forecast error for all tested day types using the SMA model was 7.7% and 26.2% respectively. Table 5-1 depicts the average sixty-minute SMA forecasting errors realized for both work weeks.

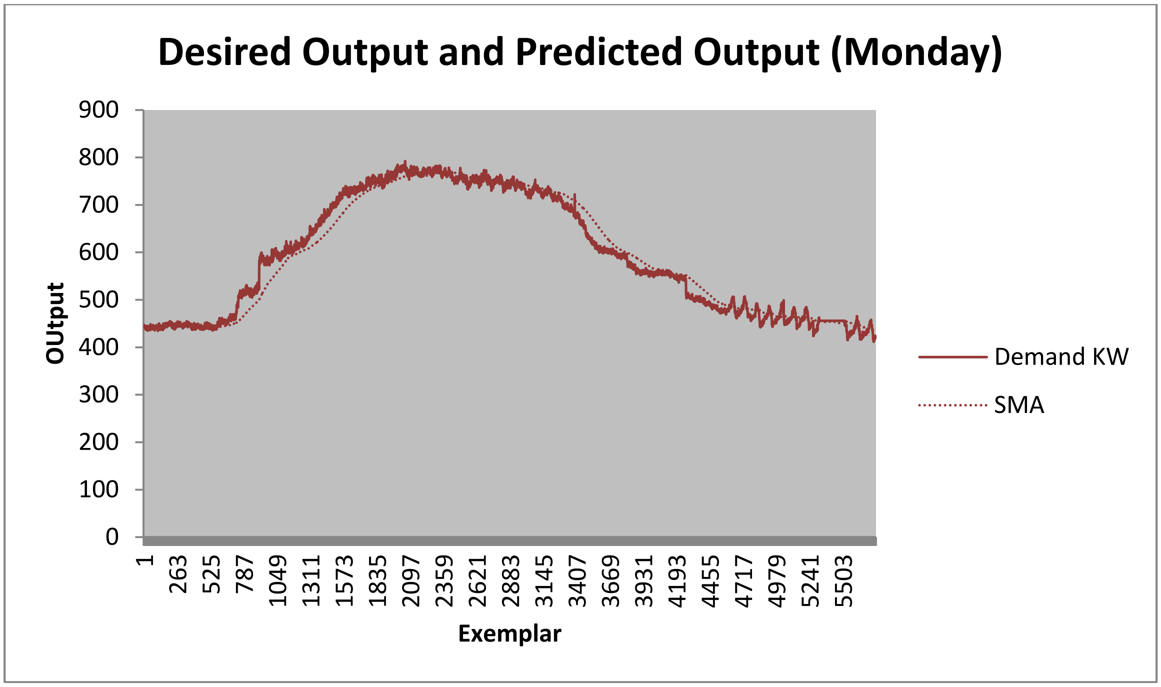


Figure 5-8 Desired output and predicted SMA output (Monday).

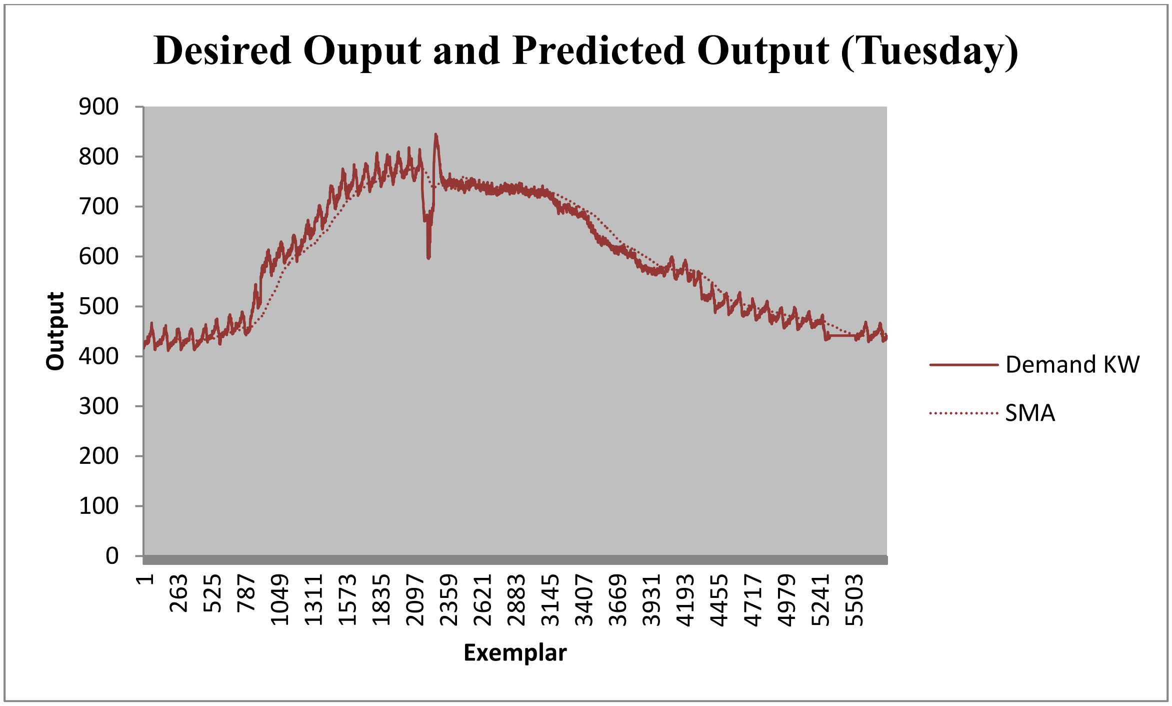


Figure 5-9 Desired output and predicted SMA output (Tuesday).

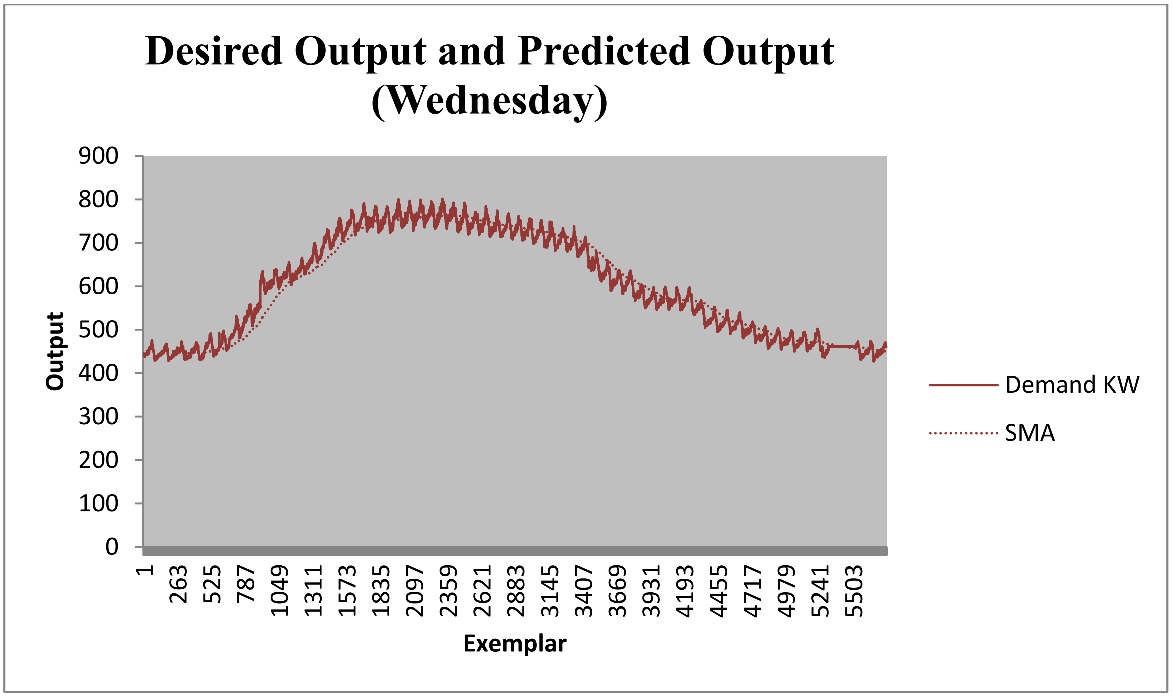


Figure 5-10 Desired output and predicted SMA output (Wednesday).

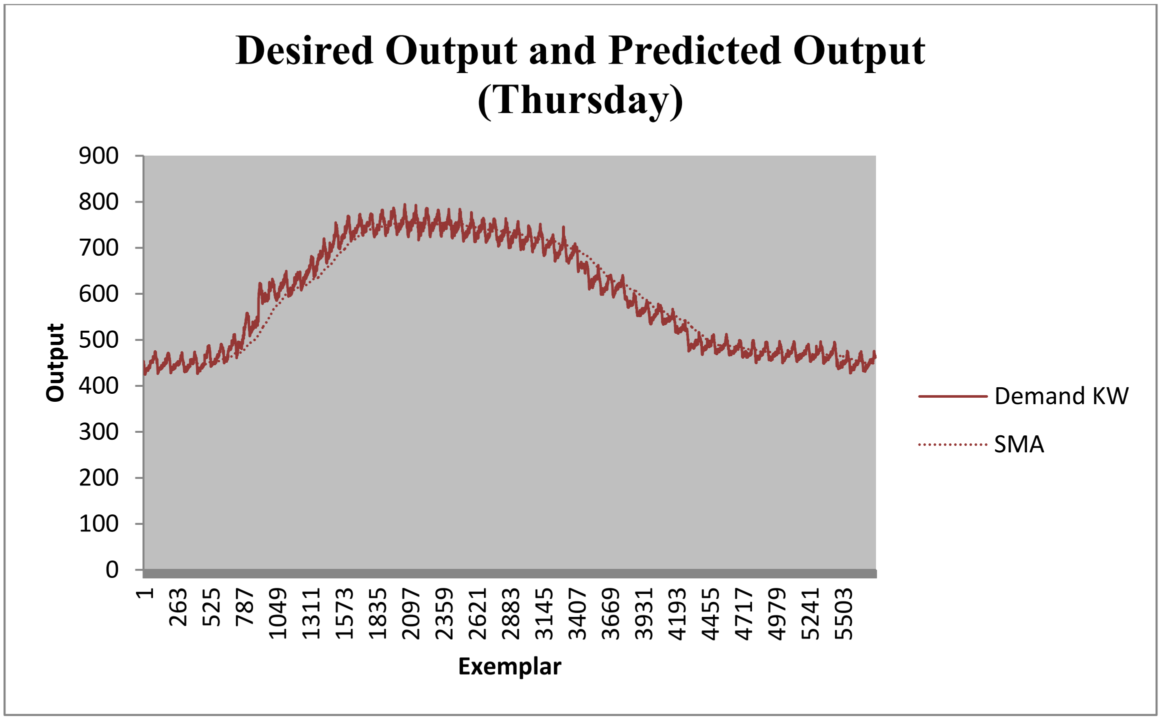


Figure 5-11 Desired output and predicted SMA output (Thursday).

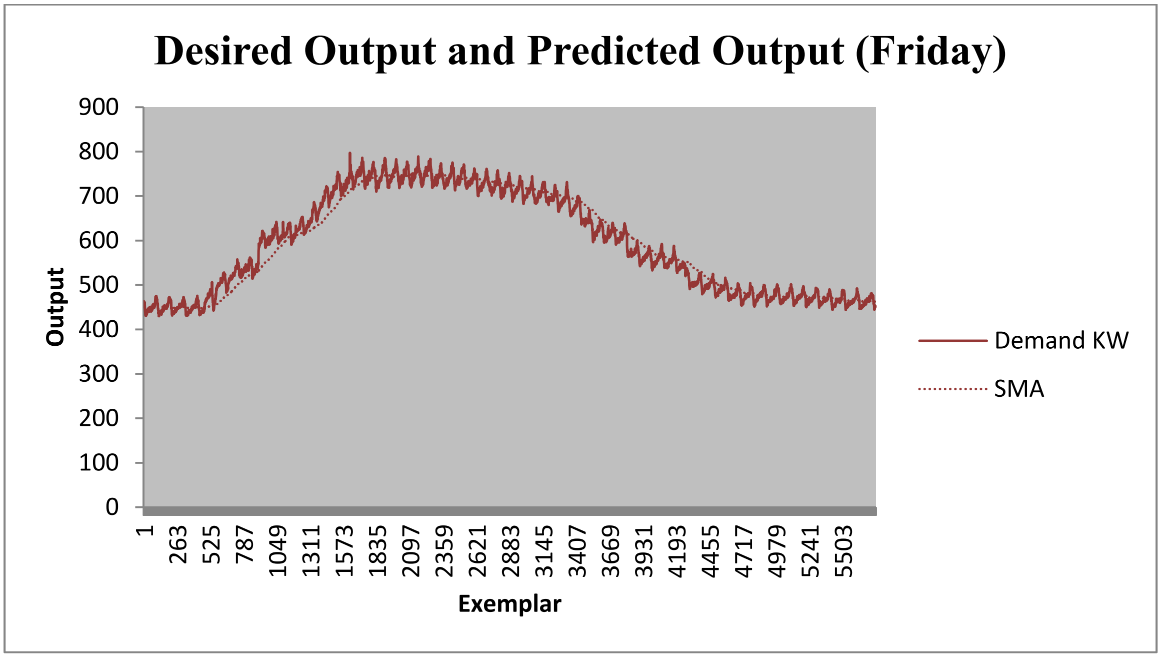


Figure 5-12 Desired output and predicted SMA output (Friday).

Linear Regression

For another benchmark performance comparison model, linear regression using Minitab ver. 16.2.4 was applied to the original 90-day data set. The regression equation resulted as:

Demand KW = 128 – (0.00799 * Time Type)
– (15.8 * Day Type)
+ (1.44 * HVAC KW)
+ (1.35 * Exterior Temperature)
+ (1.41 * Humidity)

The regression equation was then applied to the same twenty-four hour data test periods with respective forecast errors depicted in Table 2. The average and maximum forecast error for all tested day types using linear regression was 17.3% and 45.1% respectively. Table 5-1 depicts the average sixty-minute linear regression forecasting errors realized for both work weeks.

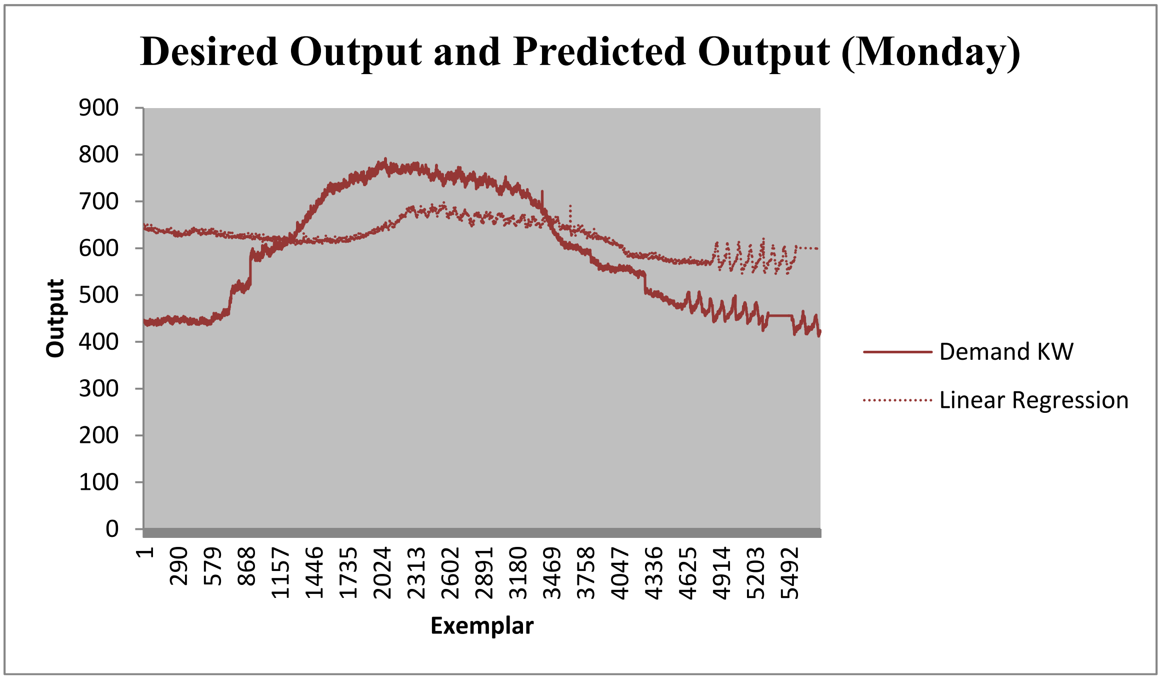


Figure 5-13 Desired output and predicted Linear Regression output (Monday).

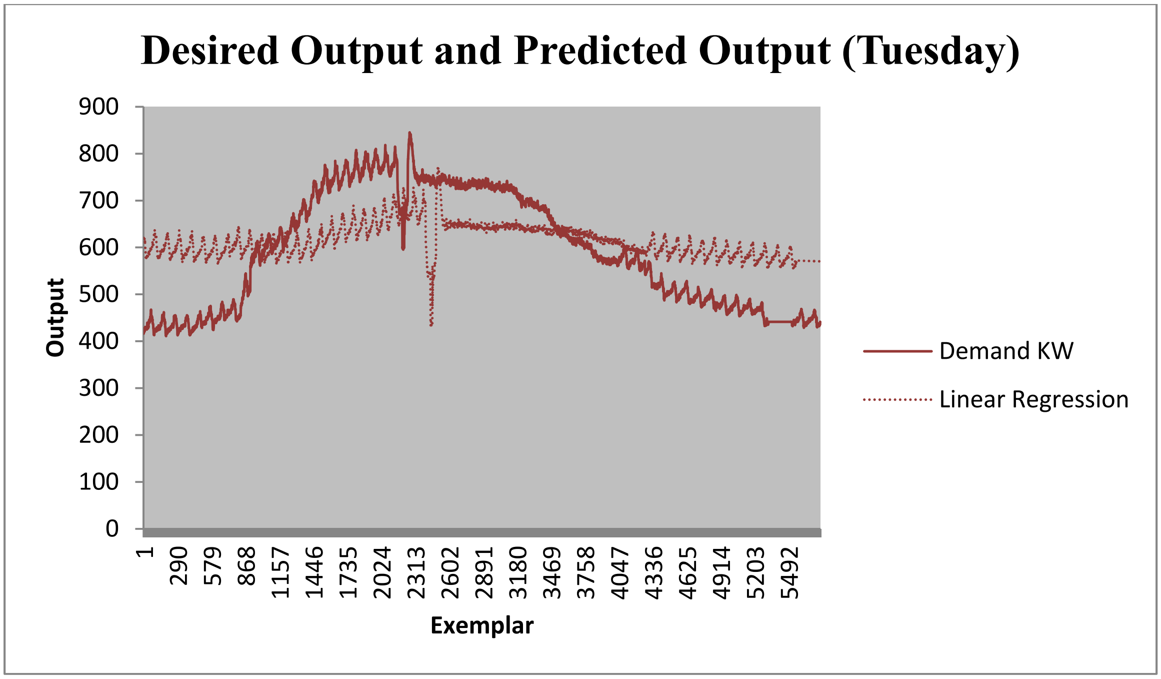


Figure 5-14 Desired output and predicted Linear Regression output (Tuesday).

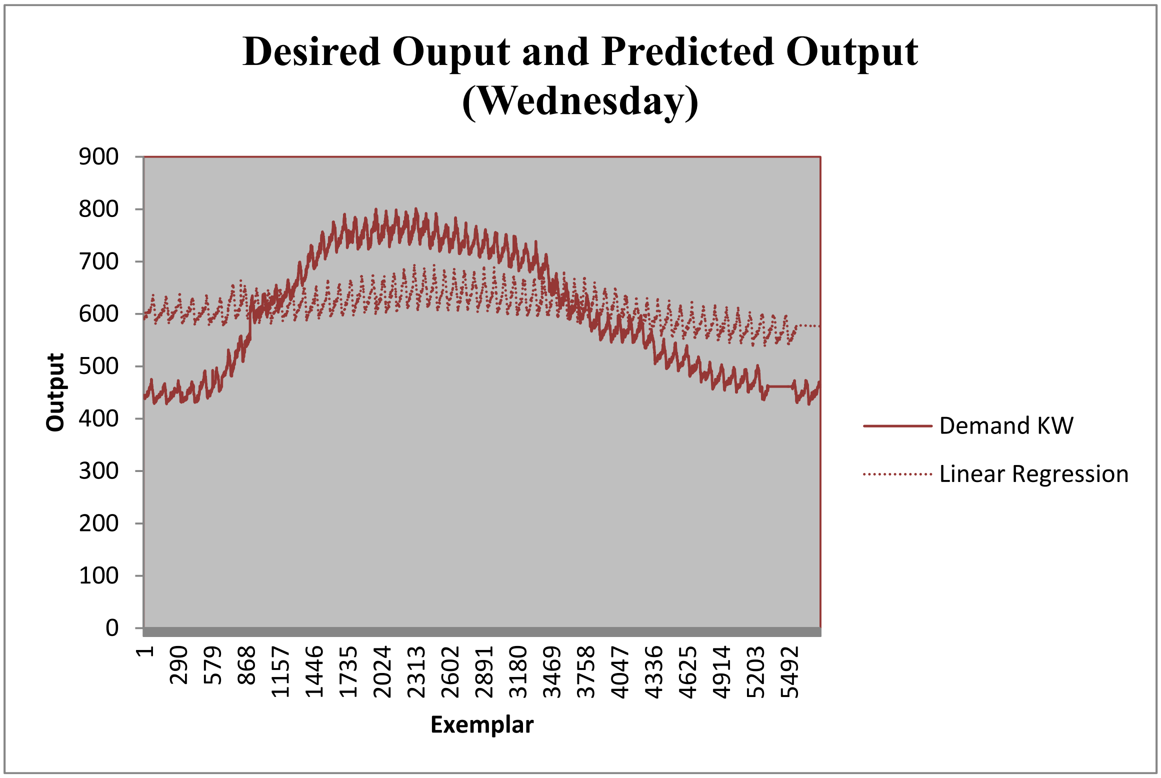


Figure 5-15 Desired output and predicted Linear Regression output (Wednesday).

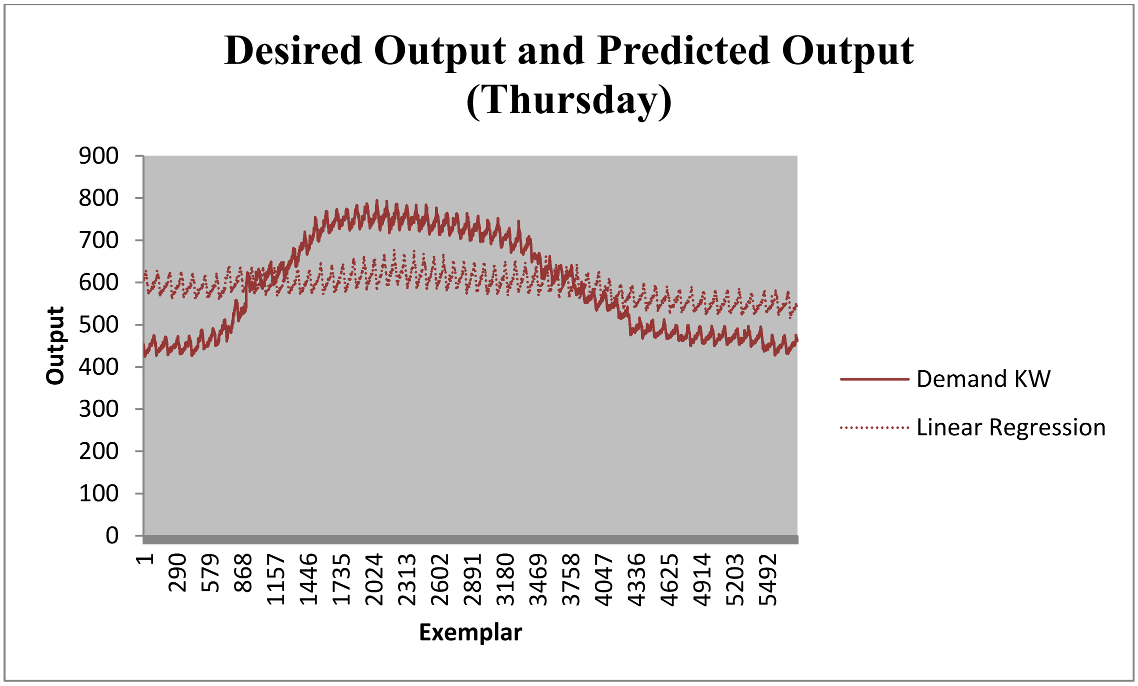


Figure 5-16 Desired output and predicted Linear Regression output (Thursday).

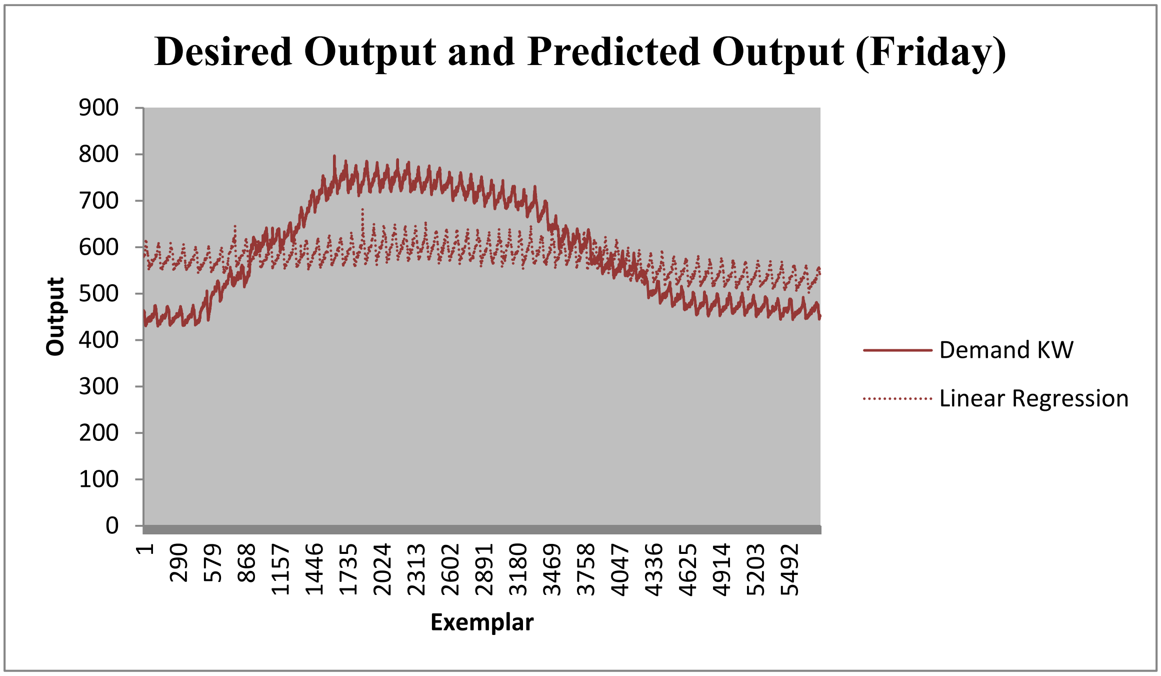


Figure 5-17 Desired output and predicted Linear Regression output (Friday).

Multivariate Adaptive Regression Splines (MARSplines)

Also, for ANN model performance testing, Multivariate Adaptive Regression Splines (MARSplines) were implemented using StatSoft's STATISTICA ver. 12 and were applied to the original 90-day data set. MARSplines, developed by Friedman in 1991, were used here as a nonparametric regression procedure and do not assume any functional relationship between the dependent and independent variables. Rather, MARSplines relate the dependent and independent variables using a set of coefficients and basis functions derived directly from the regression data. MARSplines create regression equations for multiple unique regions within the input space. Furthermore, when the relationship between the predictors and the dependent variables is non-monotone and difficult to approximate with parametric models, MARSplines are capable of creating effective forecast models.

The STATISTICA MARSplines Regression equation resulted as:

$$
\begin{aligned}
\textbf{Demand KW} = {} & 959.2215 + 4.8276 * \text{MAX}(0, \textbf{HVAC KW} - 298.9255) \\
& - 4.1129 * \text{MAX}\ (0, 298.9255 - \textbf{HVAC KW}) \\
& - 0.0098 * \text{MAX}\ (0, \textbf{Time Type} - 2038) \\
& - 0.1079 * \text{MAX}\ (0, 2038 - \textbf{Time Type}) \\
& - 16.1355 * \text{MAX}\ (0, \text{Day Type} - 1) \\
& - 2.5124 * \text{MAX}\ (0, \textbf{HVAC KW} - 215.9989) \\
& - 0.8278 * \text{MAX}\ (0, \textbf{External Temperature} - 62.1023) \\
& - 0.9916 * \text{MAX}\ (0, 62.1023 - \textbf{External Temperature}) \\
& - 0.7992 * \text{MAX}\ (0, \textbf{Humidity} - 33.512) \\
& - 0.8891 * \text{MAX}\ (0, 33.512 - \textbf{Humidity}) \\
& - 0.0721 * \text{MAX}\ (0, \textbf{Time Type} -2695) \\
& + 0.0655 * \text{MAX}\ (0, \textbf{Time Type} - 4775) \\
& - 3.0973 * \text{MAX}\ (0, \textbf{HAVAC KW} - 348.6209) \\
& - 4.9213 * \text{MAX}\ (0, \textbf{External Temperature} - 67.428) \\
& + 3.6512 * \text{MAX}(0, \textbf{External Temperature} - 73.9144) \\
& - 4.1132 * \text{MAX}\ (0, \textbf{Humidity} - 37.128) \\
& + 3.7511 * \text{MAX}(0, \textbf{Humidity} - 41.212)
\end{aligned}
$$

The MARSplines equation was then applied to the same twenty data test periods with respective forecast errors depicted in Table 2. The average and maximum forecast error for all tested day types using MARSplines was 7.0% and 22.5% respectively. Table 5-1 depicts the average sixty-minute MARSpines forecasting errors realized for both work weeks.

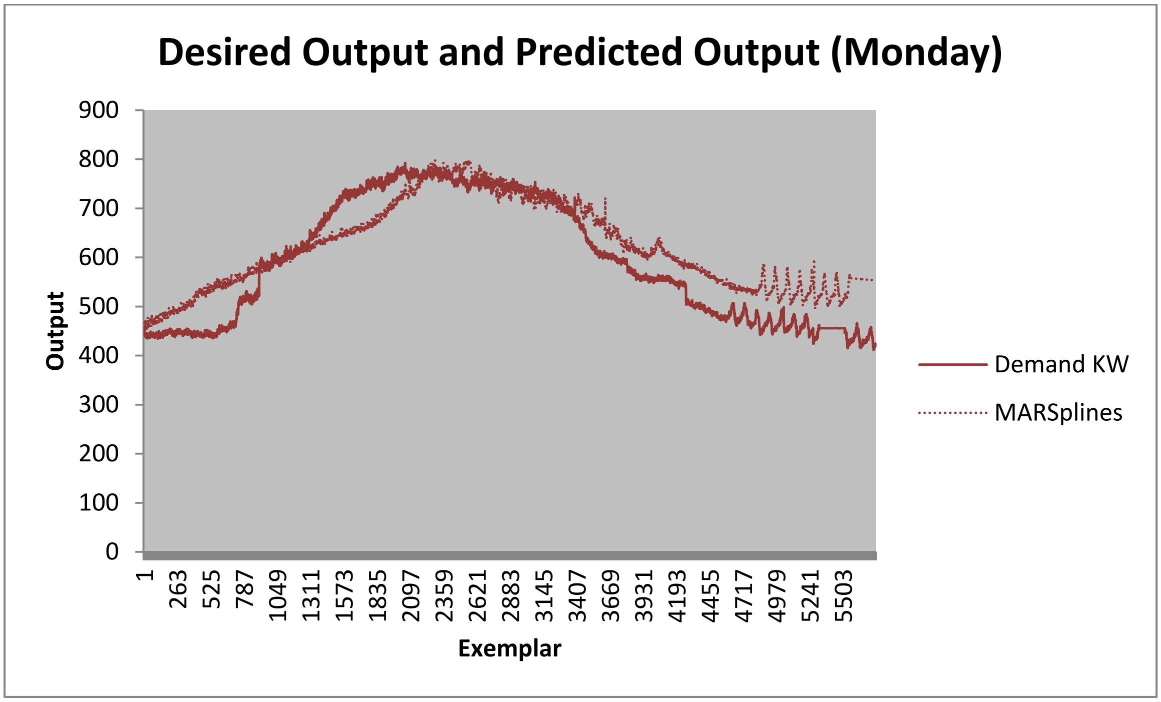


Figure 5-18 Desired output and predicted MARSplines output (Monday).

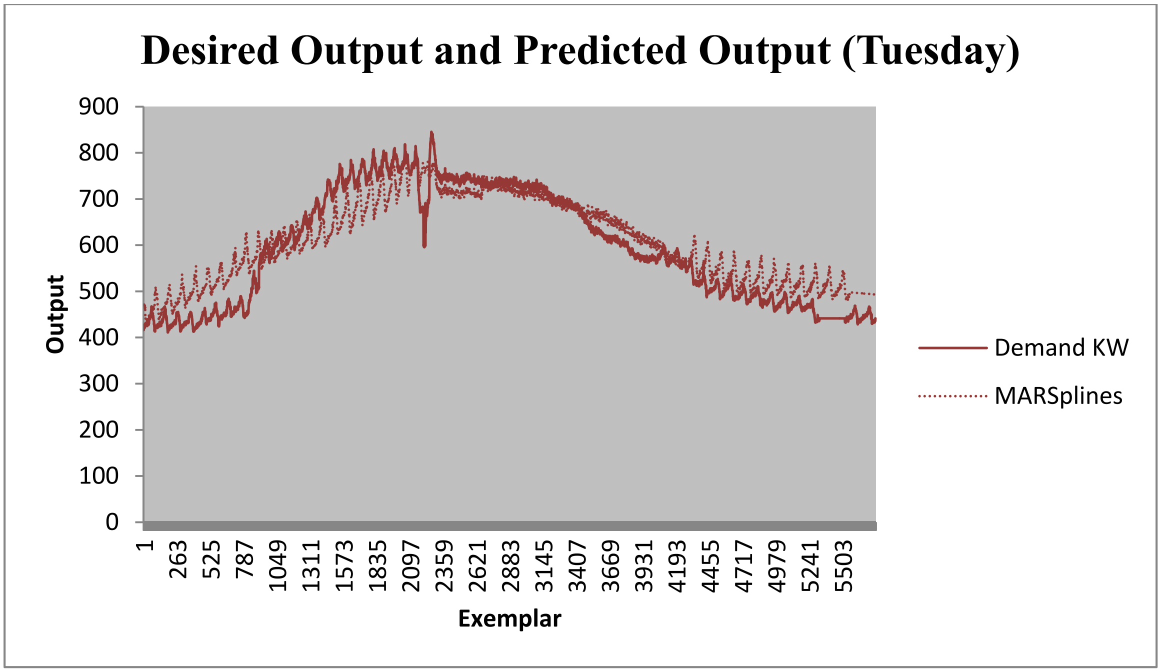


Figure 5-19 Desired output and predicted MARSplines output (Tuesday).

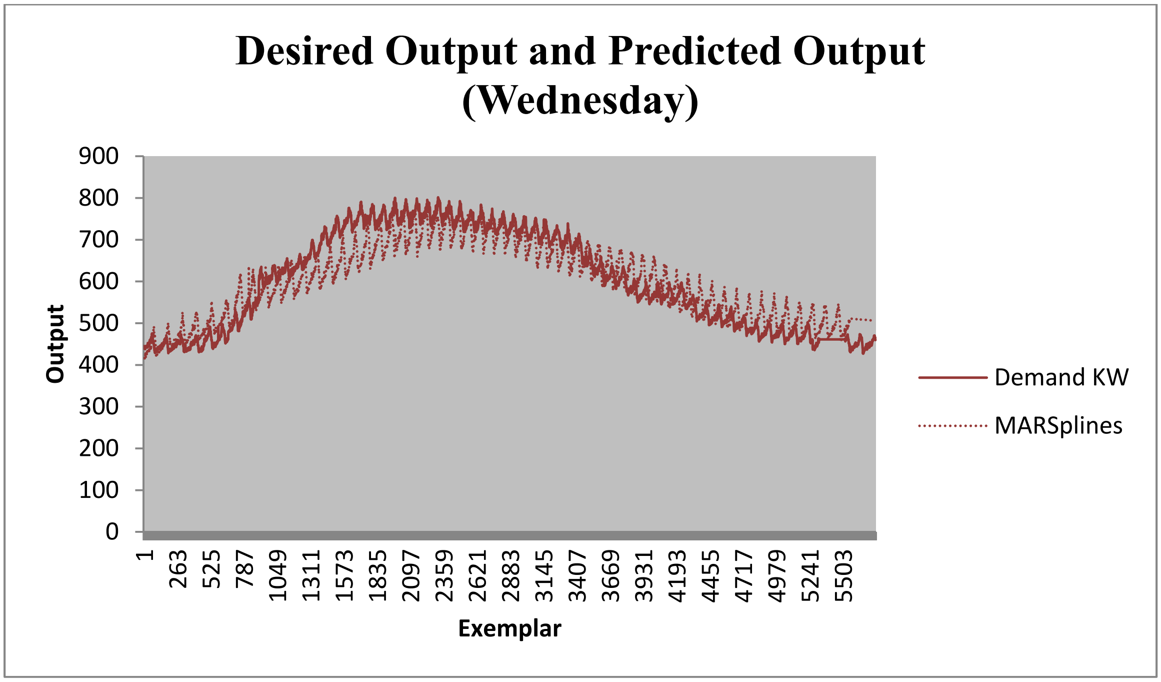


Figure 5-20 Desired output and predicted MARSplines output (Wednesday).

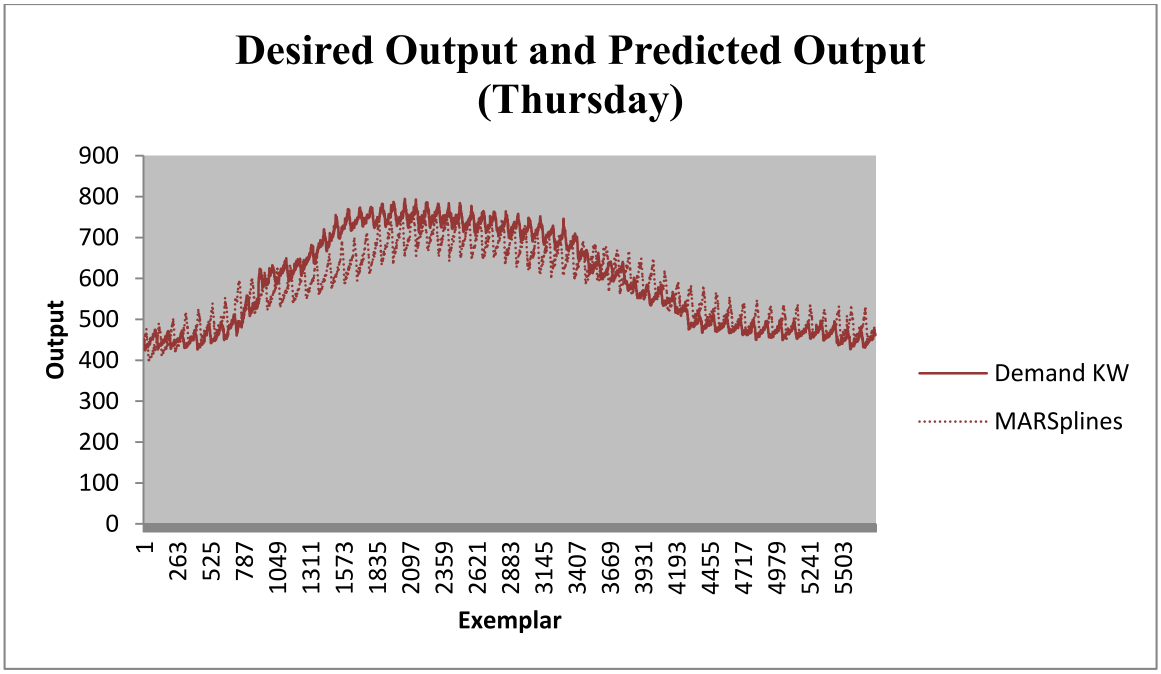


Figure 5-21 Desired output and predicted MARSplines output (Thursday).

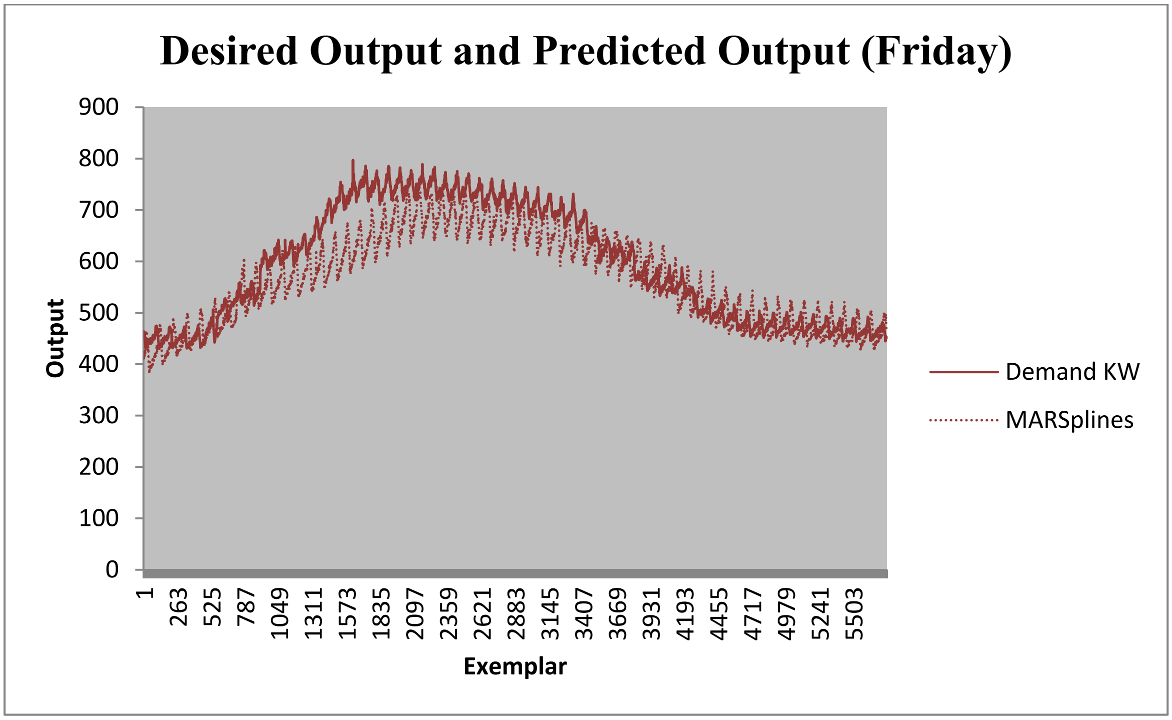


Figure 5-22 Desired output and predicted MARSplines output (Friday).

Table 5-1. Sixty-minute forecast errors (%) of electrical KW demand vs. actual.

	Model	12:00 AM	1:00 AM	2:00 AM	3:00 AM	4:00 AM	5:00 AM	6:00 AM	7:00 AM	8:00 AM	9:00 AM	10:00 AM	11:00 AM	12:00 PM	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM	6:00 PM	7:00 PM	8:00 PM	9:00 PM	10:00 PM	11:00 PM
Monday																									
	SMA	1.5	1.9	17.2	22.3	13.4	17.5	10.4	5.9	3.5	2.3	2.9	3.5	4.1	11.3	12.9	10.5	6.9	12.1	8.2	5.5	4.9	3.5	4.1	5.9
	Regression	45.1	43.2	42.1	22.1	7.2	6.2	10.5	17.2	18.4	11.2	15.4	12.9	9.7	9.2	5.4	7.2	12.1	15.4	16.2	17.2	25.4	28.2	31.5	39.5
	MARSplines	6.5	12.2	19.2	7.2	2.1	2.1	8.2	8.5	4.2	2.5	5.4	2.5	2.9	3.1	8.2	7.5	9.2	11.2	16.3	11.5	14.1	15.2	14.8	22.5
	ANN	0.8	2.2	4.9	3.9	5.2	5.3	3.9	5.1	4.2	4.5	3.9	3.3	2.1	1.9	3.6	3.7	3.5	3.5	4.1	3.2	2.1	2.3	2.9	2.1
Tuesday																									
	SMA	6.2	5.2	15.2	26.2	18.2	17.2	9.1	5.5	7.9	3.5	3.9	3.2	4.5	8.2	10.2	10.8	6.2	11.2	7.5	3.9	4.1	7.9	3.1	2.9
	Regression	37.1	37.3	38.2	33.1	5.9	10.2	16.2	15.9	16.3	12.2	17.2	13.2	10.8	7.9	3.9	3.8	8.2	7.2	19.2	22.2	22.4	22.1	32.1	29.2
	MARSplines	6.9	11.2	15.1	15.2	5.9	6.2	10.2	7.9	2.9	8.2	10.2	2.5	2.9	1.3	3.9	5.9	6.1	2.2	7.2	5.9	7.2	8.0	14.2	10.2
	ANN	3.2	4.1	5.5	3.9	1.1	1.9	3.2	3.5	3.2	3.3	3.9	4.1	4.0	3.2	5.0	4.2	5.2	3.2	6.8	4.6	4.2	4.1	4.2	3.0
Wednesday																									
	SMA	3.5	4.5	19.2	23.1	13.2	12.2	9.2	5.2	2.9	2.9	3.9	4.2	4.3	7.2	11.2	13.8	7.2	9.2	9.3	5.5	5.9	5.2	4.9	2.2
	Regression	33.2	38.2	25.2	19.8	19.2	12.0	13.2	15.1	12.3	15.4	15.2	12.2	10.9	10.7	5.5	5.2	5.3	7.5	15.2	17.2	21.2	25.4	23.1	33.2
	MARSplines	2.8	6.5	6.5	6.3	5.9	7.8	11.0	9.0	5.9	4.2	2.9	2.8	3.8	4.1	4.2	5.2	7.5	6.9	5.9	7.2	6.2	8.2	6.0	11.2
	ANN	2.0	1.4	3.2	4.2	8.2	8.1	6.2	4.9	3.3	3.3	3.5	2.2	2.1	2.1	2.1	2.2	6.2	7.1	7.2	5.4	4.2	4.5	4.9	5.4
Thursday																									
	SMA	2.7	4.9	17.2	21.0	13.2	13.9	9.1	4.4	3.8	1.9	2.2	2.8	3.8	7.2	11.2	11.6	12.2	11.4	7.2	4.8	3.9	3.8	3.8	2.2
	Regression	32.1	30.2	24.3	15.2	5.5	7.2	18.2	20.1	22.1	21.2	18.2	17.2	16.5	15.4	7.2	4.9	5.3	9.2	12.5	17.2	17.1	16.8	18.1	25.2
	MARSplines	5.9	5.3	4.9	5.9	6.2	8.2	11.5	10.2	9.2	4.9	4.7	5.9	6.1	5.9	5.3	5.9	5.8	5.9	4.9	4.2	5.1	5.2	4.2	4.2
	ANN	2.9	2.7	2.7	3.5	6.5	7.6	6.1	2.9	1.1	1.2	1.0	1.0	0.9	0.8	0.7	3.3	5.1	4.9	7.1	7.2	7.2	7.6	6.4	4.9
Friday																									
	SMA	1.9	11.1	18.2	17.2	13.5	13.2	6.9	2.5	3.5	3.9	4.2	3.0	3.0	6.5	9.5	10.2	6.2	10.2	8.2	3.2	2.0	2.1	2.4	3.7
	Regression	26.6	28.2	19.2	9.2	6.2	10.2	19.2	22.1	22.1	23.5	22.1	17.2	15.1	14.5	9.2	5.0	5.1	5.2	9.1	15.2	14.9	14.5	14.2	17.2
	MARSplines	6.4	4.2	5.2	6.1	7.1	12.5	11.2	11.3	6.9	6.7	7.4	4.9	4.9	6.2	4.9	4.7	5.2	5.8	5.8	3.9	5.9	5.7	5.2	4.9
	ANN	3.3	3.2	4.1	5.9	7.1	8.2	8.2	2.9	1.9	1.1	1.1	1.2	3.1	3.2	3.9	4.2	4.1	3.9	3.8	3.7	2.9	2.8	3.2	3.9

The average forecast errors for the two week testing period for all tested methods are depicted in Figure 5-22. Additionally, the average sixty-minute forecast errors (%) for each weekday type are described in table 5-2.

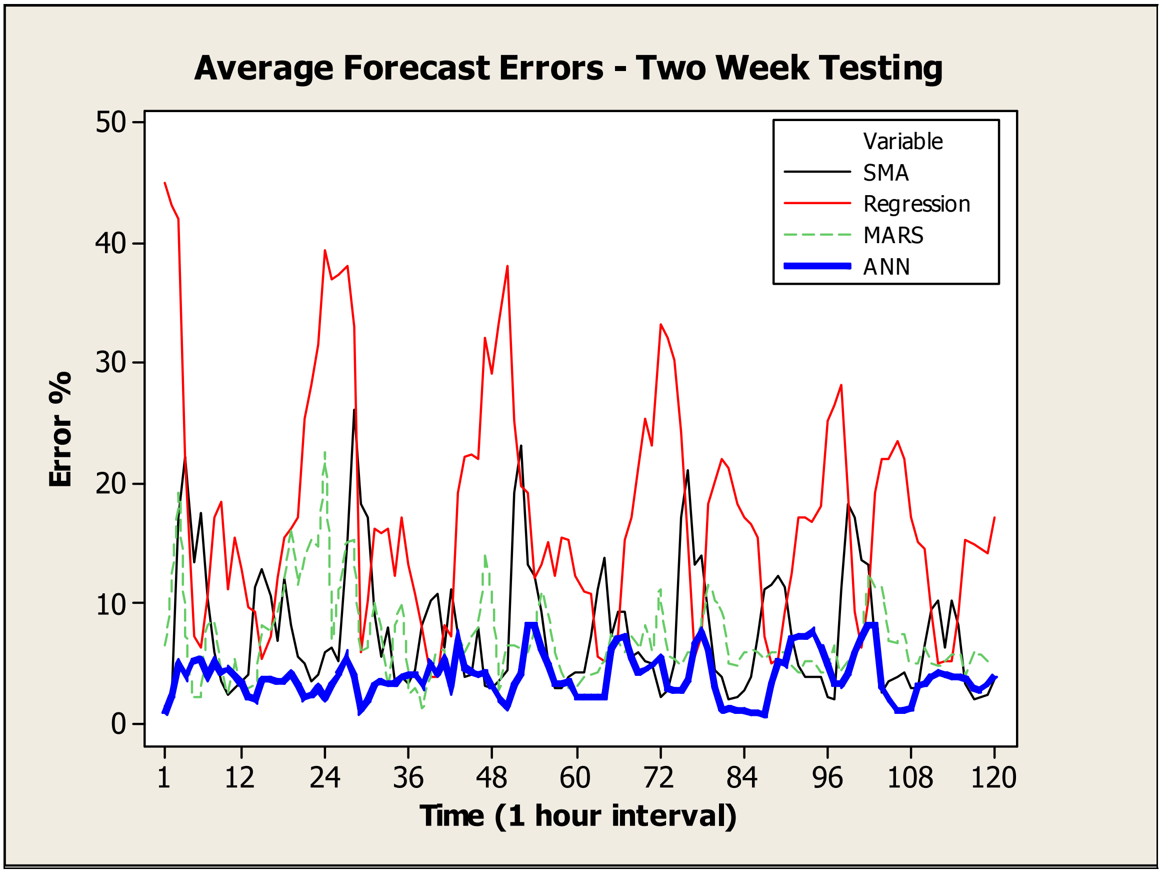


Figure 5-23. Two week forecast errors predicted electrical KW demand vs. actual.

Table 5-2. Average sixty-minute forecast errors (%) (twenty four-hour period) electrical KW demand vs. actual.

		SMA	Regression	MARSplines	ANN
Monday					
	Average	8.0	19.5	9.0	3.4
	Max	22.3	45.1	22.5	11.3
Tuesday					
	Average	8.4	18.4	7.4	3.9
	Max	26.2	38.2	15.2	12.8
Wednesday					
	Average	7.9	17.1	6.2	4.3
	Max	23.1	38.2	11.2	14.2
Thursday					
	Average	7.5	16.5	6.1	4.0
	Max	21.0	32.1	11.5	16.6
Friday					
	Average	6.9	15.2	6.4	3.8
	Max	18.2	28.2	12.5	18.2
Week					
	Week Average	7.7	17.3	7.0	3.9
	Week Max	26.2	45.1	22.5	18.2

Controlled Peak Demand Through Simulated Intelligent Electrical Loading

Using the successfully developed ANN with acceptable forecast error performance, an attempt was made at controlling peak demand through simulated intelligent electrical loading. As was discovered during the development of the ANN, the HVAC load consumption was variable and comprised approximately forty percent of total building electrical demand (approximately 300 KW). Peak demand occurrences observed during the data logging demonstrated constant and full use of the HVAC system and its respective subcomponents. By staggering the main chillers and their subcomponents on a 30 minute cycle through simulation, 150 KW (approximately 40%) of total building load was theoretically curtailable. Given possible extra loading on the single chiller during the 30 minute cycle, it was conservatively estimated that approximately 12.5% of total building KW could be shed during periods of experienced peak demand.

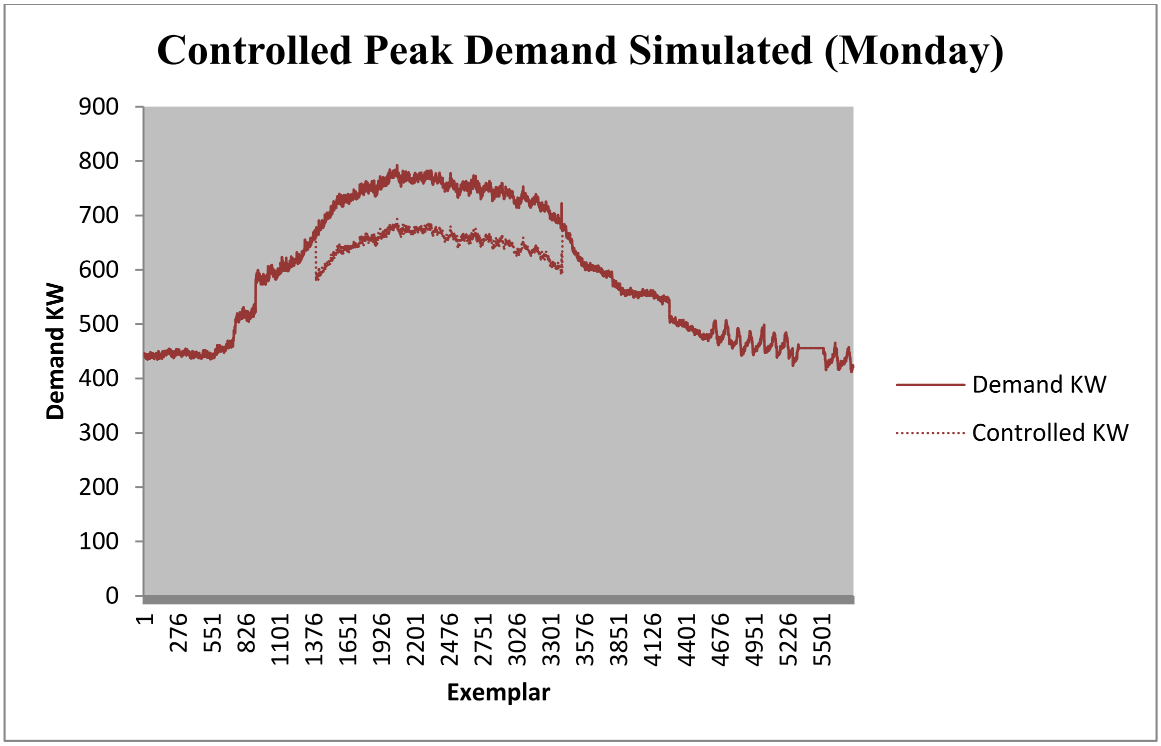


Figure 5-24. Controlled peak demand simulated (Monday).

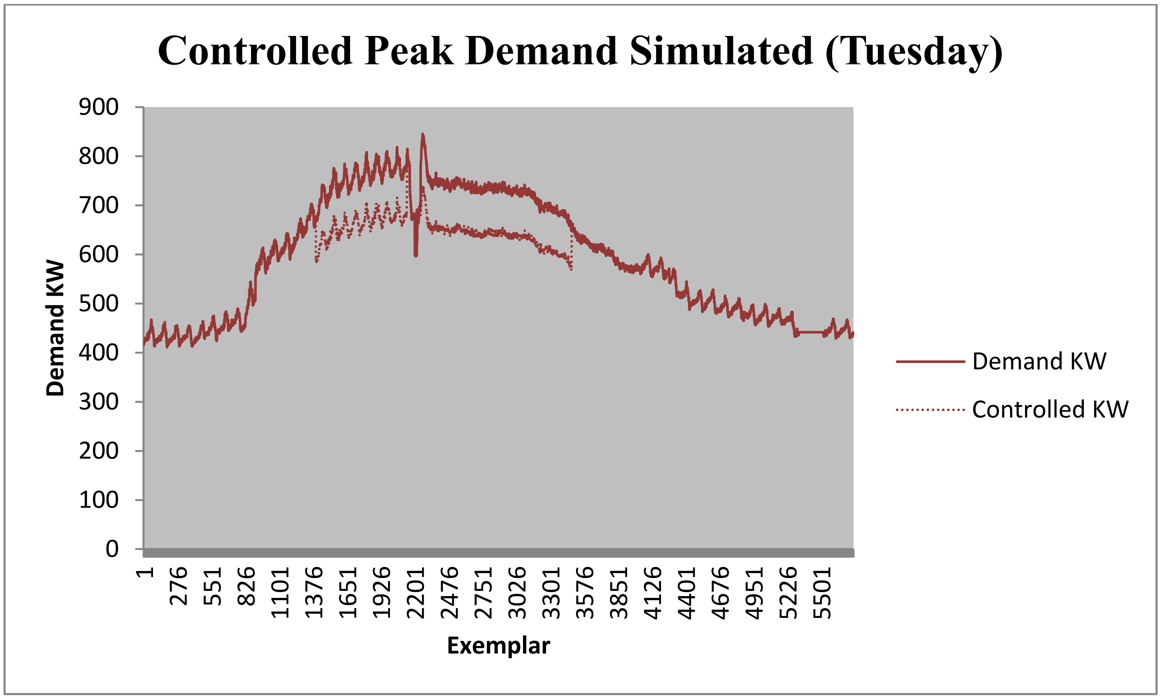


Figure 5-25. Controlled peak demand simulated (Tuesday).

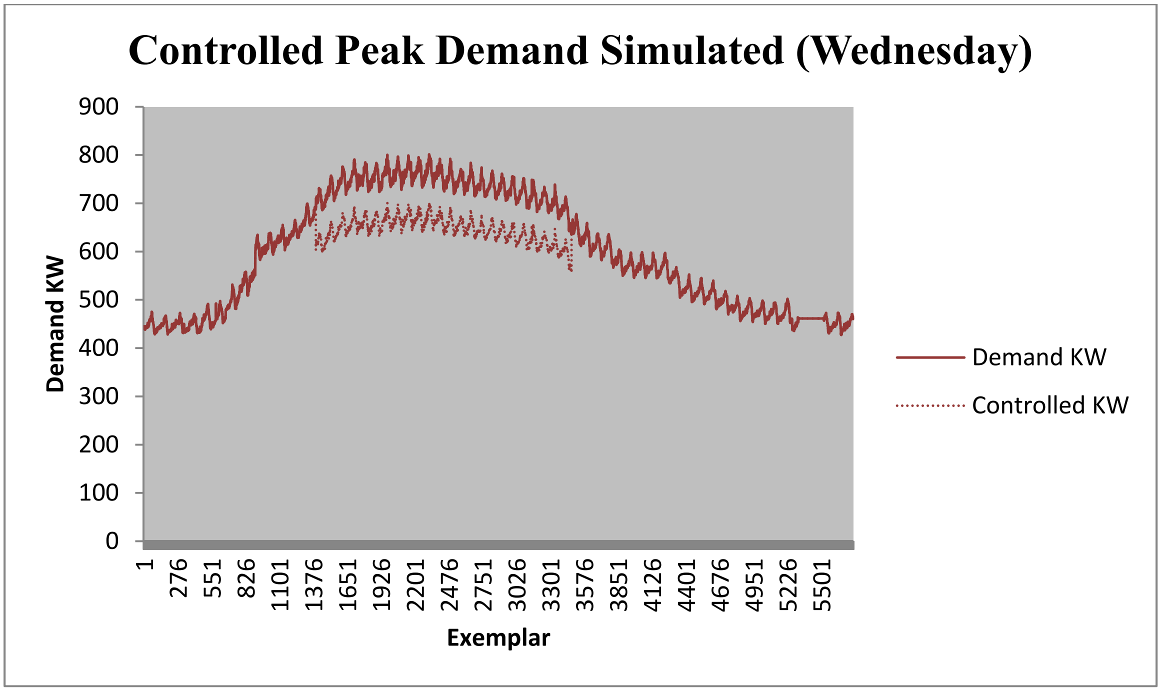


Figure 5-26. Controlled peak demand simulated (Wednesday).

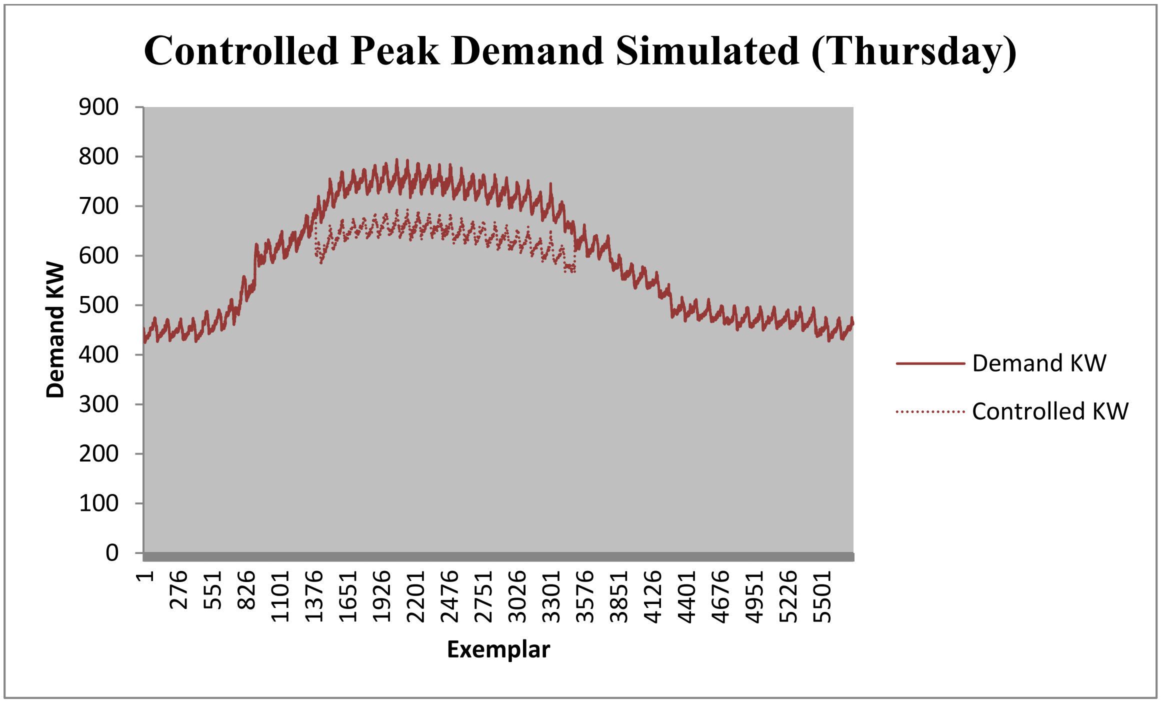


Figure 5-27. Controlled peak demand simulated (Thursday).

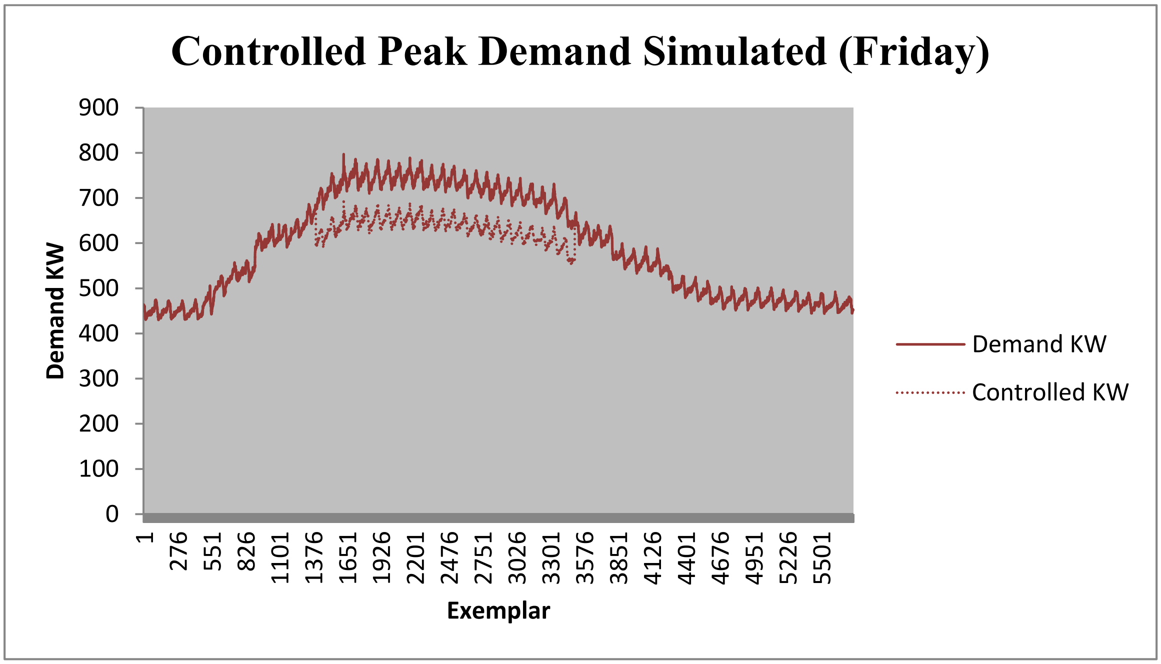


Figure 5-28. Controlled peak demand simulated (Friday).

CHAPTER 6: CONCLUSION

This research proposes a real-time energy monitoring system prototype to forecast peak demand in a large government building in an effort to augment current ICS DR programs. The proposed methodology aims to predict total building power KW demand sixty minutes into the future, thereby giving building management ample time to temporarily curtail a portion of building power consumption in order to minimize experienced peak demand (KW) during a given billing cycle. To achieve this, the model collects detailed electrical consumption data in a large government building over a ninety-day time period which are then fed to an ANN for training, cross-validation, and finally prediction.

The approach in this research used ANNs because of the extraordinary ability of ANNs to make sense of complicated or imprecise non-linear, non-stationary, and/or chaotic data which cannot be easily modeled. ANNs can extract patterns, detect complex trends, do not require a priori problem space assumptions, and do not require information regarding statistical distribution. ANNs also demonstrate adaptability to new situations through adaptive learning. ANNs produce unique representations of the information during its learning process, operate in real-time, and are capable of parallel computation. Furthermore, ANNs have inherent built-in fault tolerance as a result of redundant information coding.

The real-time energy monitoring system developed to capture the building's electrical consumption demonstrated high resolution capability of recording every main service entrance and HVAC circuit's (3-phase) power consumption in the building under study with a sample rate of 15 seconds. Operating within a customized LAN, Magnelab current transformer sensors, National Instruments DAQ devices, and National Instruments LabVIEW served as the backbone of the developed real-time energy monitoring system.

To examine and demonstrate the effectiveness of this research approach, experimental analysis was conducted on electrical consumption data recorded over a three-month period. Dynamic HVAC KW loads were able to offer accurate forecasting of total building demand. By also incorporating day type, time of day, exterior temperature, and humidity data into the developed ANN, the forecast error was minimized even further. Model performance was consistent throughout the test runs. Also, the developed ANN model was compared with alternate methods of prediction including SMA, linear regression and multivariate adaptive regression splines (MARSplines) and consistently performed better with an MAPE of 3.9% and AME of 18.2%. The SMA model performed well due to the static nature of the building's power consumption, but the MAPE was quite high at 26.2 %. Performing similarly, the MARSplines approach had an MAPE of 7.0% and AME of 22.5%. Finally, linear regression had the highest MAPE of 17.3% and highest AME of 45.1%.

Given sixty-minute forecast ability with low error using the ANN approach, it is theoretically possible for building management to temporarily curtail a portion of the building load whenever approaching a predetermined peak in demand. Due to seasonality effects on the building's HVAC load, acceptable peak demand loads differ from month to month. It is thus necessary for building management to determine an acceptable peak demand load maximum for each billing cycle. The real-time energy monitoring system together with the sixty-minute ANN forecast signals an upcoming breach of the predetermined peak demand load maximum. Building curtailment policy sheds unnecessary loads during these events in order to control overall peak loading and prevent an unwanted peak demand occurrence. Depending on building management, curtailment is manual and/or automated. This ANN model would be particularly useful for efficient and cost-effective peak demand energy management of multiple government or corporate building complexes (i.e. municipalities, corporate campuses, hospitals, universities, etc.) already under or capable of operating under one centralized energy management or building management system. The proposed model is entirely scalable and can be implemented for multi-building peak demand control. Existing sub-meters at such sites would serve to provide pertinent real-time and historical data to the ANN model and control procedures. Overall reduced peak demand would have a noticeable and beneficial financial impact on such building systems. If implemented on a large scale across many building systems including city municipalities and other large energy end-users, there would be added benefit to the electric utility provider and environment through efficient and reduced power generation capacity. Such reduction and efficient usage of power generation would undoubtedly contribute to the energy sustainability of local municipalities and their communities.

Finally, the model developed in this research was implemented and tested during one of two major local weather periods. The model proved effective during the particular weather period studied with similar performance expected during similar future weather periods. In order to measure the model’s robustness during a dissimilar weather period, additional testing and data capture during the other weather period type is necessary.

Future Work

Much work remains with regard to electrical peak demand reduction. In addition to ANNs, several other forecasting techniques currently exist and several new techniques are being explored, developed, and/or refined.

The ultimate goal of this research was to develop a practical method for predicting electrical peak demand to be employed by a building's end-user management. Ideally, the ANN approach proposed here would be most suited for those environments with some to a lot of variability with respect to peak demand (i.e. manufacturing). Here the building load present as mostly static with a variable HVAC Load. Another approach using a controlled-load policy approach may also be effective with controlling overall experienced peak demand. A comparison between a forecasting approach and load-control policy approach would be useful. With a load-control approach, however, peak-demand control would have to be automated; whereas, with an ANN forecasting approach, peak demand control could be accomplished through manual control.

Testing of the developed ANN during a unique weather period would also be beneficial in validating the model across unique weather periods. Additional testing in different weather locations would also be useful.

LIST OF REFERENCES

Albadi, M. H., & El-Saadany, E. F. (2008). A summary of demand response in electricity markets. Electric Power Systems Research, 78(11), 1989-1996.

Avci, M., Erkoc, M., Rahmani, A., & Asfour, S. (2013). Model predictive HVAC load control in buildings using real-time electricity pricing. Energy and Buildings.

Backer, D. (2007, May). Power Quality and Asset Management The Other. In Rural Electric Power Conference, 2007 IEEE (pp. C6-C6). IEEE.

BChydro. (2009). *Smart Grid at BChydro: Current Status.* http://grouper.ieee.org/groups/td/dist/da/doc/2009-07%20BC%20Hydro%20%20 SmartGRID-Status.pdf

Brown, R. E. (2008, July). Impact of smart grid on distribution system design. In Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE (pp. 1-4). IEEE.

Cappers, P., Goldman, C., & Kathan, D. (2010). Demand response in US electricity markets: Empirical evidence. Energy, 35(4), 1526-1535.

Croall, I. F., & Mason, J. P. (1992). Industrial Applications of Neural Networks: Project ANNIE Handbook. (pp. 7-16). Springer-Verlag New York, Inc..

Divan, D., & Johal, H. (2006, August). A smarter grid for improving system reliability and asset utilization. In Power Electronics and Motion Control Conference, 2006. IPEMC 2006. CES/IEEE 5th International (Vol. 1, pp. 1-7). IEEE.

Electricity Advisory Committee. (2009). Keeping the Lights on in a New World. US Department of Energy, Washington, DC, 50.

Farhangi, H. (2010). The path of the smart grid. Power and Energy Magazine, IEEE, 8(1), 18-28.

Faria, P., & Vale, Z. (2011). Demand response in electrical energy supply: An optimal real time pricing approach. Energy, 36(8), 5374-5384.

Fausett, L. V. (1994). Fundamentals of neural networks: architectures, algorithms, and applications (pp. 1-37). Englewood Cliffs: Prentice-Hall.

Framework, N. I. S. T. (2010). Roadmap for Smart Grid Interoperability Standards. NIST special publication, 1108.

Hart, David G. (2008). Proc. 2008 IEEE Power and Energy in the 21st Century, 1-2.

Hassoun, M. H. (1995). Fundamentals of artificial neural networks. (pp. 1-33). MIT Press.

Hu, Y. H., & Hwang, J. N. (Eds.). (2010). Handbook of neural network signal processing. (pp. 1-30). CRC press.

International Energy Agency. (2003). The Power to choose: demand response in liberalised electricity markets. OECD Publishing.

Kartalopoulos, S. V., & Kartakapoulos, S. V. (1997). Understanding neural networks and fuzzy logic: basic concepts and applications. (pp. 1-34). Wiley-IEEE Press.

Kirschen, D. S. (2003). Demand-side view of electricity markets. Power Systems, IEEE Transactions on, 18(2), 520-527.

Klement, Kevin C. (2005). Propositional Logic - History, *Internet Encyclopedia of Philosophy,* http://www.iep.utm.edu/prop-log/#H2

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115-133.

Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). Neural and adaptive systems: fundamentals through simulations with CD-ROM. John Wiley & Sons, Inc..

Rahman, S. (2009). Smart grid expectations [in my view]. Power and Energy Magazine, IEEE, 7(5), 88-84.

River, C. (2005). Primer on demand-side management with an emphasis on price-responsive programs. prepared for The World Bank by Charles River Associates, Tech. Rep.

Saint, B. (2009, April). Rural distribution system planning using Smart Grid Technologies. In Rural Electric Power Conference, 2009. REPC'09. IEEE (pp. B3-B3). IEEE.

Sioshansi, R., & Short, W. (2009). Evaluating the impacts of real-time pricing on the usage of wind generation. Power Systems, IEEE Transactions on, 24(2), 516-524.

Torriti, J., Hassan, M. G., & Leach, M. (2010). Demand response experience in Europe: Policies, programmes and implementation. Energy, 35(4), 1575-1583.

U.S. Department of Energy, The Smart Grid: An Introduction, http://energy.gov/oe/downloads/smart-grid-introduction-0

QDR, Q. (2006) U.S. Department of Energy. Benefits of demand response in electricity markets and recommendations for achieving them. A report to the United States Congress.

US Department of Energy. Benefits of demand response in electricity markets and recommendations for achieving them. Report to the United States Congress, February 2006, Available from: <http://eetd.lbl.gov/EA/EMS/reports/congress-1252d.pdf>, [accessed in September 2010].

Vale, Z., Pinto, T., Praça, I., & Morais, H. (2011). MASCEM: electricity markets simulation with strategic agents. Intelligent Systems, IEEE, 26(2), 9-17.

Walawalkar, R., Fernands, S., Thakur, N., & Chevva, K. R. (2010). Evolution and current status of demand response (DR) in electricity markets: Insights from PJM and NYISO. Energy, 35(4), 1553-1560.

Wang, J., Bloyd, C. N., Hu, Z., & Tan, Z. (2010). Demand response in China. Energy, 35(4), 1592-1597.

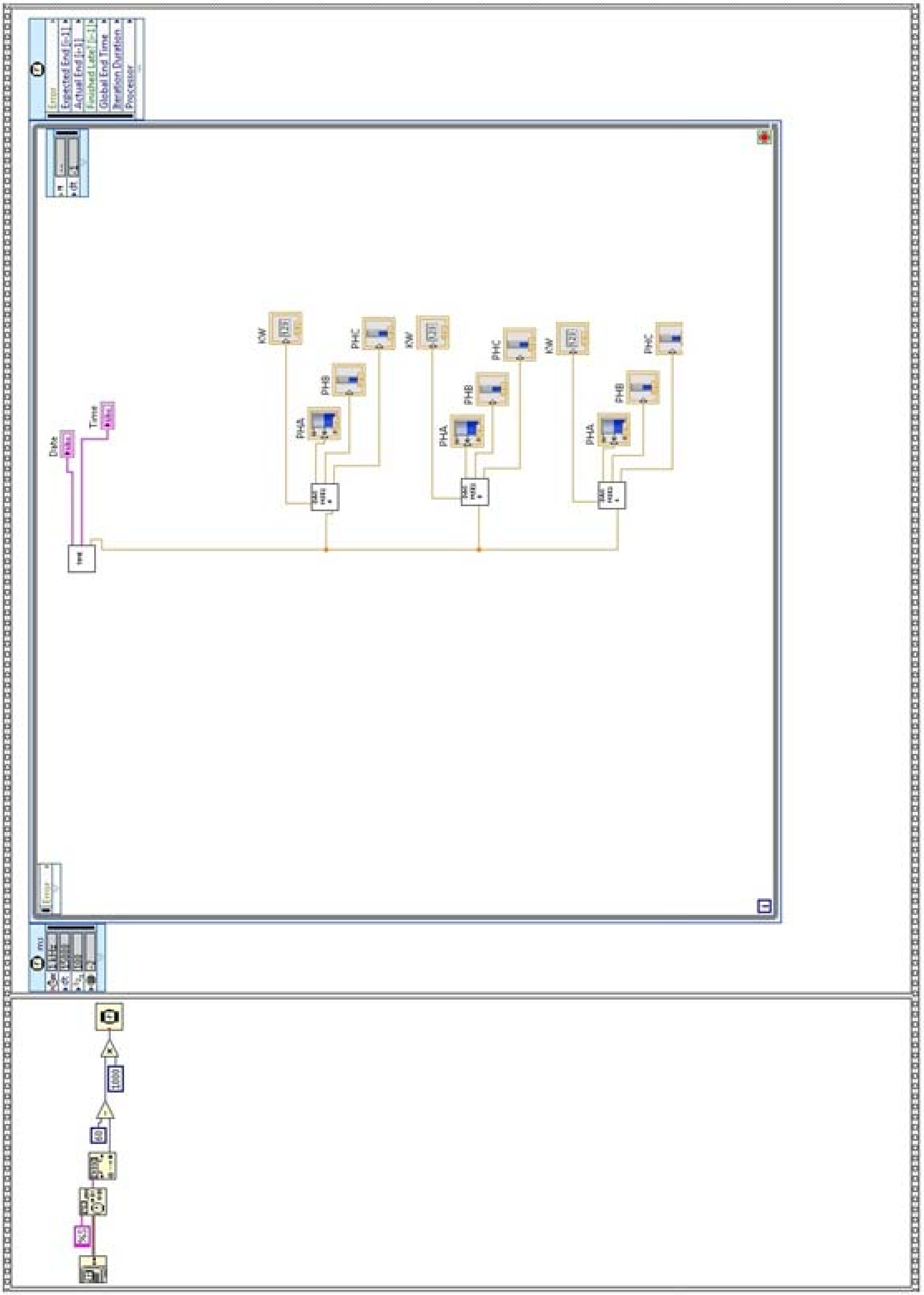
Wei, X., Yu-hui, Z., Jie-lin, Z. (2009). Energy-efficient Distribution in Smart Grid. *Sustainable Power Generation and Supply,* 1-6.

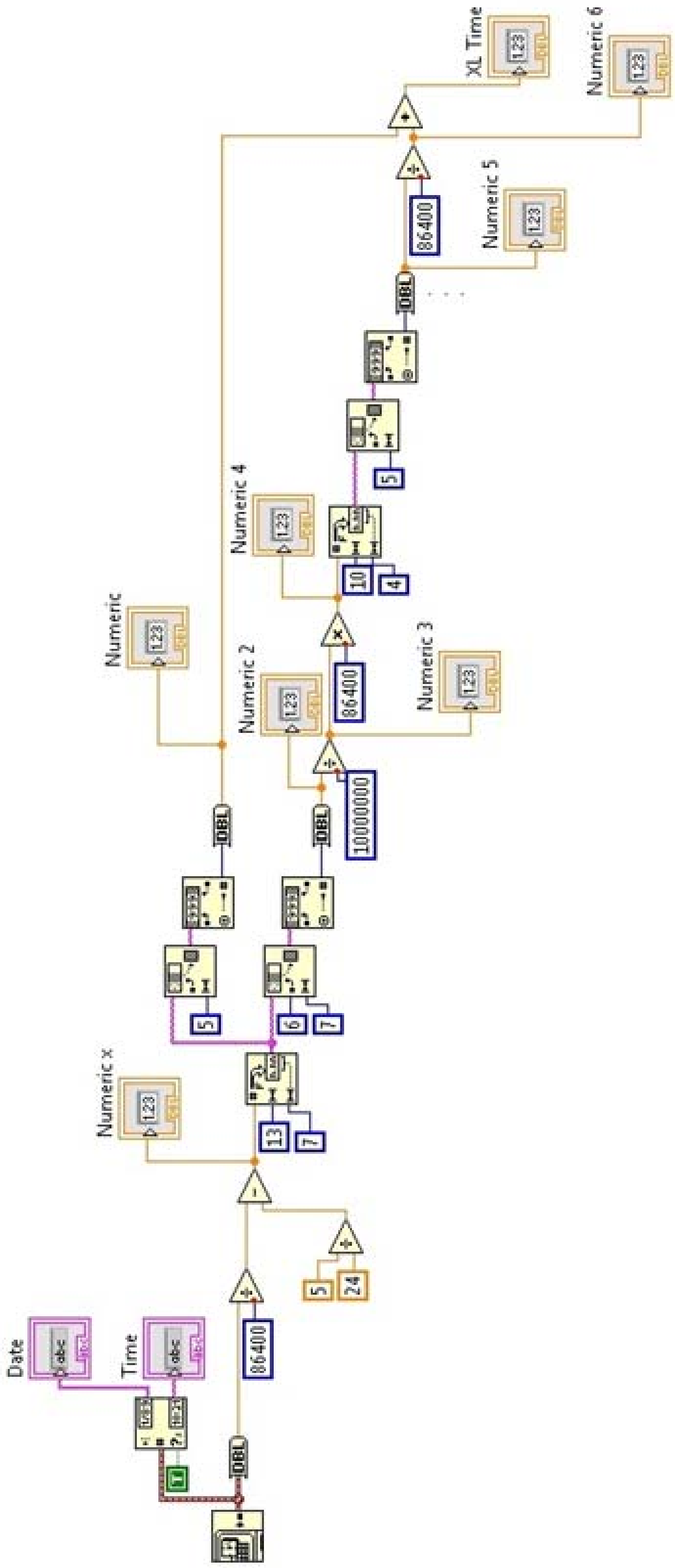
Woo, C. K., & Greening, L. A. (2010). Guest editors' introduction. Energy, 35(4), 1515-1517.

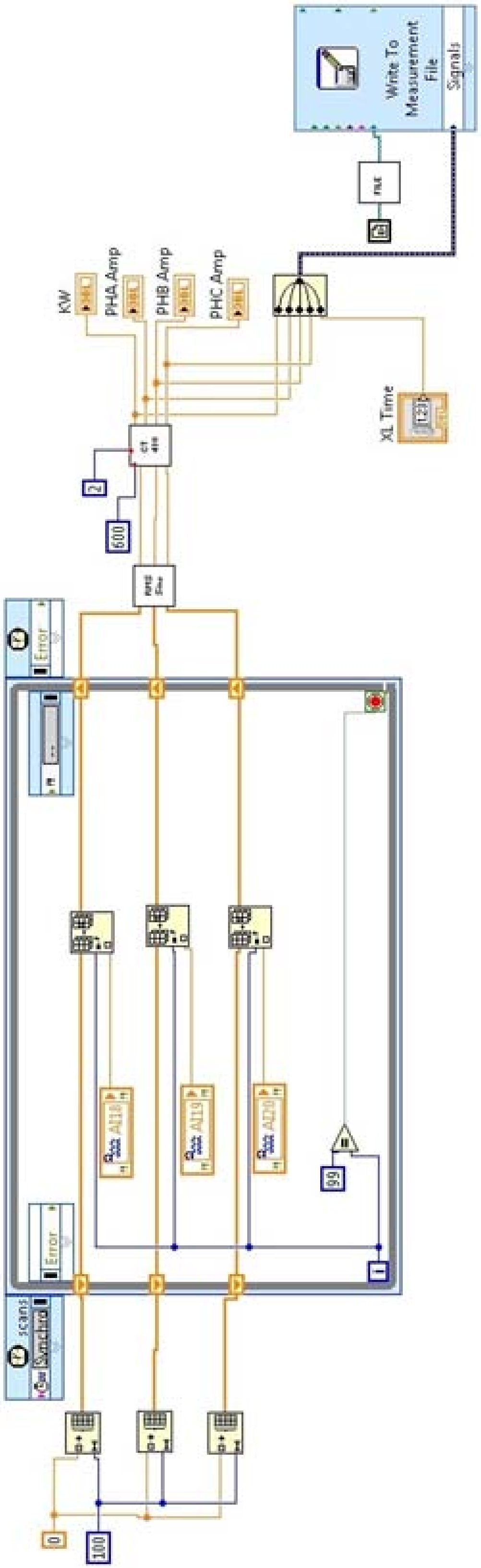
Yinger, R. J. (2006, May). Southern California Edison's Distribution Circuit of the Future. In Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES (pp. 339-341). IEEE.

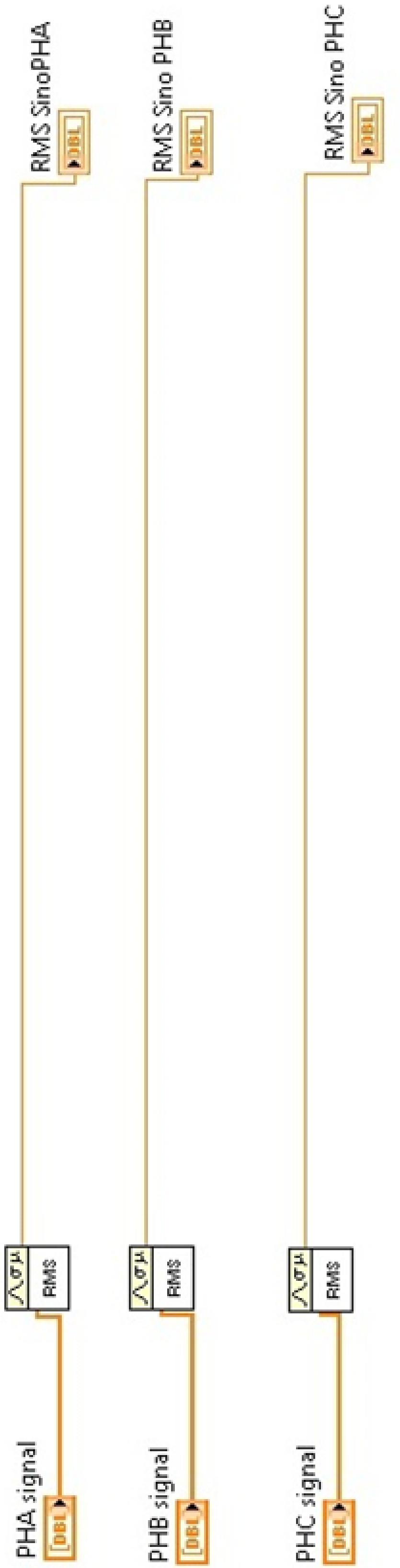
Youssef, H. A., Al-Makky, M. Y., & Abd-Elwahab, M. M. (2003). Evaluation of a proposed neural network predictive model for grind-hardening. Alexandria Engineering Journal, 42(4), 411-417.

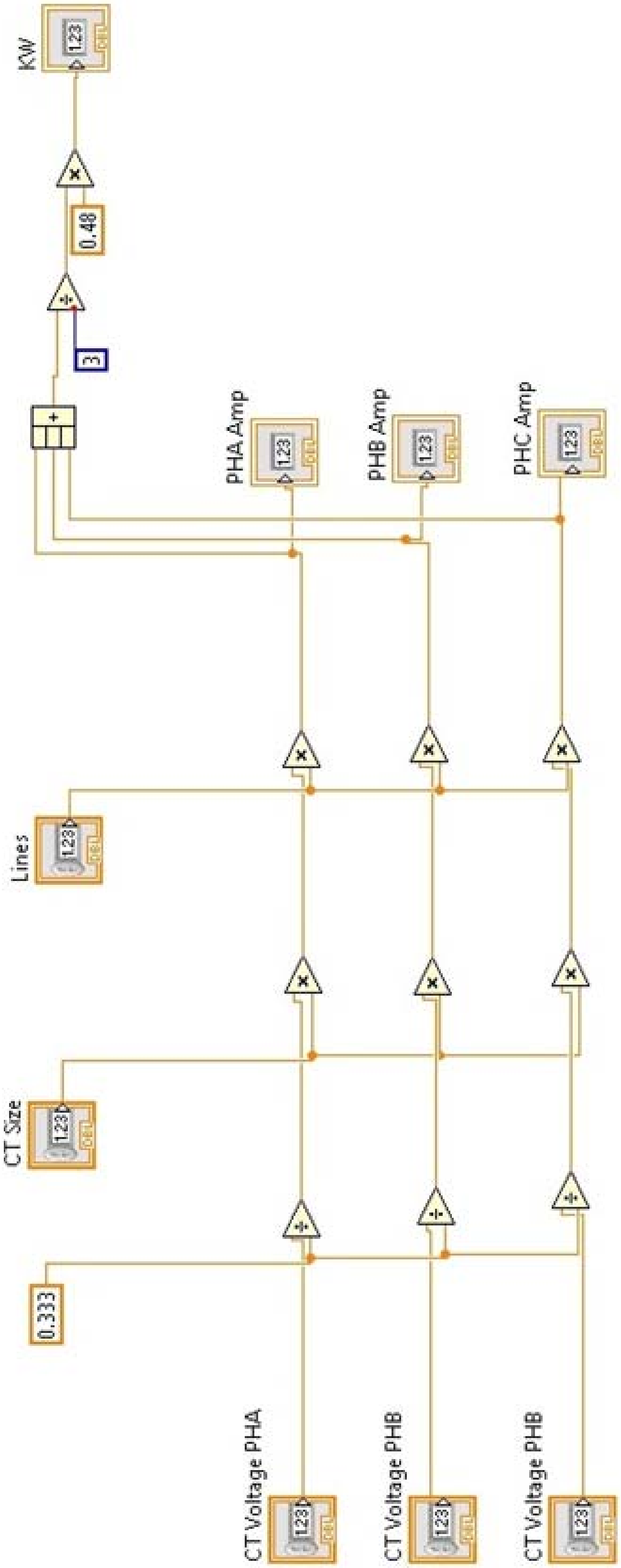
APPENDIX A

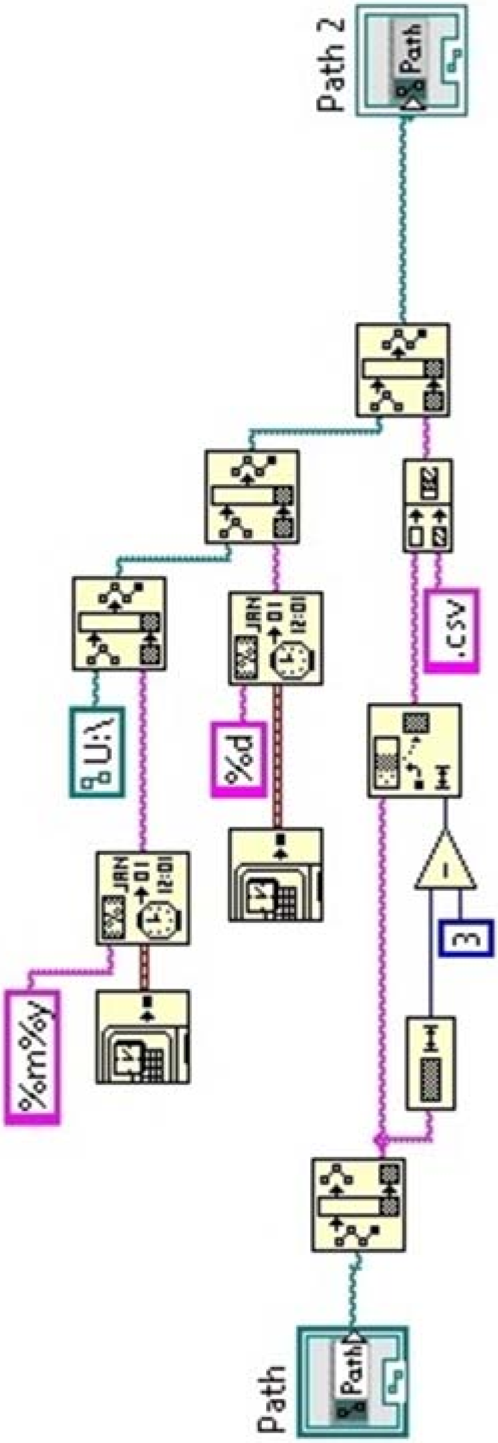

Error
Expected End [i-1]
Actual End [i-1]
Finished Late? [i-1]
Global End Time
Iteration Duration
Processor
Date
Time
KW
PHA
PHB
PHC
KW
PHA
PHB
PHC
KW
PHA
PHB
PHC
Error
60
1000

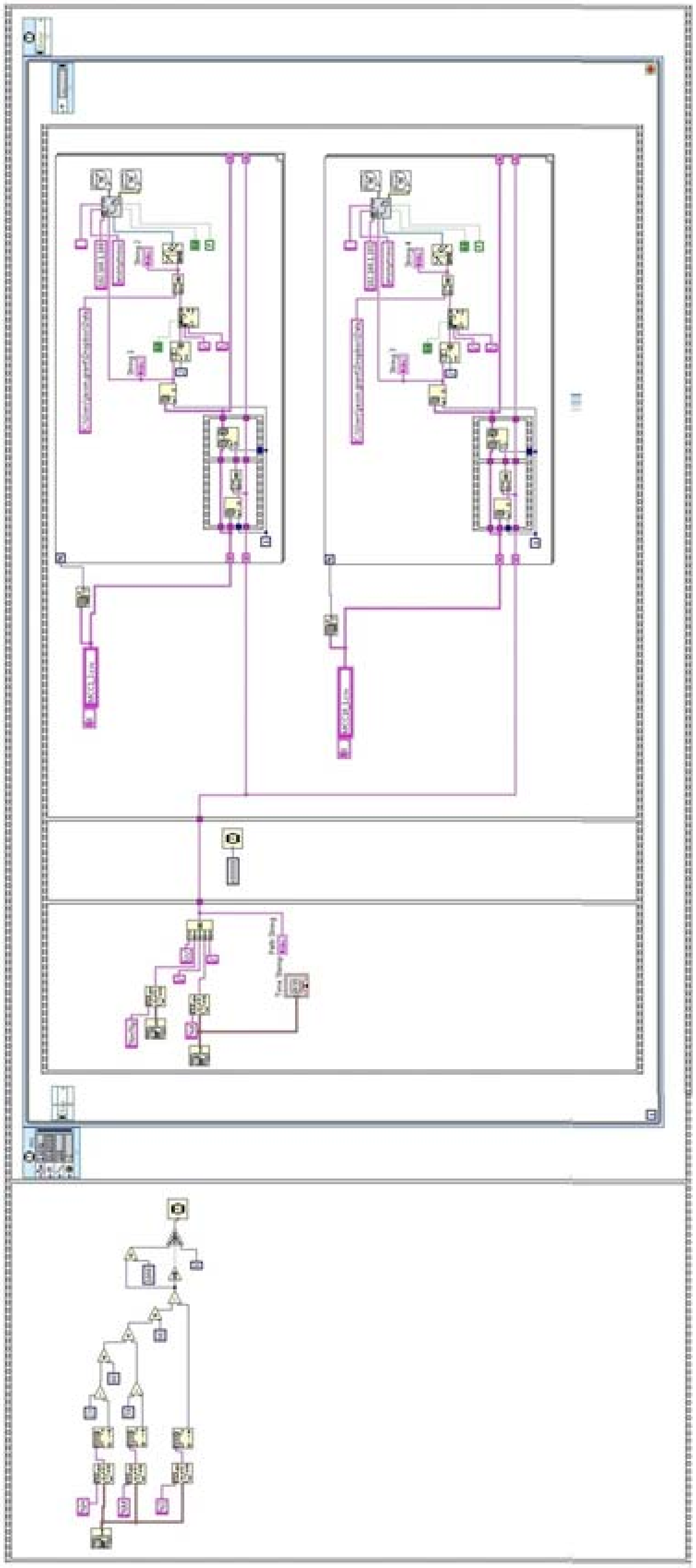

Date
Time
DBL
86400
5
24
Numeric x
13
7
5
6
7
DBL
DBL
Numeric
10000000
Numeric 2
Numeric 3
86400
Numeric 4
10
4
5
DBL
86400
Numeric 5
XL Time
Numeric 6

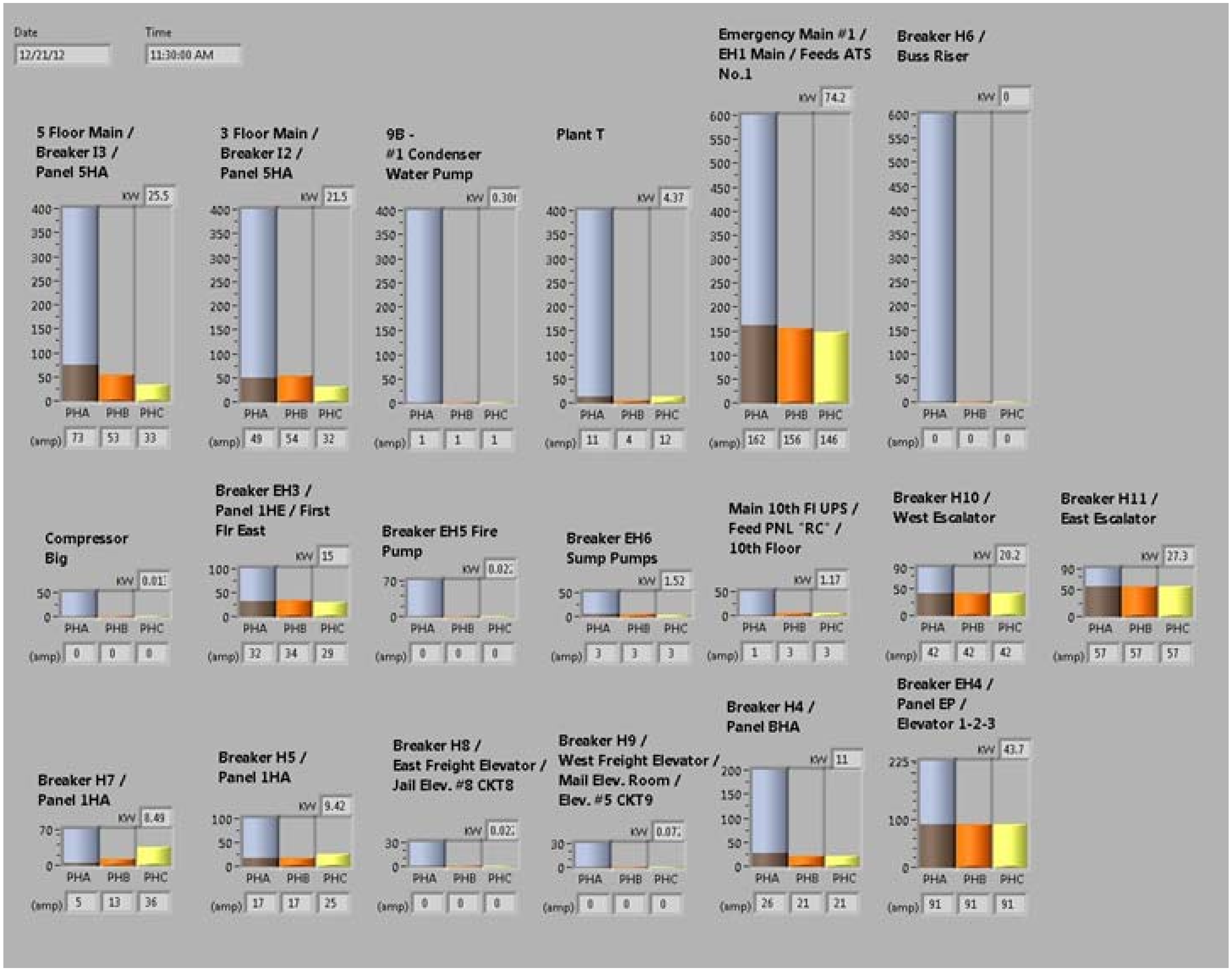

Write To Measurement File
Signals
KW
PHA Amp
PHB Amp
PHC Amp
XL Time
123
2
600
Error
AI18
AI19
AI20
99
scans
Synchro
0
100


PHA signal
DBL
RMS
RMS SinoPHA
DBL
PHB signal
DBL
RMS
RMS Sino PHB
DBL
PHC signal
DBL
RMS
RMS Sino PHC
DBL


KW
0.48
3
PHA Amp
PHB Amp
PHC Amp
Lines
CT Size
0.333
CT Voltage PHA
CT Voltage PHB
CT Voltage PHB


Path 2
Path
%m%y
U:\
%d
.csv
3
JAN
01
12:01





Date 12/21/12

Time 11:30:00 AM

5 Floor Main / Breaker I3 / Panel 5HA — KW 25.5 — (amp) PHA 73, PHB 53, PHC 33

3 Floor Main / Breaker I2 / Panel 5HA — KW 21.5 — (amp) PHA 49, PHB 54, PHC 32

9B - #1 Condenser Water Pump — KW 0.30 — (amp) PHA 1, PHB 1, PHC 1

Plant T — KW 4.37 — (amp) PHA 11, PHB 4, PHC 12

Emergency Main #1 / EH1 Main / Feeds ATS No.1 — KW 74.2 — (amp) PHA 162, PHB 156, PHC 146

Breaker H6 / Buss Riser — KW 0 — (amp) PHA 0, PHB 0, PHC 0

Compressor Big — KW 0.01 — (amp) PHA 0, PHB 0, PHC 0

Breaker EH3 / Panel 1HE / First Flr East — KW 15 — (amp) PHA 32, PHB 34, PHC 29

Breaker EH5 Fire Pump — KW 0.02 — (amp) PHA 0, PHB 0, PHC 0

Breaker EH6 Sump Pumps — KW 1.52 — (amp) PHA 3, PHB 3, PHC 3

Main 10th Fl UPS / Feed PNL "RC" / 10th Floor — KW 1.17 — (amp) PHA 1, PHB 3, PHC 3

Breaker H10 / West Escalator — KW 20.2 — (amp) PHA 42, PHB 42, PHC 42

Breaker H11 / East Escalator — KW 27.3 — (amp) PHA 57, PHB 57, PHC 57

Breaker H7 / Panel 1HA — KW 8.49 — (amp) PHA 5, PHB 13, PHC 36

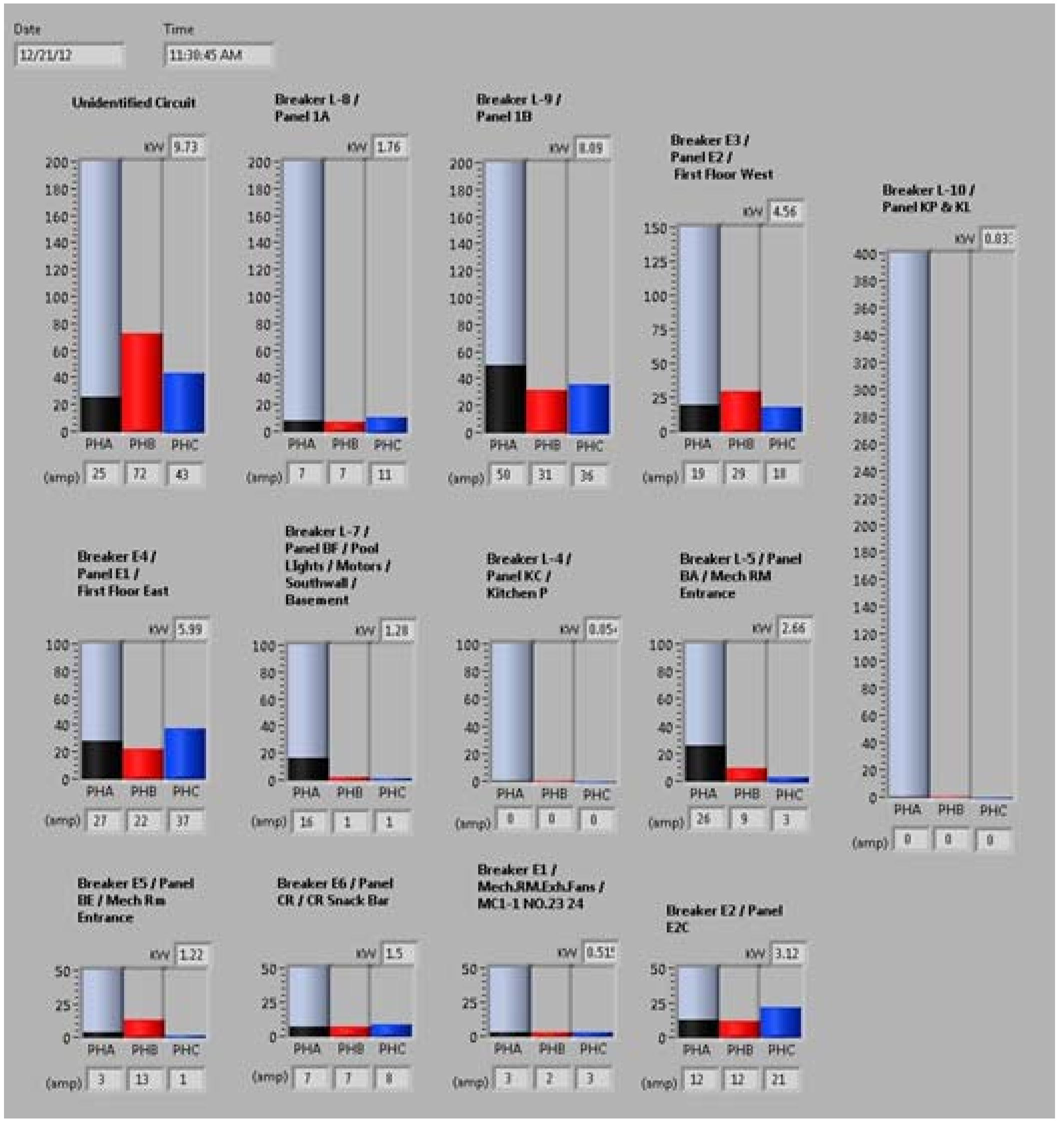
Breaker H5 / Panel 1HA — KW 9.42 — (amp) PHA 17, PHB 17, PHC 25

Breaker H8 / East Freight Elevator / Jail Elev. #8 CKT8 — KW 0.02 — (amp) PHA 0, PHB 0, PHC 0

Breaker H9 / West Freight Elevator / Mail Elev. Room / Elev. #5 CKT9 — KW 0.07 — (amp) PHA 0, PHB 0, PHC 0

Breaker H4 / Panel BHA — KW 11 — (amp) PHA 26, PHB 21, PHC 21

Breaker EH4 / Panel EP / Elevator 1-2-3 — KW 43.7 — (amp) PHA 91, PHB 91, PHC 91



Date 12/21/12

Time 11:30:45 AM

Unidentified Circuit

KW 9.73

(amp) PHA 25 PHB 72 PHC 43

Breaker L-8 / Panel 1A

KW 1.76

(amp) PHA 7 PHB 7 PHC 11

Breaker L-9 / Panel 1B

KW 8.09

(amp) PHA 50 PHB 31 PHC 36

Breaker E3 / Panel E2 / First Floor West

KW 4.56

(amp) PHA 19 PHB 29 PHC 18

Breaker L-10 / Panel KP & KL

KW 0.03

(amp) PHA 0 PHB 0 PHC 0

Breaker E4 / Panel E1 / First Floor East

KW 5.99

(amp) PHA 27 PHB 22 PHC 37

Breaker L-7 / Panel BF / Pool Lights / Motors / Southwall / Basement

KW 1.28

(amp) PHA 16 PHB 1 PHC 1

Breaker L-4 / Panel KC / Kitchen P

KW 0.05

(amp) PHA 0 PHB 0 PHC 0

Breaker L-5 / Panel BA / Mech RM Entrance

KW 2.66

(amp) PHA 26 PHB 9 PHC 3

Breaker E5 / Panel BE / Mech Rm Entrance

KW 1.22

(amp) PHA 3 PHB 13 PHC 1

Breaker E6 / Panel CR / CR Snack Bar

KW 1.5

(amp) PHA 7 PHB 7 PHC 8

Breaker E1 / Mech.RM.Exh.Fans / MC1-1 NO.23 24

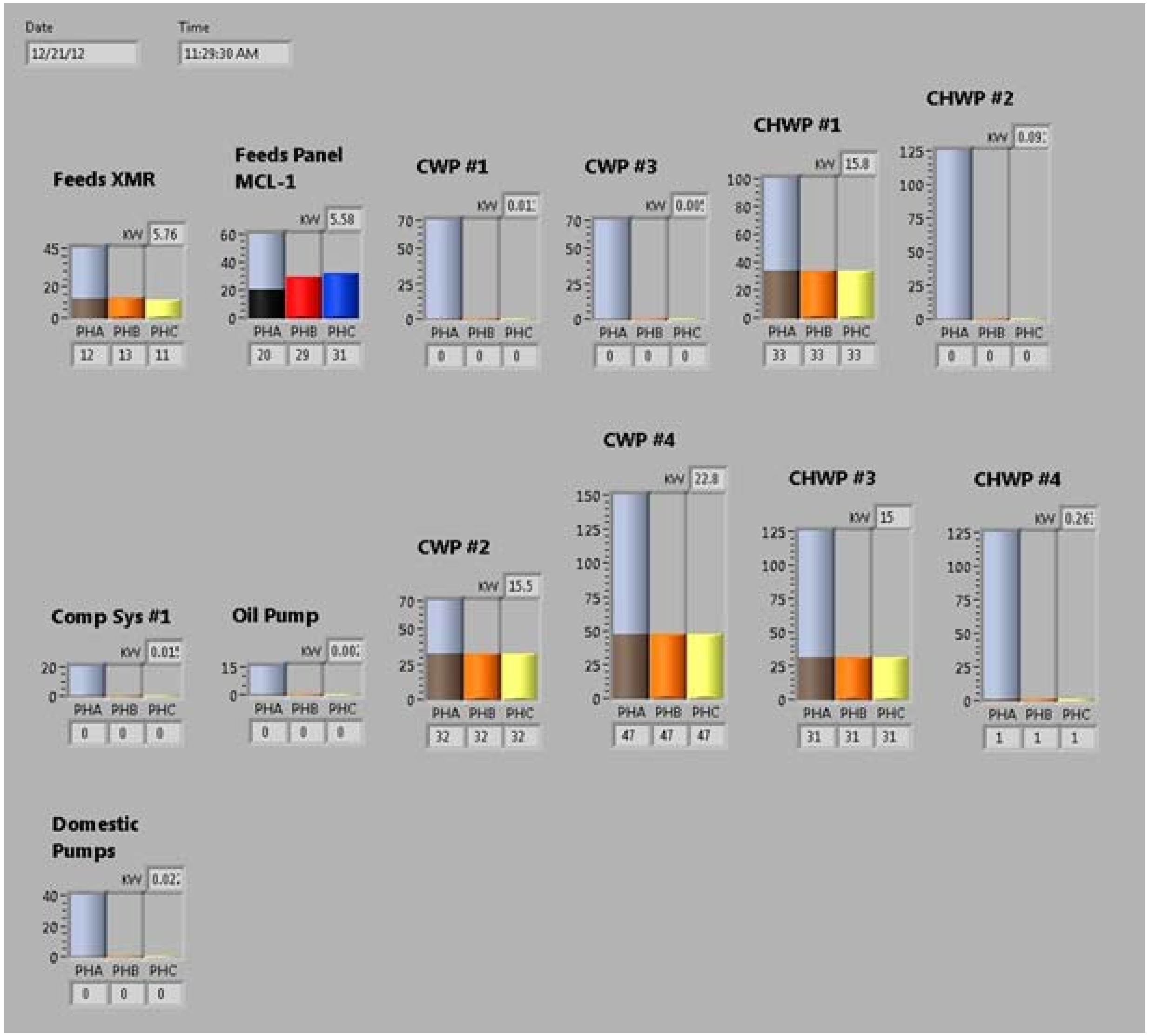
KW 0.51

(amp) PHA 3 PHB 2 PHC 3

Breaker E2 / Panel E2C

KW 3.12

(amp) PHA 12 PHB 12 PHC 21



Date 12/21/12

Time 11:29:30 AM

Feeds XMR
KW 5.76
PHA 12 | PHB 13 | PHC 11

Feeds Panel MCL-1
KW 5.58
PHA 20 | PHB 29 | PHC 31

CWP #1
KW 0.01
PHA 0 | PHB 0 | PHC 0

CWP #3
KW 0.00
PHA 0 | PHB 0 | PHC 0

CHWP #1
KW 15.8
PHA 33 | PHB 33 | PHC 33

CHWP #2
KW 0.09
PHA 0 | PHB 0 | PHC 0

Comp Sys #1
KW 0.01
PHA 0 | PHB 0 | PHC 0

Oil Pump
KW 0.00
PHA 0 | PHB 0 | PHC 0

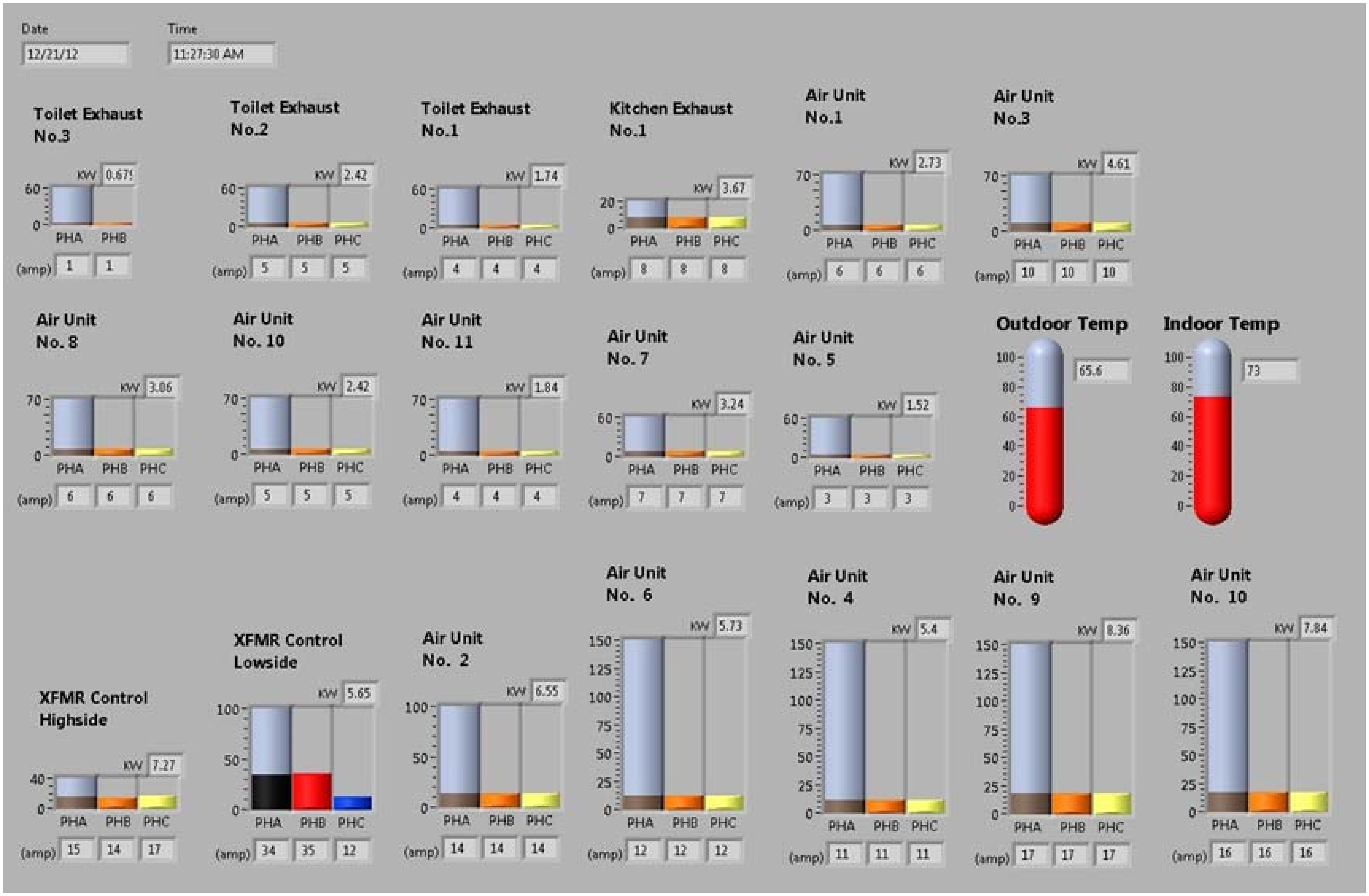
CWP #2
KW 15.5
PHA 32 | PHB 32 | PHC 32

CWP #4
KW 22.8
PHA 47 | PHB 47 | PHC 47

CHWP #3
KW 15
PHA 31 | PHB 31 | PHC 31

CHWP #4
KW 0.26
PHA 1 | PHB 1 | PHC 1

Domestic Pumps
KW 0.02
PHA 0 | PHB 0 | PHC 0



Date 12/21/12

Time 11:27:30 AM

Toilet Exhaust No.3
KW 0.67
PHA PHB
(amp) 1 1

Toilet Exhaust No.2
KW 2.42
PHA PHB PHC
(amp) 5 5 5

Toilet Exhaust No.1
KW 1.74
PHA PHB PHC
(amp) 4 4 4

Kitchen Exhaust No.1
KW 3.67
PHA PHB PHC
(amp) 8 8 8

Air Unit No.1
KW 2.73
PHA PHB PHC
(amp) 6 6 6

Air Unit No.3
KW 4.61
PHA PHB PHC
(amp) 10 10 10

Air Unit No. 8
KW 3.06
PHA PHB PHC
(amp) 6 6 6

Air Unit No. 10
KW 2.42
PHA PHB PHC
(amp) 5 5 5

Air Unit No. 11
KW 1.84
PHA PHB PHC
(amp) 4 4 4

Air Unit No. 7
KW 3.24
PHA PHB PHC
(amp) 7 7 7

Air Unit No. 5
KW 1.52
PHA PHB PHC
(amp) 3 3 3

Outdoor Temp 65.6

Indoor Temp 73

XFMR Control Highside
KW 7.27
PHA PHB PHC
(amp) 15 14 17

XFMR Control Lowside
KW 5.65
PHA PHB PHC
(amp) 34 35 12

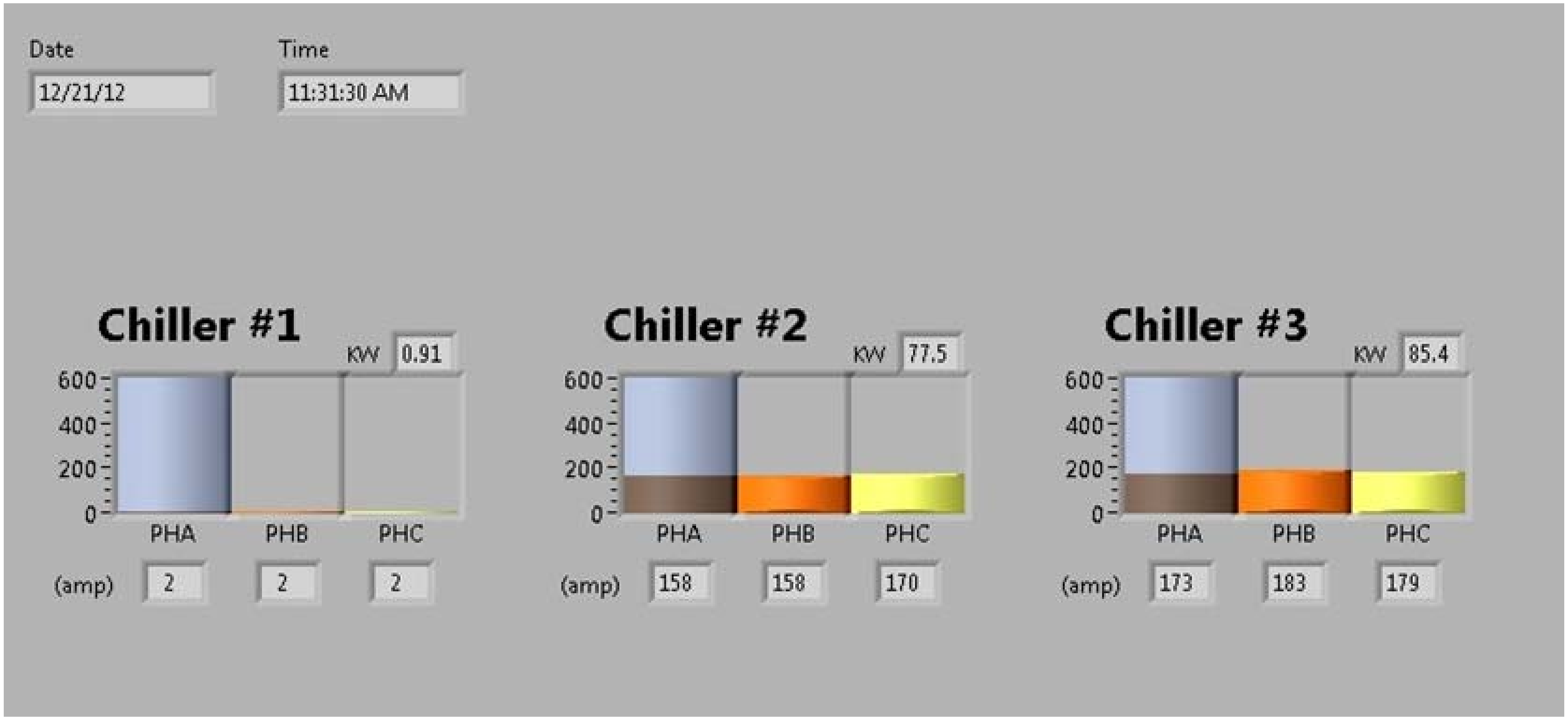
Air Unit No. 2
KW 6.55
PHA PHB PHC
(amp) 14 14 14

Air Unit No. 6
KW 5.73
PHA PHB PHC
(amp) 12 12 12

Air Unit No. 4
KW 5.4
PHA PHB PHC
(amp) 11 11 11

Air Unit No. 9
KW 8.36
PHA PHB PHC
(amp) 17 17 17

Air Unit No. 10
KW 7.84
PHA PHB PHC
(amp) 16 16 16



Date 12/21/12

Time 11:31:30 AM

Chiller #1

KW 0.91

	PHA	PHB	PHC
(amp)	2	2	2

Chiller #2

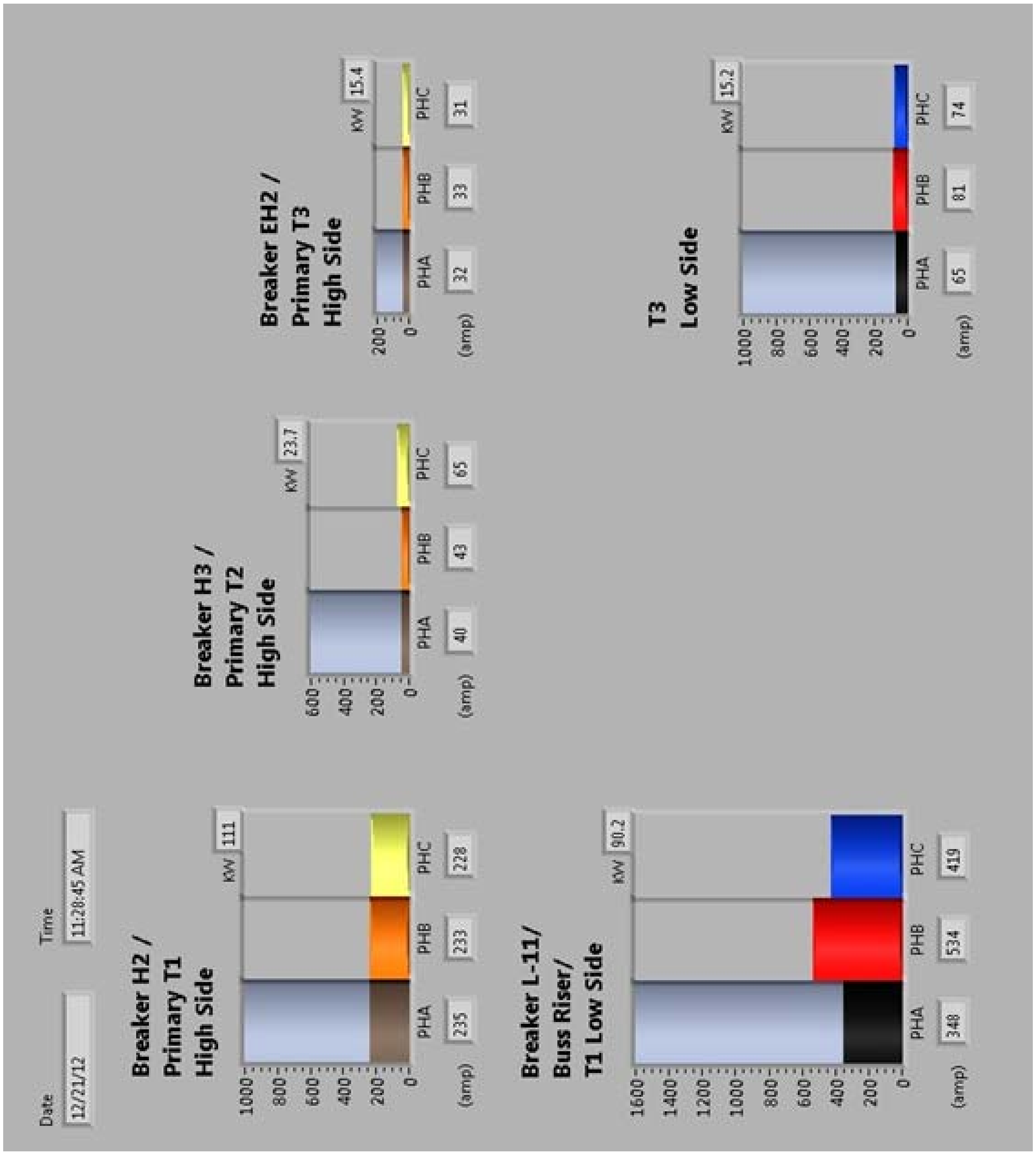
KW 77.5

	PHA	PHB	PHC
(amp)	158	158	170

Chiller #3

KW 85.4

	PHA	PHB	PHC
(amp)	173	183	179



Date 12/21/12

Time 11:28:45 AM

Breaker H2 / Primary T1 High Side

PHA 235 | PHB 233 | PHC 228 | KW 111

Breaker L-11/ Buss Riser/ T1 Low Side

PHA 348 | PHB 534 | PHC 419 | KW 90.2

Breaker H3 / Primary T2 High Side

PHA 40 | PHB 43 | PHC 65 | KW 23.7

Breaker EH2 / Primary T3 High Side

PHA 32 | PHB 33 | PHC 31 | KW 15.4

T3 Low Side

PHA 65 | PHB 81 | PHC 74 | KW 15.2

APPENDIX B

```
Sub DC()

    Dim strFilename As String
    Dim strSaveAs As String
    Dim strPath As String
    Dim wbkTemp As Workbook
    Dim strFull As String

    Dim i As Integer
    i = 1
    Dim j As Integer
    j = 1
    Dim k As Integer
    k = 0
    Dim l As Integer
    l = 1
    Dim pos As Integer
    pos = 2
    Dim total As Integer
    total = 0
    Dim count As Integer
    count = 0

   Application.ScreenUpdating = False

    strPath = "C:\Users\JLG\Desktop\ExcelTest\20\"
    strFilename = Dir(strPath & "*.csv")

    Do While Len(strFilename) > 0
        Set wbkTemp = Workbooks.Open(strPath & strFilename)
        i = 1
        total = 0
        pos = 2
        count = 0
        If strFilename = "TEMPS.csv" Then
            wbkTemp.Sheets(1).Columns(1).Insert
            wbkTemp.Sheets(1).Columns(1).Insert
        ElseIf strFilename = "MCC10_1.csv" Then
            wbkTemp.Sheets(1).Columns(1).Insert
        End If

        Do While wbkTemp.Sheets(1).Range("F1").Offset(i - 1, 0) > 0
            total = total + 1
            i = i + 1
        Loop
    For j = 1 To total - 1
      wbkTemp.Sheets(1).Range("G1").Offset(j, 0).Value =
wbkTemp.Sheets(1).Range("F1").Offset(j, 0).Value -
wbkTemp.Sheets(1).Range("F1").Offset(j - 1, 0).Value
    Next j


    For j = 1 To total
      k = wbkTemp.Sheets(1).Range("G1").Offset(pos - 1, 0).Value / 0.000173
      If k > 1.9 Then
        For l = 1 To k - 1
          wbkTemp.Sheets(1).Rows(pos + l - 1).EntireRow.Insert
        Next l
        wbkTemp.Sheets(1).Rows(pos - 1).Copy
        wbkTemp.Sheets(1).Rows(pos & ":" & pos + k - 2).PasteSpecial
        count = 0
        For l = 1 To k
          wbkTemp.Sheets(1).Range("F1").Offset(pos + l - 2).Value =
wbkTemp.Sheets(1).Range("F1").Offset(pos + l - 3) + 0.0001736
          count = count + 1
        Next l
        pos = pos + count
        wbkTemp.Sheets(1).Range("G1").Offset(pos - 2).Value =
wbkTemp.Sheets(1).Range("F1").Offset(pos - 2).Value -
wbkTemp.Sheets(1).Range("F1").Offset(pos - 3).Value
        Else
          pos = pos + 1
        End If
    Next j

    If InStr(1, wbkTemp.Sheets(1).Range("F1").Value, ".") Then
      wbkTemp.Sheets(1).Rows(1).Copy
      wbkTemp.Sheets(1).Range("A1").EntireRow.Insert
      wbkTemp.Sheets(1).Range("F1").Value = Int(wbkTemp.Sheets(1).Range("F1"))
    End If

    If pos = 5760 Then
      wbkTemp.Sheets(1).Rows(5759).Copy
      wbkTemp.Sheets(1).Rows(5760).EntireRow.Insert
      wbkTemp.Sheets(1).Range("F5760").Value =
wbkTemp.Sheets(1).Range("F5759").Value + 0.0001736
    End If

    wbkTemp.Sheets(1).Columns(1).Delete
    wbkTemp.Sheets(1).Columns(6).Delete
```

```
        wbkTemp.Date1904 = True
        wbkTemp.Sheets(1).Columns(5).NumberFormat = "mm/dd/yyyy hh:mm:ss AM/PM"

        If strFilename = "TEMPS.csv" Then
            wbkTemp.Sheets(1).Columns(1).Delete
            wbkTemp.Sheets(1).Columns(1).Delete
        ElseIf strFilename = "MCC10_1.csv" Then
            wbkTemp.Sheets(1).Columns(1).Delete
        End If

        strFull = strPath & strFilename
        strFull = Replace(strFull, ".csv", ".xlsx")
        wbkTemp.SaveAs Filename:=strFull, FileFormat:=51
        strFilename = Dir
        wbkTemp.Close True

    Loop

End Sub
```

```
Sub x()

   Dim strFilename As String
   Dim strSaveAs As String
   Dim strPath As String
   Dim wbkTemp As Workbook


   Dim i As Integer
   i = 4
   Dim j As Integer
   j = 1
   Dim k As Integer
   k = 4

   Application.ScreenUpdating = False

   Set NewBook = Workbooks.Add
   With NewBook
      .Title = "ALL DAY DATA"
      .Subject = "Data"
      .SaveAs
Filename:="C:\Users\JLG\Desktop\ExcelTest\all\ALLDAYDATA_20.xlsx"
   End With


   strPath = "C:\Users\JLG\Desktop\ExcelTest\20\"
   strFilename = Dir(strPath & "*.xlsx")

   Do While Len(strFilename) > 0
      Set wbkTemp = Workbooks.Open(strPath & strFilename)
         If strFilename = "MCC10_1.xlsx" Then
            wbkTemp.Sheets(1).Columns(4).Copy
            NewBook.Sheets(1).Range("A1").PasteSpecial
            wbkTemp.Sheets(1).Columns("A:C").Copy
            NewBook.Sheets(1).Range("B1").PasteSpecial
            NewBook.Sheets(2).Range("A1").Value = Replace(strFilename, ".xlsx", "")
         ElseIf strFilename = "TEMPS.xlsx" Then
            wbkTemp.Sheets(1).Columns("A:B").Copy
            NewBook.Sheets(1).Range("A1").Offset(0, i).PasteSpecial
            NewBook.Sheets(2).Range("A1").Offset(0, j).Value = Replace(strFilename,
".xlsx", "")
         Else
            wbkTemp.Sheets(1).Columns("A:D").Copy
            NewBook.Sheets(1).Range("A1").Offset(0, i).PasteSpecial
```

```
        NewBook.Sheets(2).Range("A1").Offset(0, j).Value = Replace(strFilename,
".xlsx", "")
      End If
      j = j + 1
      i = (j * 4) - 3
      Application.CutCopyMode = False
      wbkTemp.Close True
      strFilename = Dir

  Loop

    NewBook.Sheets(1).Columns(5).Delete
    NewBook.Sheets(2).Columns(2).Delete
    NewBook.Sheets(1).Rows(1).Insert
    NewBook.Sheets(1).Range("A1").Value = "Time"
    NewBook.Sheets(1).Range("B1").Value = NewBook.Sheets(2).Range("A1").Value
& " KW"
    NewBook.Sheets(1).Range("C1").Value = NewBook.Sheets(2).Range("A1").Value
& " PHA"
    NewBook.Sheets(1).Range("D1").Value = NewBook.Sheets(2).Range("A1").Value
& " PHB"


    For i = 2 To j - 2
      NewBook.Sheets(1).Range("A1").Offset(0, k).Value =
NewBook.Sheets(2).Range("A1").Offset(0, i - 1).Value & " KW"
      k = k + 1
      NewBook.Sheets(1).Range("A1").Offset(0, k).Value =
NewBook.Sheets(2).Range("A1").Offset(0, i - 1).Value & " PHA"
      k = k + 1
      NewBook.Sheets(1).Range("A1").Offset(0, k).Value =
NewBook.Sheets(2).Range("A1").Offset(0, i - 1).Value & " PHB"
      k = k + 1
      NewBook.Sheets(1).Range("A1").Offset(0, k).Value =
NewBook.Sheets(2).Range("A1").Offset(0, i - 1).Value & " PHC"
      k = k + 1
    Next

    NewBook.Sheets(1).Range("A1").Offset(0, k).Value =
NewBook.Sheets(2).Range("A1").Offset(0, i - 1).Value & " Exterior"
    k = k + 1
    NewBook.Sheets(1).Range("A1").Offset(0, k).Value =
NewBook.Sheets(2).Range("A1").Offset(0, i - 1).Value & " Interior"
    k = k + 1
    For i = 1 To 306
      NewBook.Sheets(1).Columns(i).AutoFit
    Next i


    NewBook.Sheets(1).Columns(1).NumberFormat = "mm/dd/yyyy hh:mm:ss
AM/PM"
    NewBook.Date1904 = True
    NewBook.Save
    NewBook.Close


End Sub
```