ARIZONA STATE UNIVERSITY
POWER SYSTEMS ENGINEERING RESEARCH CENTER

SYMBOLIC DYNAMIC MODELS FOR HIGHLY VARYING POWER SYSTEM LOADS

Diwakar Tewari

February, 2002

ABSTRACT

Representation and modeling of loads in a power system are very important, as a system may have different types of complex loads. The work reported here is aimed at loads which are highly varying in nature, such as electric arc furnaces and steel rolling mills. These loads do not have fully accepted physical models because of the unpredictable nature of load current, but they often have a rich set of operating data over a wide range of operation.

The application of Symbolic Dynamics in load modeling, as presented in this thesis, is a novel concept. The scope of this concept is twofold. One aspect is the utilization of time series historical data of voltages and currents to formulate the model. The model might be used in routine power engineering applications such as power flow studies, transient stability studies and short circuit studies. The second aspect is the utilization of historical voltage and current data to predict future load current values. The predictions could then be used for power conditioning. An example of the power conditioning application is active filtering of unwanted components of load current.

Symbolic Dynamics is based on the coding of signals as ‘symbols’, which appear in groups known as ‘words’. Thus, bus voltage and load current are descretized to be represented by symbols. The states of these time varying signals can be represented by words. A ‘symbolic dynamic dictionary’ is formed from historical time series data. The words and the symbols themselves are treated much as they might in a natural language.

Symbolic dynamic modeling has been applied to both synthetic and alternating current (AC) electric arc furnace data. The tests have been used to compare the symbolic dynamic model response to the actual measurements. Comparisons presented herein

include statistical tests, electric power quality indices, and a new proposed index to compare nonsinusoidal signals.

TABLE OF CONTENTS

		Page
LIST OF TABLES	. . . .	ix
LIST OF FIGURES	. . . .	xi
NOMENCLATURE	. . . .	xii

CHAPTER

1 INTRODUCTION

1.1	Motivation	1
1.2	Scope	1
1.3	Power system load modeling	2
1.4	Literature search on power system load modeling .	3
1.5	Power conditioning	7
1.6	Literature search on power conditioning . .	8
1.7	Symbolic Dynamics	9
1.8	Literature search on Symbolic Dynamics . .	10
1.9	Markov chain model	12
1.10	Literature search on Markov chain model . .	12

2 SYMBOLIC DYNAMICS

2.1	Introduction	14
2.2	Some definitions	14
2.3	Coding a signal in Symbolic Dynamics . .	16
2.4	Dictionary formation	17
2.5	Comparison of two signals	18

CHAPTER			Page
	2.6	Forecasting a signal	19
	2.7	Development of a model based on multiple symbol checking (Model 2)	21
3	SYNTHETIC TESTS AND RESULTS		
	3.1	Introduction	24
	3.2	Nomenclature	24
	3.3	Test condition	25
	3.4	Analysis tools	25
	3.5	Test results	29
4	TESTS ON INDUSTRIAL ARC FURNACE DATA		
	4.1	Introduction	39
	4.2	Data transformation	40
	4.3	Test condition	41
	4.4	Test results	42
	4.5	Discussion	53
5	CONCLUSIONS AND FUTURE WORK		
	5.1	Conclusions	56
	5.2	Recommendations	58
REFERENCES			60
APPENDIX			
A	MATLAB CODE FOR THE PROPOSED WORK		
	A.1	Dictionary formation	65
APPENDIX			Page

	A.2	Comparing two dictionaries	67
	A.3	Forecasting a signal using Model 1 . . .	73
	A.4	Forecasting a signal using Model 2, case 1 . .	77
	A.5	Forecasting a signal using Model 2, case 2 . .	84
B	ILLUSTRATIVE EXAMPLE OF KOLMOGOROV-SMIRNOV TESTS		
	B.1	An illustrative example of the KS test . .	94
	B.2	Solution of the example	94
C	EAF AND REAL DATA RECORDING		
	C.1	Real data used in the proposed work . . .	99

LIST OF TABLES

Table		Page
3.1	Test statistics for categories A and B	26
3.2	Test statistics for categories C,D and E	26
3.3	Variations of *CSI* with phase angle	30
3.4	Variations of *CSI* with S/N ratio, Test *ST-B03*	32
3.5	Results for category C tests	33
3.6	Results for category D, Tests *ST-D01*, *ST-D02*, and *ST-D03*	35
3.7	Results for category E, Tests *ST-E01*, *ST-E02*, and *ST-E03*	37
4.1	EAF specification	39
4.2	Test statistics for EAF tests	43
4.3	Results for *RT-D01*	43
4.4	Results for *RT-D02*	44
4.5	Results for *RT-D03*	45
4.6	Results for *RT-D04*	45
4.7	Error in the average power for test *RT-D05*	47
4.8	Results for *RT-E01*	48
4.9	Results for *RT-E02*	49
4.10	Results for *RT-E03*	50
4.11	Execution time comparison	55
5.1	Error in the proposed model when tested for EAF data	57
B.1	Data sets under consideration	94

Table		Page
B.2	Calculation of functions for KS test	95
B.3	Value of α for the KS test (taken from [20]) . . .	97
C.1	Sample data sheet for 3-phase EAF current and voltage . .	102

LIST OF FIGURES

Figure		Page
2.1	Representation of a signal using the concept of Symbolic Dynamics	16
2.2	Forming a Symbolic Dynamic dictionary	17
2.3	Comparing two signals	18
3.1	Synthetic signals for *ST-B01*	30
3.2	Synthetic circuit to generate current data	34
3.3	Summary of test results for Model 1 and Model 2 . . .	38
4.1	Transforming the real data	40
4.2	RMS value comparison with historical data in tests *RT-D02, RT-D03* and *RT-D04*	46
4.3	RMS value comparison with actual data in tests *RT-D02, RT-D03* and *RT-D04*	46
4.4	Process of power conditioning for the test *RT-E04* . . .	51
4.5	Comparison of actual and conditioned current waveshape for phase A	52
4.6	Comparison of actual and conditioned current waveshape for phase B .	52
4.7	Comparison of actual and conditioned current waveshape for phase C .	53
5.1	Pictorial representation of alternative approaches to power system load modeling	59
C.1	A sample of EAF current for phase A	99
C.2	A sample of EAF current for phase B	100
C.3	A sample of EAF current for phase C	100
C.4	Electric arc furnace schematic	101

NOMENCLATURE

α	Significance level
c	A dictionary for signal χ
γ	A dictionary for signal γ
CDSM	Composite dynamic static model
CNLRF	Constrained nonlinear recursive filter
CSI	Common signal index
CW_F	Cumulative fractional occurrence in working dictionary
D	Symbolic dynamical dictionary
DSAP	Distribution system aggregation program
EAF	Electric arc furnace
EPRI	Electric power research institute
f	Fractional occurrence associated with the dictionary
FACTS	Flexible AC transmission systems
FFT	Fast Fourier transform
fix	Matlab function to round off a number
F_m	Empirical distribution function for the population having m number of elements
f_{xi}	Fractional occurrence of i^{th} word in X dictionary
GA	Genetic algorithm
GABPE	Genetic algorithm based parameter estimation

G_n	Empirical distribution function for the population having n number of elements
H_0	Null hypothesis: $F(t)=G(t)$ for all t
H_1	Hypothesis: $F(t) \neq G(t)$ for at least one t
IM	Induction motor
IO	Input output
I_{pred}	Predicted value of current
IREQ	Hydro Quebec Institute of Research
J	Two-sided, two-sample Kolmogorov Smirnov statistics
KS test	*Kolmogorov-Smirnov* test
M_D	Mini dictionary, derived from the symbolic dynamic dictionary D
M_F	Fractional occurrence associated with mini dictionary
N	Number of cells used to descretize a signal
NLDM	Nonlinear dynamic models
N_w	Maximum word length in a dictionary
PRS	Pseudo random sequences
PSCAD	Power system computer aided design.
$Q(s)$	Significance function in KS test
q^*_a	Parameter in the KS significance function
rand	Matlab function for generating a random number
RT	Real test, performed on actual EAF data
S/N	Signal to noise ratio

ST	Synthetic test, performed on computer generated data
SVC	Static var compensator
UPFC	Unified power flow control
UPS	Uninterruptible power supplies
UTA	University of Texas at Arlington
V_{pred}	Predicted value of voltage
VSCS	Variable structure control systems
W_D	Working dictionary
W_F	Fractional occurrence associated with working dictionary
X	Width of each cell
XX-Ynn	*XX* type of test, numbered *nn* belonging to the category *Y*
Z	Ordered values for combined populations in KS test

CHAPTER 1

INTRODUCTION

1.1 Motivation

Representation and modeling of loads in a power system is very important, as a system may have different types of complex loads. Some loads are highly varying in nature like electric arc furnaces, steel rolling mills and other electric loads involved in different processes in the steel industry. These loads still do not have fully accepted physical models because of the unpredictable nature of the load current, but they have a rich set of data over a wide range of operation.

The motivation of the proposed approach is to develop a load model for highly varying loads by using time series measurements of voltage and current only. The proposed model can also be used to study distortion in non-periodic and poorly behaved signals. This work introduces the concept of Symbolic Dynamics to accomplish the objectives mentioned above. Symbolic Dynamics is a mathematical approach in which signals are discretized and represented as sequence of symbols. Another motivation of the work reported here is to use a very innovative approach for load modeling, and again, Symbolic Dynamics fits this description.

1.2 Scope

The scope of this project is twofold. One element is the utilization of time series historical data of voltages and currents of a load or industrial process in order to formulate a model *M*. The model *M* might be used in routine power engineering applications such as power flow studies, transient stability studies and short circuit studies. These are general applications of load models.

The second element in the scope is to use historical voltage and current data to predict future load current values. These predictions could be used for power conditioning. An example of this application is active filtering of unwanted components of load currents.

In order to accomplish the twofold objectives indicated, a relatively novel approach is proposed. This is the application of Symbolic Dynamics.

1.3 Power system load modeling

As far as the stable operation of a power system is concerned, it is important to match the electric load on the system to the electric output of the generators. For this, the representation of the load is an important issue, because in a large power system, the system load consists of a large number of complex loads such as motors, heaters, incandescent and fluorescent lamps, refrigerators, furnaces, and compressors. It is also difficult to estimate the exact composition of the load as the composition varies depending on the time, weather conditions and many other complex factors. The concept of generation-load balance in the steady state may be taken to be a stability issue. If the generation-load balance is nonzero, excess or deficient energy will migrate to the system rotating elements, and loss of stability may occur. Power system load modeling is an important factor from the 'stability analysis' point of view.

Traditionally load models can be classified into two broad categories:

Static load model: Most of the loads in a power system respond very rapidly to the change in voltage and frequency. The steady state of the load response is achieved quickly. Such cases can be analyzed by using static models. In a static model, the active power component P and the reactive power component Q are the functions of the bus voltage magnitude and frequency.

Dynamic load model: There are many loads which cannot be analyzed in the above manner. In such cases, it is important to account for the dynamics of the load component. In a dynamic load model, the active and reactive power components at any instant of time are functions of the bus voltage magnitude at past and present instants of time.

Power engineers need various power system mathematical models for studying the performance and control of the power system. Conventional power system modeling demands parameters and characteristics of system components. In many cases, what exists in the field may be uncertain because of system complexity, age and other inaccuracies.

Power engineers can circumvent this uncertainty through the use of measurements to formulate a mathematical model that is consistent with the measurements. The process of power system modeling involves physical analysis, mathematical deductions, modeling induction and the utilization of measurements.

1.4 Literature search on power system load modeling

Many papers have been published on load modeling using different techniques. The literature of this area spans about 100 years. Recent efforts focus on accuracy, highly varying loads, and power quality. An EPRI report [1], published in 1979, is basically about load modeling. This report describes the research project *RP849*, which involves three major contractors, Hydro Quebec Institute of Research (IREQ), General Electric (GE), and University of Texas at Arlington (UTA). To form a load model, different load components were tested in the UTA laboratories and field test sites. In these tests, the power variation of each component was recorded under slow and rapid variation of supply voltage and frequency. The load modeling process in this research project utilizes a computer program, “Distribution System Aggregation Program (DSAP)”. DSAP simulates the system load to produce a data set of power responses to input variations (voltage and/or frequency). Using this data set as an input to a curve fitting subroutine, a mathematical formula can be obtained that best matches the data. This derived mathematical relationship is the load model for the given system.

In 1998, Wang and Pahalawaththa published a paper [2] describing the progress in developing a device and a technique for power system load modeling. Their application is based on flexible AC transmission systems (FACTS) technologies. The system identification technique which the authors used entails a correlation method with pseudo random sequences (PRS) for signal injection and load modeling. Simulation studies have been done using PSCAD and MATLAB. For the noise injection, a low power noise source was developed; adopting the concept used in unified power flow control (UPFC) devices. This noise injection technique is capable of producing variations in the system frequency and the voltage as seen by the loads. These frequency and voltage variations are very important for the dynamic modeling of frequency as well as the voltage response of the load as they provide necessary perturbation for voltage and frequency. The cross correlation method, used in the proposed model, is based on the fact that for a linear system with certain input signals the cross correlation of the system input and output will give the impulse response. The simulation results show that the approach is practical and accurate for power system load modeling.

A new modeling technique, based on real data obtained on site in industrial power systems was introduced in 1998 by Wu and Wen. [3] This technique is a combination of genetic algorithm, evolutionary programming and evolution strategies. Different from conventional optimization techniques, genetic algorithm is a population-based algorithm. The population of all possible solution sets is generated in a stochastic manner and the best-found solution set acts as parents for successive generations. This technique, called intelligent learning technique is based on system measurement. The authors proposed an equivalent area load model and to optimize the parameters of this model, an improved genetic algorithm (GA) was developed. Simulation results show that this method is capable of finding a precise model for the load area in a real power system. Moreover it is a practical method as the data required for this can be easily measured in a power system.

Ju, Handschin and Karlsson published a paper on nonlinear dynamic load modeling using GA in 1995 [4]. Many nonlinear dynamic models (NLDM) have been developed in the past. Composite induction motor (IM) models have long been studied and are popular. The effect of voltage phase angle on load power is considered in the IM model but it has not been taken into consideration in the input output (IO) models. The IO models can represent induction motor loads as well as the other dynamic loads. This paper is an effort to include the phase angle into the IO model. The authors developed a composite dynamic static model (CDSM), which includes the effect of voltage angle on transient active power. They used GA for the estimation of parameters. CDSM can be applied to both angle stability and the voltage stability. The GA based parameter estimation (GABPE) approach has been successfully applied to induction motor models, input output models, neural network models, and results show that this approach is simple but powerful.

A new approach was proposed in 1989 by Ma and Ju for dynamic load modeling [5]. They used a constrained nonlinear recursive filter (CNLRF) for parameter estimation, which was based on the data obtained from field tests. A power system may consist of different types of complex loads with different characteristics. To aggregate these different kind of loads and to represent them in the form of equivalent models is important as far as the modeling and analysis is concerned. Induction machines are major loads in any power system in most cases. There are two approaches for the aggregation of induction motors, one is theoretical aggregation, and the other is identification aggregation. Theoretical aggregation requires the parameters of all individual loads and transmission and distribution lines, which is not possible practically. Identification aggregation methods are based on the least square parameter estimation algorithm and require iterative solution of the state equations. The authors suggested an improved method of identification aggregation for the dynamic load model, which requires only data obtained from field tests. Thus the requirement of individual load parameters and the iterative solutions of state equations can be eliminated.

In 1999-2000, Dinesh at Arizona State University worked on the load modeling [6]. His work also aims at the highly varying loads. He used the concept of Markov chains to model the load. According to this concept, the future of a system is dependent only on the present state of the system. System states are represented by the time series data of current and/or voltage. By using these data, Dinesh tried to regenerate the response of the original system without looking at the original data. Results show that the response of signal and the original signal were fairly in agreement.

1.5 Power conditioning

Power line disturbances in any system are mainly caused by electric noise and by voltage fluctuations, such as transient spikes, surges, sags, and outages. These disturbances may severely affect the operation of microprocessor based electronic equipment. Voltage fluctuations and noisy signals may cause computing and printing errors, improper data transfer, data losses and even damage to sensitive circuitry.

A power conditioner attenuates these harmful disturbances to prevent malfunctioning of sensitive electronic equipment and thus providing reliable operation. Depending on the load type and the type of noise in the supply bus, a power conditioner may consist of a simple passive filter, rectifier-inverter configuration, UPS, isolation transformer, automatic voltage regulator and active filter.

1.6 Literature search on power conditioning

Power conditioning is an important topic in the area of power quality and many papers have been published on this topic.

In 1998 Takeda and Artisuka published a paper on the applications of active filter for power conditioning in distribution networks [7]. In this paper, active filters have been shown as the most effective power conditioner for power quality improvement. Active filter can be used for harmonic current compensation, reactive current compensation, voltage sag compensation, voltage flicker compensation, and negative phase sequence compensation in distribution networks. Performance of an active filter greatly depends on the control parameters. Determining these parameters for respective applications, mentioned above, has been discussed in the paper. Selection of the control parameters, particularly for arc furnace flicker compensation is very important. Two cases of such compensation have been discussed in this paper, one is using active filter alone, and the other is using a combination of an active filter and a static var compensator (SVC). Compensation characteristic of an active filter for arc furnace application is analyzed and a relationship between compensation characteristic and control parameter has been obtained.

An active power filter for non-linear AC loads has been proposed by Nastramn, Cajhen, Seliger, and Jereb [8]. They have shown the experimental use of a filter for different loads. Power unit of the filter consists of a bridge circuit having semiconductor switches and a capacitor across this bridge. Depending on the direction of current flow through the semiconductor switches, the capacitor can be discharged or charged, thereby increasing or decreasing the value of load current. The principle for controlling the current reference in this approach helps in stabilizing the load voltage. Thus the active filter proposed by the authors, not only performs the filtering function for the higher order harmonics but also exhibits the characteristic of a variable inductance, providing additional stabilization.

In 1992, Carpita and Marchesoni conducted an experimental study of a power conditioning system using sliding mode control [9]. Traditional techniques used for the control of power conditioning system are harmonic elimination [10], optimal [11], subharmonic [12], and pulse width modulation methods. These are the open loop methods that require a close loop to maintain the desired value of output voltage. According to the authors, control actions, which make use of RMS output voltage as feedback, are not fully satisfactory because of their long settling times. The sliding mode operation, discussed in this paper, is a particular phenomenon of variable structure control systems (VSCS) theory. This scheme is reliable for the design of switching regulators. Experimental and simulation results of this study show the high dynamics and robustness of the implemented system and the authors validate the proposed approach for power conditioning and UPS systems.

1.7 Symbolic Dynamics

Symbolic Dynamics is a branch of mathematics, which deals with spaces consisting of infinite sequences of symbols. Invented by Hadamard in 1898, it was developed in 1930s and applied to one-dimensional mapping since 1970s [13].

A Symbolic Dynamical system can be defined as a space consisting of infinite sequences of symbols taken from a finite set. Consider a map like

$$x_{n+1} = f(\lambda, x_n) \quad x \in i.$$

To know the complete description of the discrete time evolution of this map would require the knowledge of the whole set $\{x_i,\ i=0,1,2\ldots\}$. However, a rough idea of the description can be obtained by retaining the key feature of the evolution and ignoring the actual numbers in the set. In order to accomplish this change of notation, the phase space can be divided into different regions, denoting each region by a symbol, e.g., *A, B, X, Y…* Each number x_i will be represented by a symbol depending on the region in which it falls. If the regions are divided more thoughtfully, a set of rules can be defined for the order of these symbolic sequences. Using a finite number of symbols it is possible to generate an infinite sequence of symbols, which will resemble in nature with the original sequence. The concept is basically one of discretization of a continuous signal into discrete descriptors or symbols.

1.8 Literature search on Symbolic Dynamics

In 1994, Wu and Chua published a paper in which they developed the theory of Symbolic Dynamics on piecewise linear maps [14]. Using examples in signal processing, namely digital filters with overflow nonlinearity and sigma delta modulators, they demonstrated the application of Symbolic Dynamics on maps, which are composed of signum functions. The authors have also proved various results about periodic points and periodic sequences. In the Wu-Chua paper, the authors mainly concentrated on the set of initial conditions, which generates the same symbolic sequence. They discussed many properties of these sets considering different cases like a set with: 1) a single point, 2) zero measures, 3) nonzero measures. Considering probability as a measure of these sets they reached some conclusions on the generation of a particular sequence on the system.

In 1993 Peleties and DeCarlo used Symbolic Dynamics in controlling and modeling the macroscopic behavior of a 3-switched system [15]. They divided the system state space into eight disjoint regions compatible with a topological conjugate set of Symbolic Dynamics and thus a Petrinet model was developed for the reachability analysis of the system. With the help of this Petrinet, the authors tried to simulate the behavior of 3-switched system in terms of region-to-region behavior. Motion in a continuous/discrete variable system can be represented by a parallel shift map motion in Symbolic Dynamics. Thus using Symbolic Dynamics, analysis of the desired aggregate behavior is more convenient and it is also easier to deduce the new aggregate behavior.

In 1990 Azzouz and Hasler studied the chaotic dynamics of the *R-L* diode circuit [16]. They defined the itineraries for Symbolic Dynamics and also discussed some properties. In the paper, the authors have shown an urgent need for the rigorous analysis of the chaotic behavior of electrical circuits, by which it would be easier to prove the absence or presence of chaos. In the paper they tried to study the orbit structure of an *R-L* diode circuit. For this study they introduced the two-dimensional boundary transformation map. Unfortunately Azzouz and Hasler could arrive only at conjecture instead of a theorem that establishes the chaotic nature of a circuit.

.

1.9 Markov chain model

A Markov chain is a special type of stochastic process and a stochastic process is a collection of random variables. These terms are defined below [17].

Random variable: A random variable is a function or rule that assigns real numbers to each event in a sample space. Sample space can be defined as a set of all possible outcomes of an experiment.

Stochastic process: A stochastic process is a family of random variables defined for a sample space *S*. If the numbers of the family are countable then the process will be denoted by X_1, *X2*, *X3*… and it is called a discrete time process. If the members are uncountable, then the process will be denoted by $\{X_t; t$ ³ $0\}$ and the series is called a continuous time process. In discrete time process, the values X_1, X_2 … are called the state space and the process is called a chain.

Markov property: In an experiment if the probability of an outcome in future depends only on the present outcome (and not on the outcomes in the past), then it can be said that the experiment satisfies the 'Markov property'.

A Markov chain can be defined as *a stochastic discrete time process, which satisfies the Markov property.*

1.10 Literature search on Markov chain models

In 1999, Francher and Hulse published a paper in which they used a Markov chain model for testing of cables on a distribution voltage substation [18]. In this model Francher and Hulse applied nondestructive testing techniques to common distribution equipment. It simply identifies the equipment population having higher probability of failure. This technique is useful for gateway cables. Gateway cables are several hundreds feet long, and the testing of these cables requires them to be out of service during the test. Because of the required outage, it becomes expensive and inconvenient to test these cables.

Using a Markov chain model, the authors defined various possible states such as equipment population, failure, repair and testing. Transition between these states was characterized probabilistically. The Markov based testing program was able to identify a subpopulation that has a much higher probability of failure.

In 1993, Csenki used a Markov model for the reliability analysis of the recovery blocks [19]. The technique of the recovery blocks is a well-known fault tolerant software method in which many alternative modules are used for the same problem. Finally the results are verified by an acceptance test. Experiments show that the failure points in the input domain of the software are not isolated but form clusters, each one attributable to a common software fault. The author has described a reliability model for a recovery block. This model consists of a primary module, an acceptance test and two alternate modules. A Markov model is developed to obtain first and second moments of the successful input points for recovery blocks and thus it is used to study reliability of recovery blocks when the series of input values traverse nested clusters of failure points in the input domain.

CHAPTER 2

SYMBOLIC DYNAMICS

2.1 Introduction

The process of load modeling can be divided into two main parts, comparison of signals and forecasting of a signal. This chapter deals with these two features in detail. Comparing two signals means to determine the degree of similarity between the two signals. A conventional method to do so is based on the comparison of a signal with a sine wave and then calculating total harmonic distortion (THD) index. But this method is applicable to periodic signals only. Further, the periodic signal must possess a Fourier series. The proposed scheme uses the concept of Symbolic Dynamics to code a signal to develop an index that reflects the degree of similarity between the two signals.

2.2 Some definitions

The following are useful definitions in the general field of Symbolic Dynamics.

Signal: A signal is defined as a single dimensional array containing time series data of voltage or current.

Word: A word is a combination of consecutive data in the signal. A word can have one or more numerical values depending on the word length selected.

Dictionary: A dictionary is a collection of words formed from a signal array. A dictionary can be a multidimensional array, containing words from different signals, for example voltage signal and current signal.

Frequency: The number of time a word appears in a dictionary is called the frequency of that particular word in the dictionary.

Fractional occurrence: Fractional occurrence is the ratio of frequency of a word to the total number of words in that dictionary. f_{xi} denotes the fractional occurrence of i^{th} word in dictionary X.

Common signal index (CSI): The common signal index is a measure of distortion in two signals. A separate two-dimensional array is formed by taking all the common words and their frequencies in both dictionaries. For these common words, a common signal index is defined by using the fractional occurrences of the words in both dictionaries. The following five *CSIs* are considered,

$$CSI_1 = \sum_{i=1} (f_{xi}*f_{yi})/(f_{xi}+f_{yi})$$

$$CSI_2 = \sum_{i=1} (f_{xi}*f_{yi})/(1+|f_{xi}-f_{yi}|)$$

$$CSI_3 = \sum_{i=1}^{N} |f_{xi}-f_{yi}|/(1+f_{xi}+f_{yi})$$

$$CSI_4 = \sum_{i=1}^{N} (f_{xi}*f_{yi})/(1+f_{xi}+f_{yi})$$

$$CSI_5 = \sum_{i=1}^{N} (f_{xi}*f_{yi})/(f_{xi}+f_{yi} +f_{xi}*f_{yi})$$

where N is the total number of different words in the common dictionary.

Considering the results of various synthetic tests to be described in Chapter 3, CSI_1 was found to be the most appropriate. Therefore all the other *CSIs* were used only for comparative purposes in connection with synthetic tests.

2.3 Coding a signal in Symbolic Dynamics

In reality, signals are in the form of a time series data of voltage or current. These time series data can be treated as a series of symbols where each symbol represents a unique numerical value for a quantity (voltage or current). A signal can be shown as a single dimensional array of symbols,

$$c=[Q, W, E, R, T, Y, U, I, O, P, Q, E, S, D, F, G, H\ldots].$$

Different consecutive symbols can be grouped together to form a word, and each word represents a state of the signal. A signal can be coded by showing the transition between consecutive states. Transition is depicted in Figure 2.1.

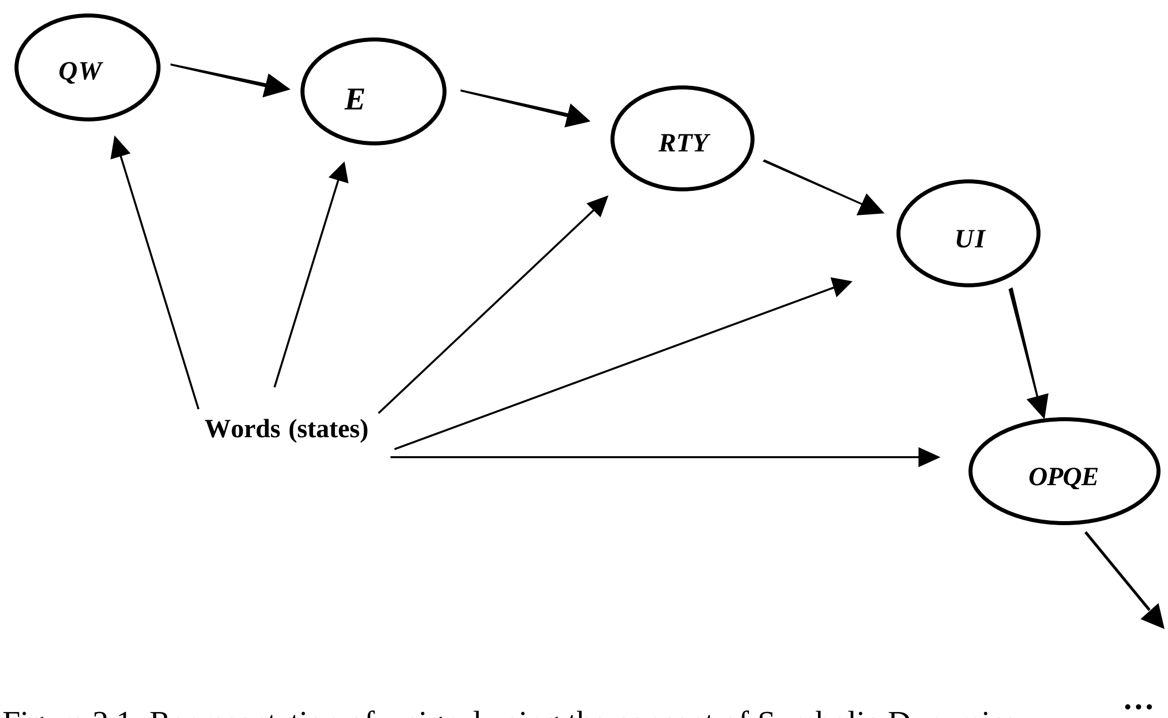
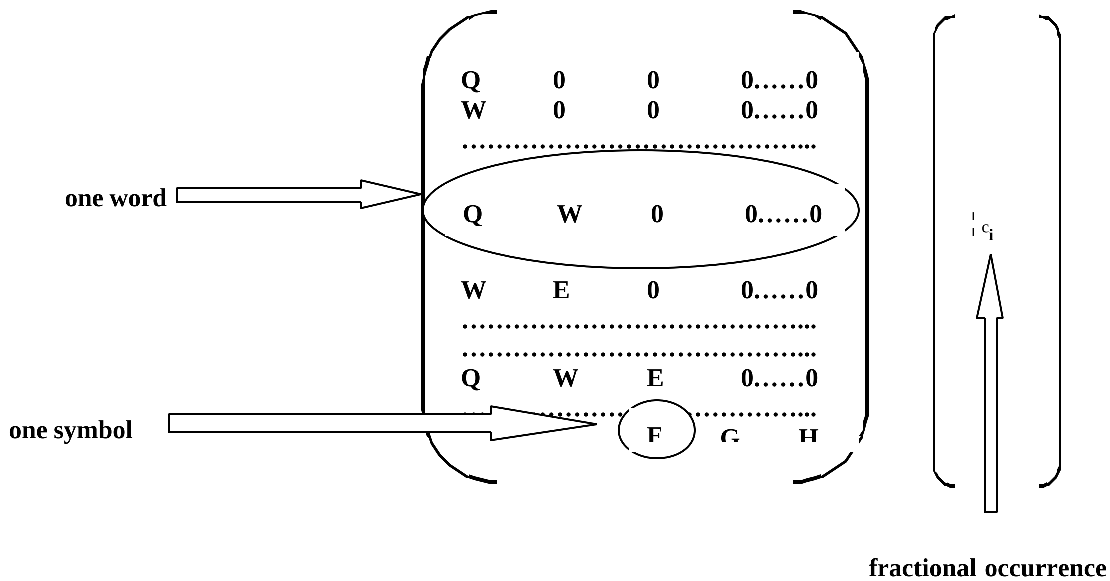


Figure 2.1 Representation of a signal using the concept of Symbolic Dynamics

2.4 Dictionary formation

A dictionary is a two dimensional array (considering one signal at a time). This array can be filled first by taking all the one symbol words, one at a time, then taking two consecutive symbols at a time, and so on, until n_w symbols at a time are taken. Note that n_w is the maximum word length defined for the dictionary. All the blank spaces are filled with zeros in the dictionary. Entries in the dictionary are called *words*. The dictionary also contains a separate column matrix that is filled with the fractional occurrence of each word in the dictionary. Figure 2.2 shows the process.



Signal: c ***=[Q, W, E, R, T, Y, U, I, O, P, Q, H,…, E, G, H]***

Figure 2.2 Forming a Symbolic Dynamic dictionary

If there are N data points in a signal (which means N symbols) and maximum word length is n_w then total number of words N_T in a dictionary can be calculated,

$$N_T = N+(N-1)+(N-2)+\ldots+(N-(n_w-1))$$

$$= n_w * N - ((n_w-1) * n_w/2)$$

$$= n_w[N-(n_w-1)/2].$$

2.5 Comparison of two signals

To compare two signals, a reference dictionary is formed from one signal. Then another dictionary is formed from the test signal. The two dictionaries are compared row by row and a mini dictionary is formed from all the common words in the two dictionaries. This mini dictionary also consists of two columns indicating the fractional occurrences of each word in each dictionary. The process is shown in Figure 2.3.

Signal 1: c =***[Q, A, Z, W, S, X…]*** Signal 2: g =***[Q, E, R, W, X, Y…]***

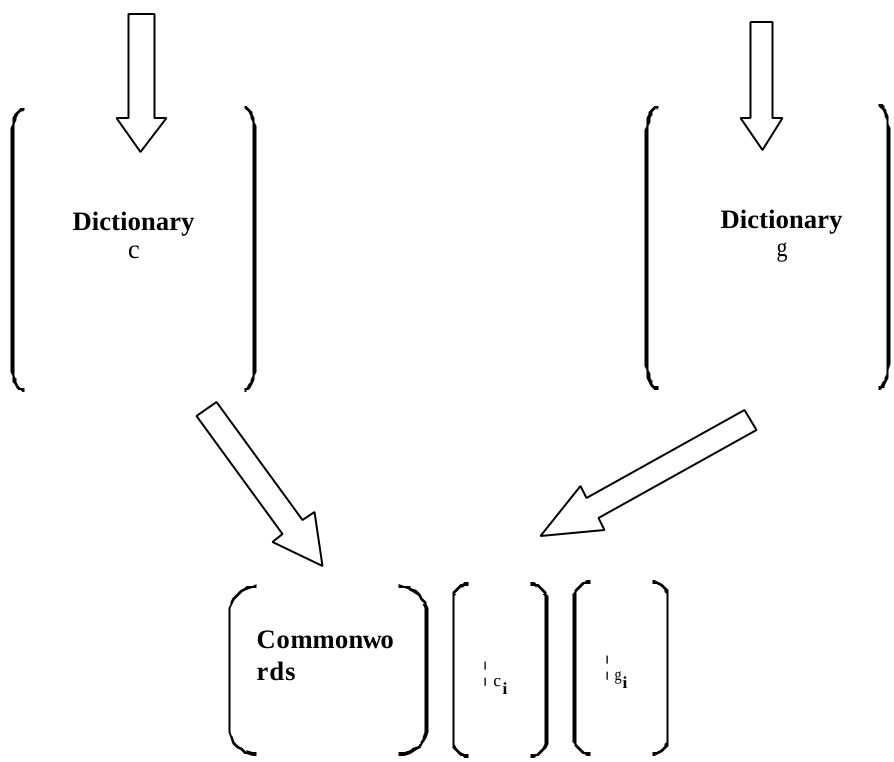


Figure 2.3 Comparing two signals

Once a mini dictionary is formed, *common signal index (CSI)* can be calculated by using the formula

$$CSI = \sum (f_{ci} * f_{gi})/(f_{ci} + f_{gi})$$

where $f_{\chi i}$ is the fractional occurrence of i^{th} common word in c dictionary, f_{gi} is the fractional occurrence of i^{th} common word in g dictionary, and CSI_{max}=0.5 (If the two signals are exactly identical). The lower the value of *CSI*, the higher the degree of dissimilarity between the two signals. Thus the value of *CSI* directly shows how much the two signals are in agreement.

2.6 Forecasting a signal

Attention turns to the forecasting or prediction of future values of a signal. For this purpose, a dictionary can be formed by taking the instantaneous values in a signal at different points. Once the dictionary is formed, a word is selected randomly from the dictionary and without looking at its past history, the subsequent word is identified. This process continues until a long sequence of words is generated. Thereafter it is possible to predict the future data from the sequence.

Model 1 (Single symbol checking)

1 Form a dictionary *[D,F]* from data, where *D* is the dictionary and *F* is the fractional occurrence of words.

2 Set count = 1.

3 Load *[D, F]* into working dictionary $[W_D, W_F]$.

4 Form CW_F column, which is the cumulative frequency of occurrence. Now the dictionary is in the form of $[W_D, W_F, CW_F]$.

5 Generate a random number *r*.

6 Find *i* such that $CW_F(i) < r$£ $CW_F\,(i+1)$. Choose $(i+1)^{th}$ word as an output.

7 Increment the count, count = count+1. If count = countstop, go to step-12 otherwise go to next step.

8 Form mini dictionary from *[D, F]* that begins with the last letter of the word, selected in step-6. Calculate the sum of the fractional occurrences of the mini dictionary and divide all the frequencies by this sum. Now the dictionary is $[M_D, M_F]$.

9 Load $[M_D, M_F]$ into $[W_D, W_F]$.

10 Go to step-4.

11 The output at step-6 is in the form of a matrix of the size of $(countstop\ x\ N_w)$ where N_w is the maximum word length. Load this matrix into a column matrix (row by row) after deleting the first column from second row onward. Now this column matrix represents the signal

It is easy to implement this approach except in the following two cases:

Case 1: No word begins with the last symbol of a particular word.

Case 2: If only a single symbol word begins with the last symbol of a particular word.

Solution: For the first case, if such a situation arises at a particular stage, the process is restarted from step-3. For the second case, to eliminate the single symbol words it is better to delete all such words from the main dictionary *D.*

To check the accuracy of the above algorithm (Model 1), synthetic tests were performed. Results of these synthetic tests showed that Model 1 did not work well for some cases. All the synthetic tests and results have been reported in Chapter 3.

The main drawback of Model 1, described above, was that the information from the historical data set was not being utilized fully. The data were used only to form a dictionary. The model was predicting new value by checking the appearance of one symbol (one data point) only, in the past history. In other words the model was never utilizing a sequence of symbols to predict the next (new) value.

2.7 Development of a model based on multiple symbol checking (Model 2)

The foregoing is a description of a 'first-cut' model for signal forecasting. The main drawbacks identified above can be alleviated through relatively minor modification of the algorithm of Model 1. The aim of the Model 2 is to overcome the drawbacks of Model 1 and enable the new algorithm to identify a long sequence of symbols to predict the next word.

Model 2

Input: historical current data in the form of a row vector.

1. Form a dictionary from the data set.
2. Scan through the dictionary and take out all the words in which the order of symbols is the same as in the data set (starting from the last data and looking backward). In the worst case a word may start with the last data point. In other words, it catches a sequence length of one symbol only.

3. Form a mini dictionary of all the above words including the sequence length of each word and their fractional occurrences in the main dictionary.
4. Modify the fractional occurrences, multiplying each element by square of its sequence length and then divide each element by the sum of all entries.
5. Form a cumulative modified fractional occurrence column, *cfr*.
6. Generate a random number and scan through the *cfr* column. If the number lies between *cfr(i)* and *cfr(i+1)*, select $(i+1)^{th}$ word as a new word.
7. Check the sequence of symbols in the word. The symbol just next to the last historical data point will represent the predicted current value.
8. Case 1: Update the historical data set by including the above value and repeat the process.

   Case 2: Update the historical data set by including the true value of current.

Case 1 and Case 2 above can be used depending on the application. If the aim is to develop a load model only, then Case 1 is appropriate. The other possibility is that this scheme can also be used for on line applications such as real time active load current compensation for power conditioning and real time generator dispatch. Case 2 is best suited to these applications.

Synthetic tests and results for both Models 1 and 2 have been presented in the next chapter.

CHAPTER 3

SYNTHETIC TESTS AND RESULTS

3.1 Introduction

To check the accuracy of the proposed Symbolic Dynamic load model, various synthetic tests were performed and on the basis of the results of these tests the model was modified and some important conclusions were made. It may be better to check a proposed model or hypothesis by implementing it on a synthetic set of input data. The advantage of doing so is that the result obtained can be compared with the expected result and the model can be modified in the first stage itself before implementing the actual test conditions.

Appendix A contains the Matlab code for all synthetic tests.

3.2 Nomenclature

Synthetic tests discussed in this chapter can be divided into five broad categories, depending on the different models / hypothesis to be checked:

A. Selection of the most suitable index for comparing two signals.

B. Justification of the selected index.

C. Forecasting a signal using Model 1 (single symbol checking).

D. Forecasting a signal using Model 2 (multiple symbol checking) with Case1 (load modeling application).

E. Forecasting a signal using Model 2 (multiple symbol checking) with Case 2 (power conditioning application).

This chapter deals with a number of tests belonging to the each category. A nomenclature has been used to refer to a particular test in this chapter. In this nomenclature, a test designator is used in the form

XX-Ynn.

The entry *XX* can be "*ST*" to designate the use of synthetic (computer generated) data. The entry *Y* represents one of the five categories, mentioned above, to which the test belongs. The numbers *nn* refer to the serial number of the test. Thus, as examples,

ST-C03 refers to the synthetic test number 3, belonging to the category *C*.

RT-E15 refers to the real test number 15, belonging to the category *E*.

Description of Case 1 and Case 2 appear in Chapter 2.

3.3 Test conditions

In the synthetic tests, some parameters that describe the synthetic data, have been assumed. Based on these parameters, a signal was formed synthetically and the proposed algorithm was implemented on this signal. All these parameters have been tabulated for different tests in Table 3.1 and Table 3.2.

3.4 Analysis tools

To analyze the results of the tests, some parameters have been taken into consideration. Especially while forecasting a signal, it is difficult to comment on the goodness of the model unless some characteristics of the expected signal and predicted signal are compared. Two important parameters to be compared in this regard are RMS and average.

Table 3.1 Test statistics for categories *A* and *B**

Test	Signal 1	Signal 2	Interval (radian)	Points per cycle	Maximum word. length	Words in dictionary.
ST-A01	$\sin(\theta)$	$\sin(\theta+\delta)$	$[0,24\pi]$	50	3	1797
ST-B01	$\sin(\theta)$	$\theta+\sin(\theta)$	$[0,24\pi]$	50	3	1797
ST-B02	$\sin(\theta)$	random no.	$[0,24\pi]$	50	3	1797
ST-B03	$\sin(\theta)$	$\sin(\theta)$+noise	$[0,24\pi]$	50	3	1797

* All the listed tests are for *CSI* verification.

Table 3.2 Test statistics for categories *C*, *D* and *E*

Test	Signal	Word length	ΔT (s)	T (s)	Cycles
ST-C01	Triangular	5-10	0.5	20	2
ST-C02	Sinusoidal	2-5	1.0	25	4
ST-C03	Sine + harmonics (h=3,5)	2-5	1.0	25	4
ST-D01	Triangular	5	0.5	40	2
ST-D02	Sinusoidal	5	1	25	4
ST-D03	Sine + harmonics (h=3,5)	5	1	25	4
ST-E01	Triangular	5	0.5	40	2
ST-E02	Sinusoidal	5	1	25	4
ST-E03	Sine + harmonics (h=3,5)	5	1	25	4

values of the signal. It is also prudent to consider the behavior of the earlier defined index (*CSI*), used for the comparison of two signals. The predicted signal and the expected signal have also been compared using a statistical technique, treating these two signals as two different data sets. The statistical test used for this purpose is *Kolmogorov-Smirnov test (KS test)*, for general differences in two populations [20].

KS test

The KS method involves identifying empirical distribution functions for the two populations and then assessing whether there are any differences between the two probability distributions. Let m and n be the number of elements in the two populations X and Y respectively. For every real number t, empirical distribution functions are defined as

$$F_m(t)=(\text{number of sample } X \text{ £ } t)/m$$

$$G_n(t)=(\text{number of sample } Y \text{ £ } t)/n.$$

The point of interest is to test the following hypotheses

$$H_0 : [F(t) = G(t) \text{ for all } t]$$

$$H_1 : [F(t) \text{ ¹ } G(t) \text{ for at least one } t].$$

If d is the greatest common divisor of m and n, then set

$$J = (mn/d) * \max_{(-\infty<t<\infty)} \{|F_m(t) - G_n(t)|\}.$$

The statistics J is called the two-sided two-sample Kolmogorov-Smirnov statistics.

Let Z denote the ordered values for combined populations X and Y such that,

$$Z(i) \text{ £ } Z(i+1).$$

Now J can be rewritten as

$$J = (mn/d) * \max_{i=1,2\ldots N} \{|F_m(Z(i)) - G_n(Z(i))|\}$$

where $N=m+n$. For large-sample populations, the above statistics can be approximated based on the asymptotic distribution of J, suitably normalized, as min (m, n) tends to infinity. Therefore,

$$J^* = (mn/N)^{1/2} * \max_{i=1,2\ldots N} \{|F_m(Z(i)) - G_n(Z(i))|\} = d/(mnN)^{1/2} * J \textbf{.}$$

For large samples,

$$P_0(J^* < s) \rightarrow \{\sum_{k=-\infty}^{\infty} (-1)^k e^{-2k^2 s^2} ,0\} \textit{ for\{s>0, s £0.}$$

The function *Q(s)* can be defined as

$$Q(s) = 1 - \sum_{k=-\infty}^{\infty} (-1)^k e^{-2k^2 s^2} \text{ , } s>0$$

$q^*{}_a$ is defined by

$$Q(q^*{}_a)=a$$

Table A.11 in [20] lists value of function *Q(s)*, Thus *a* can also be obtained by the same table. Appendix B shows a small illustrative example of the technique.

To test the earlier defined hypothesis, the *KS* method states,

Reject H_0 *if* J^* ³ $q^*{}_a$.

For example, from the cited table, for $J^*=1.5$,

$$a = Q(1.5) = 0.0222.$$

Which means the lowest significance level at which H_0 can be rejected is 2.22%. Thus for the two populations to be identical, the desired value of *a* is 1.

3.5 Test results

ST-A01: The aim of this test is to identify the best index out of five indices proposed in Chapter 2. In signal 2, δ was varied from 0 to 90 degrees to check the *CSI* in each case. Theoretically as phase angle increases, the number of common words in two signals should decrease, and correspondingly there should be a regular increasing or decreasing tendency of *CSIs*. When the program was run to obtain the numerical values, it was observed that all the *CSIs* except CSI_3 had the decreasing tendency as the phase angle increased. For phase angle zero, i.e. when both the signals are same, CSI_1 and CSI_3 has the value 0.5 and 0 respectively. This indicates that CSI_1 has the maximum value of 0.5 and CSI_3 has the minimum value of 0 and these values are for two identical signals. In all the other indices these maximum values were some arbitrary numbers. Moreover when all the indices were plotted against phase angle, CSI_1 was found to be the best. Therefore the other *CSI*s were discarded for any future analysis.

Numerical results of the test have been tabulated in Table 3.3.

ST-B01: In this test it is obvious that as θ increases signal 2 will rise higher and higher, as in Figure (3.1). Therefore the common points on the two signals will lie only in the initial period when θ is small. Comparing the two signals *CSI* was found to be 0.0462 which is well below 0.5000 and shows a very few common points.

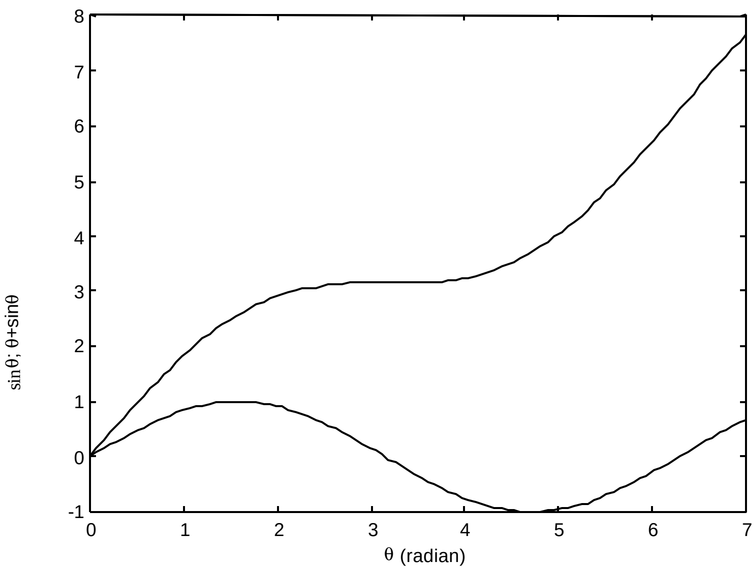


Figure 3.1 Synthetic signals for *ST-B01*

Table 3.3 Variations of *CSI* with phase angle

Serial No.	δ (degrees)	CSI_1	CSI_2	CSI_3	CSI_4	CSI_5
1	0	0.5000	0.0196	0.0000	0.0183	0.4952
2	15	0.5000	0.0196	0.0033	0.0183	0.4951
3	30	0.4511	0.0173	0.0962	0.0163	0.4469
4	45	0.4083	0.0168	0.1028	0.0158	0.4042
5	60	0.4078	0.0167	0.1045	0.0158	0.4037
6	75	0.4002	0.0166	0.1042	0.0156	0.3962
7	90	0.3638	0.0130	0.1515	0.0125	0.3637

ST-B02: Here the Matlab inline function *rand* will generate a random number uniformly distributed between 0 and 1. Therefore not many common points are expected between the two signals and hence a low value of *CSI*. The value of *CSI* obtained was 0.1469.

ST-B03: In this test signal 1 is pure signal and signal 2 is contaminated with noise having variable signal to noise ratio (*S/N*). Increasing *S/N*, it was found that *CSI* is also increasing. When *S/N* is 1, it indicates that a significant amount of noise is present in the signal and therefore a low value of *CSI* is expected. The value obtained in the test was 0.1242. When *S/N* is 10,000, it indicates that negligible amount of noise is present in the signal. In other words the pure signal and the contaminated signal are almost identical. The value of *CSI* obtained for this case was 0.5, which supports the claim that identical signals have maximum value of *CSI*.

All the values have been tabulated in Table 3.4

ST-C01, ST-C02, and ST-C03: These tests are the same type of tests, belonging to category C, for three different types of signals, namely, triangular, sinusoidal, and sinusoidal contaminated with third and fifth harmonics. Since prediction of each future value depends on random number generation, therefore three trials are reported here for each test. Comparison of different parameters of the expected and forecasted signal has been shown in Table 3.5.

Table 3.4 Variation of *CSI* with *S/N* ratio, Test *ST-B03*.

S/N	*CSI*	*S/N*	*CSI*
1	0.1242	100	0.3598
2	0.2570	1000	0.3598
3	0.2251	2000	0.4191
4	0.1796	3000	0.4636
5	0.2825	4000	0.4191
6	0.1982	5000	0.4258
7	0.2694	6000	0.4636
8	0.2771	7000	0.4636
9	0.3641	8000	0.4636
10	0.2570	9000	0.4636
50	0.2594	10000	0.5000

ST-D01, ST-D02, and ST-D03: Tests for the category *D* were also performed on three different signals. In this category, signal forecasting is done only for the purpose of load modeling. In other words the process is not in real time. This algorithm works in offline mode. Input to the model is the historical data (current values) recorded in the past. To generate the values of current, a simple circuit, shown in Figure 3.2, is assumed and different input voltage signals were applied.

Table 3.5 Results for tests in category C

Test	Index	Actual Signal	Regenerated Signal		
			Trial 1	Trial 2	Trial 3
ST-C01	*CSI*	0.5	0.4903	0.4882	0.4878
	RMS (p.u.)	3.7482	3.8951	4.0558	4.0522
	% RMS deviation	0	3.9	8.2	8.1
	Average (p.u.)	3.4390	3.6412	3.7946	3.8018
	α (KS test)	1.0	0.98	0.78	0.82
ST-C02	*CSI*	0.5	0.4898	0.4892	0.4758
	RMS (p.u.)	2.1190	2.0567	1.9937	1.9286
	% RMS deviation	0	2.9	5.9	8.9
	Average (p.u.)	2.0	1.9382	1.8724	1.8069
	α (KS test)	1.0	0.9987	0.8073	0.4503
ST-C03	*CSI*	0.5	0.4822	0.4910	0.4919
	RMS (p.u.)	2.1364	1.9369	2.0509	2.0257
	% RMS deviation	0	9.3	4.0	5.1
	Average (p.u.)	2.0	1.7987	1.9105	1.8827
	α (KS test)	1.0	0.4806	0.9981	0.97

The value of current through the circuit can easily be find out from the expression

$I = V_{in}/(R_1+R_2)$.

For the sake of simplicity, the values of R_1 and R_2 are selected as 1 ohm. Voltage values can also be predicted using this circuit. At any instant of time t, if the predicted current value is I_{pred}, then the predicted output voltage is given by

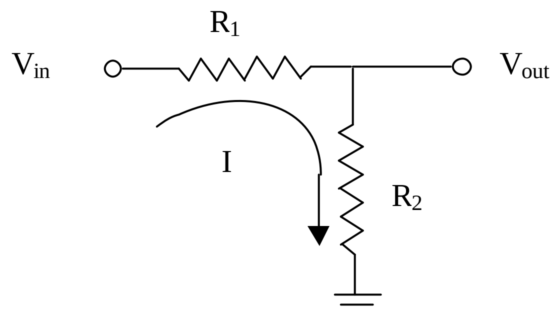


Figure 3.2 Synthetic circuit to generate current data

$$V_{pred} = V_{in} - I_{pred} * R_1.$$

Thus predicted and actual output power can also be obtained to validate the model.

Forming a dictionary with the historical data set, future values are predicted and the predicted values are not used to update the historical data set. All the results have been reported in the Table 3.6.

ST-E01, ST-E02, and ST-E03: The *E* category represents the online signal-forecasting model. Again three tests were performed for three different signals. Historical data were generated in the same manner as in the tests of category *D*. This algorithm is suitable for the real time, online applications such as active filtering of current and real time power conditioning.

In these tests, *2n* cycles of a signal are selected. From the selection, *n* cycles represent the historical data set and the next *n* cycles represent the actual value of the future current.

Table 3.6 Results for category *D* Tests *ST-D01*, *ST-D02*, and *ST-D03*

Test	Index	Actual Signal	Regenerated Signal		
			Trial 1	Trial 2	Trial 3
ST-D01	*CSI*	0.5	0.4756	0.4861	0.4892
	RMS (p.u.)	3.3307	3.5036	3.54	3.3704
	% RMS deviation	0	5.19	6.28	1.19
	Average (p.u.)	3.0000	3.275	3.2563	3.0875
	Average power (p.u.)	11.0938	8.7938	6.3344	10.2063
	α (KS test)	1.0	0.4355	0.6945	1.0
ST-D02	*CSI*	0.5	0.5	0.5	0.5
	RMS (p.u.)	1.0607	1.0607	1.0607	1.0607
	% RMS deviation	0	0	0	0
	Average (p.u.)	1.0	1.0	1.0	1.0
	Average power (p.u.)	1.1250	1.1250	1.1250	1.1250
	α (KS test)	1.0	1.0	1.0	1.0
ST-D03	*CSI*	0.5	0.5	0.5	0.5
	RMS (p.u.)	1.0695	1.0695	1.0695	1.0695
	% RMS deviation	0	0	0	0
	Average (p.u.)	1.0	1.0	1.0	1.0
	Average power (p.u.)	1.1439	1.1439	1.1439	1.1439
	α (KS test)	1.0	1.0	1.0	1.0

From the historical data set, a future current value is predicted. Each time a prediction is made, the data set is updated by the actual current. Table 3.7 shows the comparison of different parameters for actual and expected values of the future current.

In this chapter, signal comparison method, Model 1 and Model 2 were checked using synthetic data generated by computer. Advantage of synthetic test is that the models can be checked in the first stage before moving to the real tests and can be improved or eliminated depending on the results obtained. Results of synthetic tests show that Model 2 is better than Model 1 and it gives reasonable accuracy with synthetic data. Comparison of the results for Model 1 and Model 2 is shown in Figure 3.3.

Table 3.7 Results for category *E*, Tests *ST-E01*, *ST-E02*, and *ST-E03*

Test	Index	Actual Signal	Regenerated Signal		
			Trial 1	Trial 2	Trial 3
ST-E01	*CSI*	0.5	0.4665	0.4925	0.4890
	RMS	3.3307	3.3518	3.3242	3.3279
	RMS dev. (%)	0	0.63	0.19	0.085
	Average	3.0	3.0125	3.0	3.0
	Average power	11.0938	11.0625	11.075	11.0813
	α (KS test)	1.0	1.0	1.0	1.0
ST-E02	*CSI*	0.5	0.4697	0.4878	0.4735
	RMS	1.0607	1.0581	1.0607	1.0590
	RMS dev. (%)	0	0.25	0	0.16
	Average	1.0	0.9947	0.9994	0.9983
	Average power	1.1250	1.1237	1.1241	1.1241
	α (KS test)	1.0	1.0	1.0	1.0
ST-E03	*CSI*	0.5	0.4557	0.4845	0.4733
	RMS	1.0695	1.0567	1.0624	1.0659
	RMS dev. (%)	0	1.2	0.66	0.34
	Average	1.0	0.9866	0.9911	0.9952
	Average power	1.1439	1.1387	1.1414	1.1362
	α (KS test)	1.0	1.0	1.0	1.0

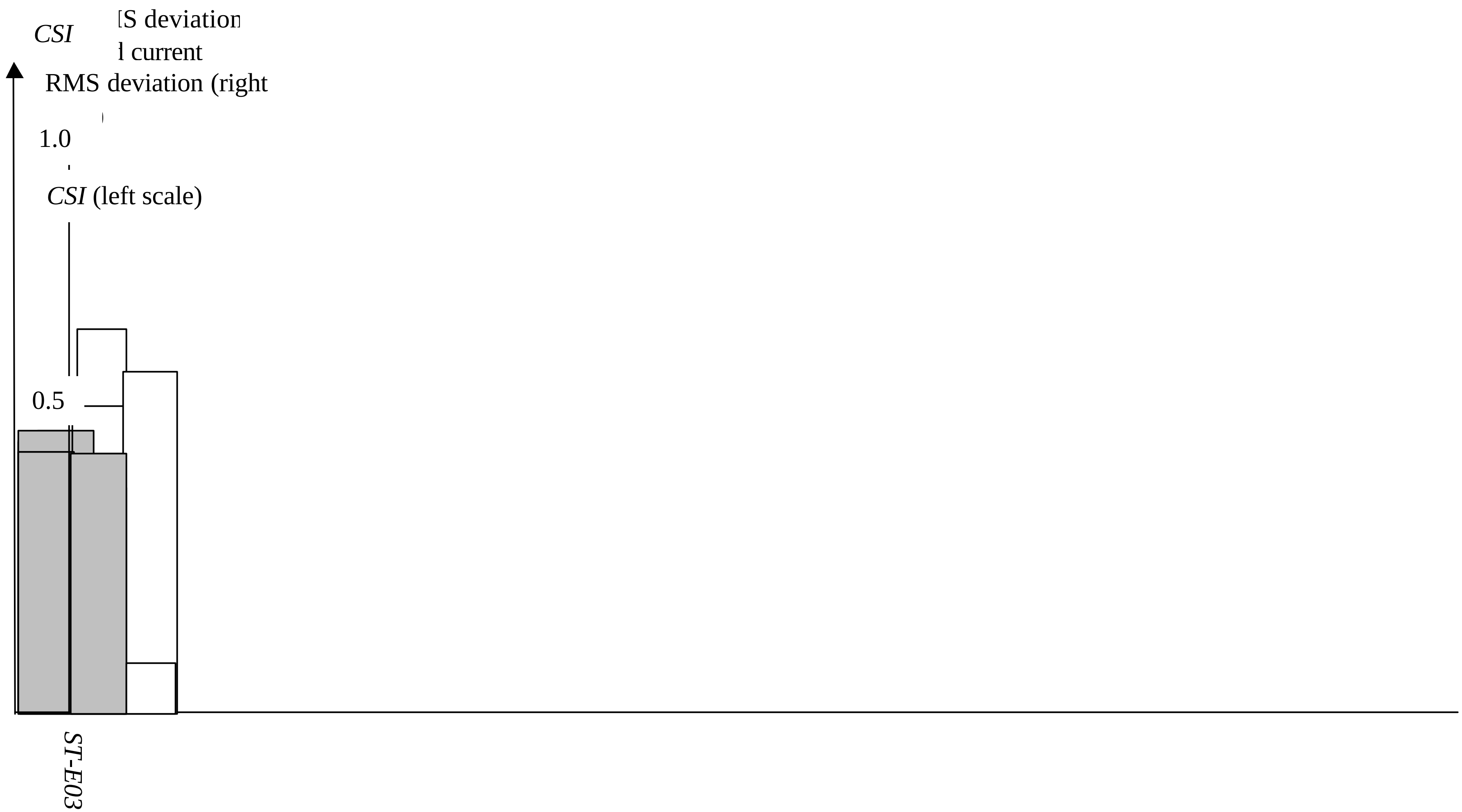


Figure 3.3 Summary of test results for Model 1 and Model 2

CHAPTER 4

TESTS ON INDUSTRIAL ARC FURNACE DATA

4.1 Introduction

This chapter deals with the tests on industrial arc furnace data. The data used for this purpose have been recorded from an electric arc furnace (EAF) under normal operating condition. The specifications of the furnace are shown in the Table 4.1. Appendix C contains the EAF schematic and sample current and voltage values. The algorithms to forecast a signal for load modeling and power conditioning application, as discussed earlier, are tested on “real” data in this chapter. The arc furnace current in different phases is forecasted using the algorithms, assuming that the supply voltage is constant. The results obtained are compared with the actual measured current. The comparison is done for RMS and average value, *CSI*, and *KS* index. Appendix A contains the Matlab code for all industrial arc furnace test.

Table 4.1 EAF specifications

4.2 Data transformation

Parameter	Value
Type	3-phase AC
Process	Scrap metal refining
Supply voltage	34.5 kV
Rated current	0.84 kA
Nominal power	50 MVA

Recorded current data from the electric arc furnace have been measured through a CT with ratio 400/5. To represent these data by symbols and thus to form a Symbolic Dynamic dictionary, the data needs to be discretized in a suitable manner. To discretize, the data set can be divided into different cells. Let N is the number of cells and $?x$ is the width of each cell, the first cell denotes the minimum value of current and the N^{th} cell denotes the maximum value of current. The process of discretization is shown in the Figure 4.1.

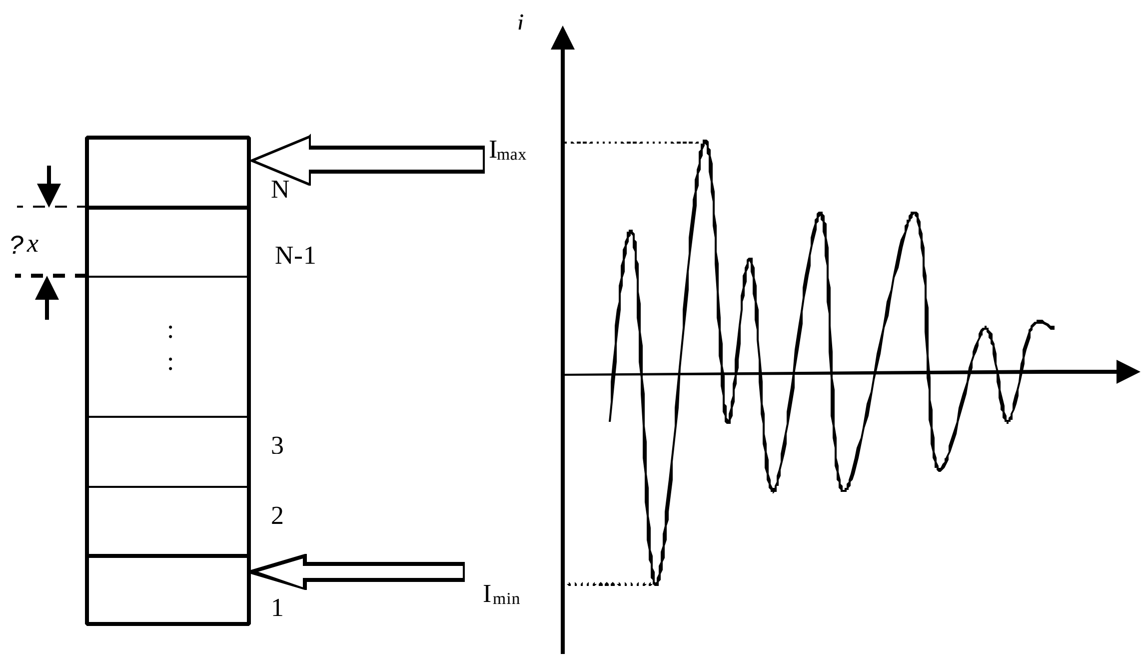


Figure 4.1 Transforming the real data

Division of cells can be done in the following two ways:

1. Constant cell width: Keeping cell width $?x$ constant, the data set can be divided into N number of cells where N depends on minimum and maximum value in the data set, thus

$$N=(I_{max}-I_{min})/\ ?x$$

2. Constant number of cells: The data set can be divided into N number of cells keeping cell width as a variable. Cell width depends on the maximum and minimum value of current in the data set, thus

$$?x = (I_{max}-I_{min})/N$$

4.3 Test conditions

The real data, obtained directly from an EAF, have been measured with a sampling frequency of 10 kHz. Therefore the number of measured data points in a cycle can be obtained as follows:

supply frequency = 60 Hz

one cycle = 1/60 second

$$\text{number of data points in one cycle} = \frac{1/60}{1/10^4} = 500/3 = 166.7.$$

or 500 data points = 3 cycles.

To form a Symbolic Dynamic dictionary and to predict future values, the real data is discretized as described above. To do so, the data set is shifted and scaled. In other words if X is the data set under consideration then transformed data set Y is obtained as

$$Y = fix\,(aX+b),$$

where *fix* is Matlab function for rounding off, a and b are constants. The values of the constants a and b depend on the maximum and minimum values in the data set and how cell division is done to transform the data.

After predicting the future data, predicted data set is shifted back to the original, by subtracting the constant b. This is done in order to get the true value of RMS error, when compared with the actual values.

In order to assess the performance of the Symbolic Dynamic algorithm, several parameters are varied. These include: word length, number of historical data points, number of predicted data points and number of cells used for transforming data. All these statistics have been tabulated in Table 4.2 for different tests.

4.4 Test results

RT-D01: In this test, measurements for 18 cycles were used as historical data set and 12 cycles of current were predicted using the load-modeling algorithm. Results obtained are reported in Table 4.3. In the table, two types of error in the RMS value have been shown: one is the deviation from the RMS historical data and the other is the deviation from the actual RMS current. Three trials of the test have been reported in the table. Deviation of RMS value from the historical data is less than 10% in all the cases while the deviation from the actual data is greater than 10% in two out of three cases, which indicates that the accuracy of the result is not very good. In Table 4.3 to Table 4.9, the units of current (used in average and RMS values) are scaled values. Because the scale factor varies from test to test, the average and RMS currents reported in Table 4.3

Table 4.2 Test statistics for different tests

Test*	Historical data	Predicted data	Word length	Number of cells
RT-D01	3000	2000	5	10
RT-D02	6000	4000	5	10
RT-D03	6000	4000	5	10
RT-D04	6000	4000	5	10
RT-D05	6000	4000	5	Variable
RT-E01	100	500	5	Variable
RT-E02	100	500	4	Variable
RT-E03	100	500	4	10
RT-E04	100	500	4	10

***** Nomenclature for the tests has been described in Section 3.1. Model 2 is used for all the tests.

Table 4.3 Results for *RT-D01*

Index		Historical current	Actual current	Predicted current		
				Trial 1	Trial 2	Trial 3
Average (p.u.)		-0.3642	-0.4887	-0.7267	0.5263	-0.1522
RMS (p.u.)		1.9632	2.1416	2.0542	1.9117	1.7911
% RMS deviation	historical			4.63	2.62	8.77
	actual			4.08	10.73	16.37
CSI			0.5	0.4723	0.4389	0.4663
J value			0	2.9567	9.1232	3.7789
KS index			1	0	0	0

cannot be compared to those in other tests (e.g., Table 4.4).However, the percentage deviations, *CSI*, *J* value, and KS indices are all unitless measures and these indices may be compared from test to test

RT-D02, RT-D03, and RT-D04: In all these tests historical data set consists of 36 cycles of current and predicted data are for 24 cycles. The three tests deal with phase A, B and C currents of the electric arc furnace mentioned earlier. Results obtained are reported in Table 4.4, Table 4.5 and Table 4.6. Considering the deviation of RMS value from historical data, out of a total of 9 cases, 5 are less than 5% and 4 are between 5 to 10%. Deviation from actual RMS current is greater than 10% in two out of a total of 9 cases. In two cases the RMS percentage error was found less than 1%. This deviation of

Table 4.4 Results for *RT-D02*

Index		Historical current	Actual current	Predicted current		
				Trial 1	Trial 2	Trial 3
Average (p.u.)		-0.3252	-0.4666	0.2121	-0.7794	-0.1389
RMS (p.u.)		1.9856	2.0622	2.0695	2.1610	2.1676
% RMS deviation	historical			4.2267	8.84	9.17
	actual			0.3526	4.79	5.11
CSI			0.5	0.4472	0.4743	0.4644
J value			0	9.2909	5.6014	6.6635
KS index			1	0	0	0

Table 4.5 Results for *RT-D03*

Index		Historical current	Actual current	Predicted current		
				Trial 1	Trial 2	Trial 3
Average (p.u.)		-0.4843	-0.4744	-0.3596	-0.1691	-0.2556
RMS (p.u.)		2.08	2.1670	1.8848	1.9820	2.1471
% RMS deviation	historical			9.56	4.9	3.02
	actual			13.02	8.53	0.92
CSI			0.5	0.4858	0.4815	0.4898
J value			0	3.6001	3.8796	3.1976
KS index			1	0	0	0

Table 4.6 Results for *RT-D04*

Index		Historical current	Actual current	Predicted current		
				Trial 1	Trial 2	Trial 3
Average (p.u.)		-0.5973	-0.5434	-0.6997	-0.3934	-0.9039
RMS (p.u.)		2.0306	1.8633	1.9582	1.9899	2.2042
% RMS deviation	historical			3.57	2.0	8.55
	actual			5.09	6.8	18.3
CSI			0.5	0.4686	0.4431	0.4422
J value			0	1.7889	3.6001	5.6237
KS index			1	0.0033	0	0

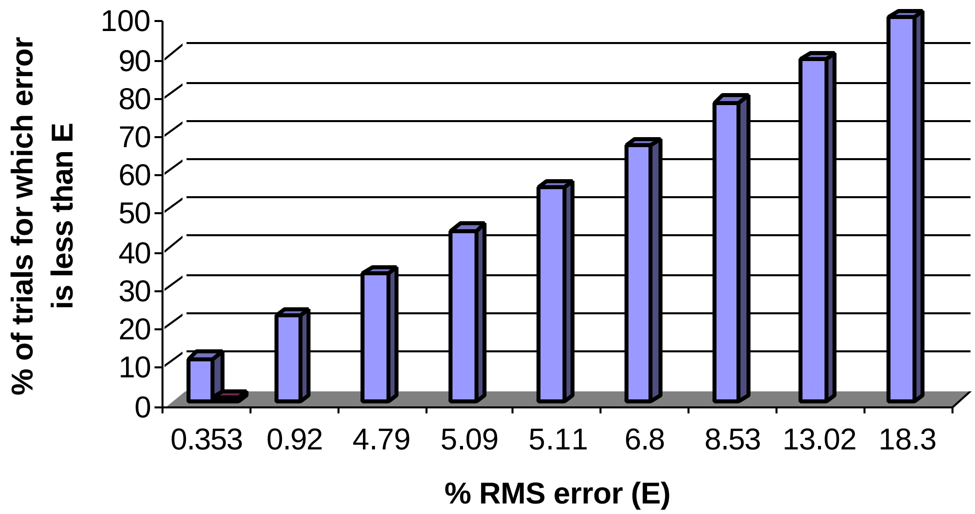


Figure 4.2 RMS value comparison with historical data in tests *RT-D02, RT-D03* and *RT-D04*

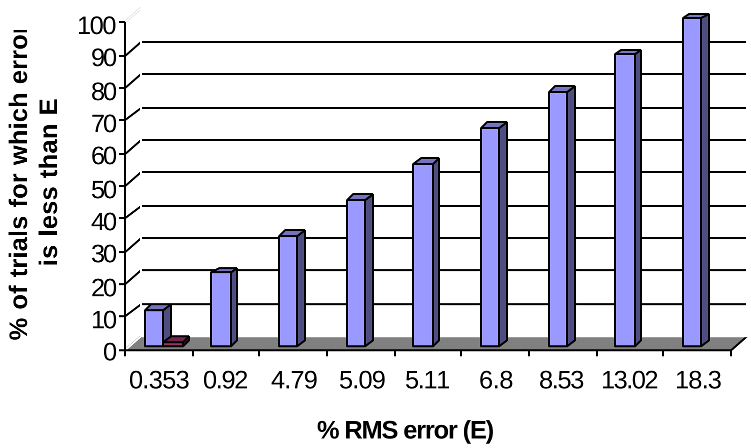


Figure 4.3 RMS value comparison with actual data in *RT-D02, RT-D03* and *RT-D04*

RT-D05: The aim of this test was to verify the proposed model from the point of view of power consumption. To perform this test, a sinusoidal voltage signal was generated synthetically with 60 Hz frequency. The value of voltage at different instants was recorded with the same sampling frequency as of real arc furnace current. The real current and synthetic voltage was discretized in the same manner. Using the algorithm, future current value was predicted in the discretized form. Assuming the voltage to be constant, average power was obtained for predicted current and actual current. For each phase, comparison of the predicted and actual average power for 10 different trials has been reported in Table 4.7.

Table 4.7 Error in the average power for test *RT-D05*

Trial	Absolute % error in average power of the predicted EAF current		
	Phase A	Phase B	Phase C
1	8.13	14.71	8.73
2	6.37	13.01	0.45
3	0.19	6.24	10.18
4	0.19	6.24	10.99
5	3.98	14.71	7.22
6	10.09	13.01	5.87
7	3.45	14.71	11.12
8	9.09	6.24	10.35
9	4.37	6.24	8.73
10	0.66	13.01	0.45
Expected error (%)	4.65	10.81	7.41

RT-E01: As mentioned in the previous chapter, *E*-category tests belong to the algorithm applicable for power conditioning. In this particular test, 100 data points of current have been taken as the historical data set and 500 points of future current have been generated. The numbers of cells are variable in this test and are 29, 61, and 62 respectively in three trials. Three trials are for phase A, B, and C current of the arc furnace respectively. Results for this test are reported in Table 4.8.

Two types of RMS error are reported here. The first is the RMS deviation within the transformed data. The second reported error is in terms of actual CT secondary

Table 4.8 Results for *RT-E01* (all the data are for EAF load current)

Index		Trial 1		Trial 2		Trial 3	
		actual	predicted	actual	predicted	actual	predicted
Average (p.u.)		-0.606	-0.534	0.518	-0.398	0.054	1.06
RMS (p.u.)		2.0194	1.9713	17.2887	16.931	17.6216	17.3620
% RMS deviation.	(1)		2.38		2.07		1.47
	(2)		11.36		4.34		3.76
CSI		0.5	0.4856	0.5	0.4745	0.5	0.4755
J value		0	0.1265	0	0.6008	0	0.5376
KS index		1	1	1	0.8643	1	0.9325
Exec. Time (hh:mm:ss)		00:25:18		02:11:02		02:14:58	

(1) RMS deviation within the transformed data.
(2) RMS deviation, comparing with actual (untransformed) current.

current (i.e., untransformed current). The first deviation shows how closely this algorithm can predict the future values. The second deviation is the actual measure of accuracy when this approach is used for a power conditioning application. The actual error in two out of three cases is less than 5%, and deviation within the transformed data is less than 3% in all the cases. *KS* index and *CSI* values are also good in this test.

RT-E02: In this test maximum word length as well as the number of cells was reduced. Therefore less execution time is expected with degraded accuracy. The number of cells used was 3,7,7 respectively in three trials. Because only three numbers cells are used in the first trial, the RMS value deviated by 71.6%. Other indices were also affected due to the small number of cells. Results are listed in Table 4.9.

Table 4.9 Results for *RT-E02* (all the data are for EAF load current)

Index		Trial 1		Trial 2		Trial 3	
		actual	predicted	actual	predicted	actual	predicted
Average (p.u.)		0.0	0.004	0.04	0.062	-0.028	-0.028
RMS (p.u.)		0	0.0632	1.41	1.4036	1.4283	1.4283
% RMS deviation	(1)		indeterminate		0.4537		0
	(2)		71.6		20.7		20.83
CSI		0.5	0.0	0.0	0.079	0.5	0.0822
J value		0	7.7476	7.7476	7.0835	0	7.3365
KS index		1	0	0	0	1	1
Exec. time (hh:mm:ss)		00:00:23		00:01:43		00:05:56	

(1) RMS deviation within the transformed data.
(2) RMS deviation, comparing with actual (untransformed) current.

RT-E03: The number of cells was kept fixed at 10 in this test, keeping the maximum word length the same as in *RT-E02* namely 4. Results obtained are shown in Table 4.10. It is evident from the results that as the number of cells and maximum word length increase accuracy increases but the execution time also increases.

Table 4.10 Results for *RT-E03* (all the data are for EAF load current)

Index		Trial 1		Trial 2		Trial 3	
		actual	predicted	actual	predicted	actual	predicted
Average (p.u.)		-0.4364	-0.4244	-0.3925	-0.4185	-0.4884	-0.5044
RMS (p.u.)		1.0375	1.0816	2.8871	2.8981	2.912	2.9069
% RMS deviati on.	(1)		0.3162		0.3784		0.174
	(2)		very high		43.89		8.63
CSI		0.5	0.4677	0.5	0.4585	0.5	0.4769
J value		0	0.3162	0	0.1897	0	0.1265
KS index		1	1	1	1	1	1
Exec. time (hh:mm:ss)		00:06:53		00:05:56		00:05:40	

(1) RMS deviation within the transformed data.

(2) RMS deviation, comparing with actual (untransformed) current.

RT-E04: To implement the proposed algorithm for power conditioning application, the fundamental component of historical current data was filtered out by performing fast Fourier transform. Subtracting this component from the predicted current, unwanted harmonic components of the current could be obtained. By subtracting these harmonics from actual value of current, the wave shape of the load current can be made pure sinusoidal. Figure 4.4 shows the whole process in brief. Figures 4.5, 4.6, and 4.7 show the comparison of actual and corrected waveform for the arc furnace current in three phases.

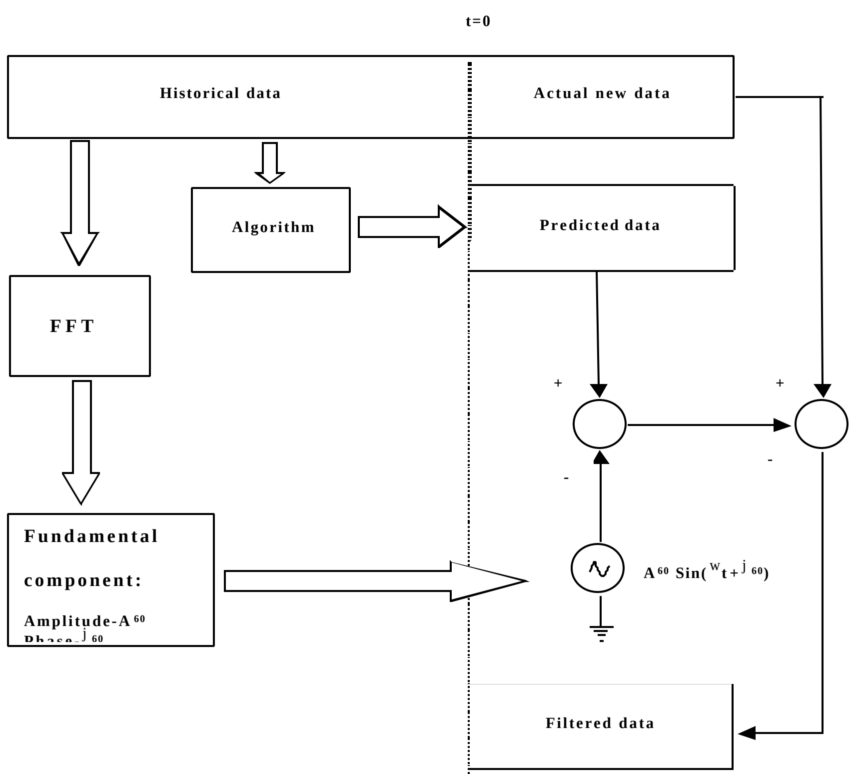


Figure 4.4 Process of power conditioning for the test *RT-E04*

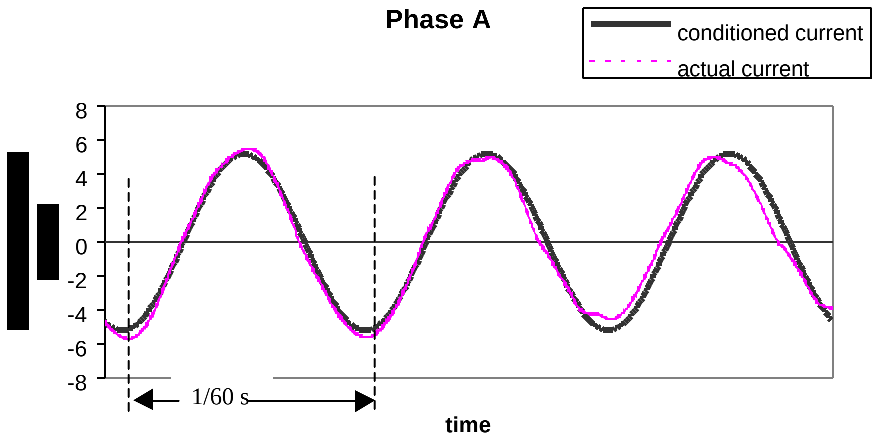


Figure 4.5 Comparison of actual and corrected current wave shape for phase-A

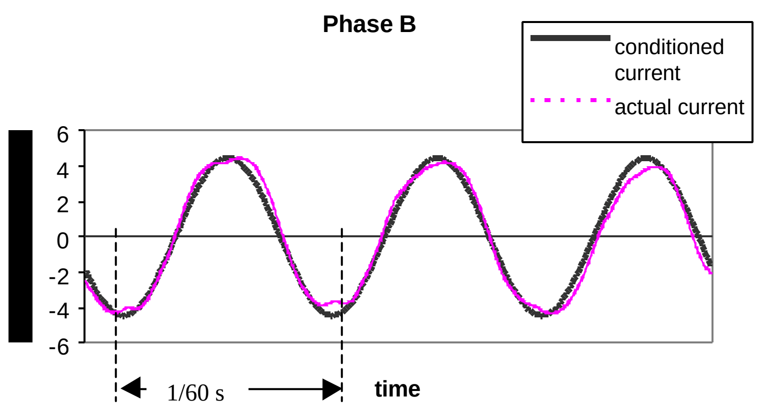


Figure 4.6 Comparison of actual and corrected current wave shape for phase-B

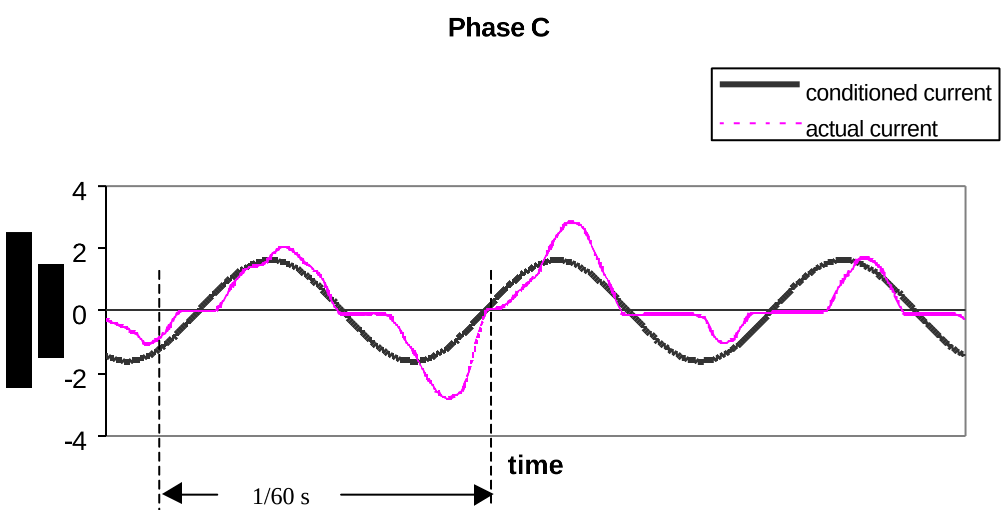


Figure 4.7 Comparison of actual and corrected current wave shape for phase-C.

4.5 Discussion

Load modeling: For load modeling applications, the proposed algorithm is executed offline, which means that the historical data set, and thus the Symbolic Dynamic dictionary is not updated once it is formed. From the point of view of comparison, the RMS value of the current is the most important parameter. *CSI* and *J* statistics are the measure of similarity between the two signals. In other words these two indices are related to the waveshape of the actual and generated signal. In load modeling, the waveshape of the predicted signal is of lesser importance. Load power depends mainly on the RMS value of the current.

Power conditioning: Power conditioning is an online real time application of the proposed algorithm. The historical data set and the Symbolic Dynamic dictionary are updated by including the actual current value each time a new value is predicted. By predicting the future value of current, it is possible to inject an out of phase current into the system and thus attenuating unwanted components. Therefore the wave shape of the forecasted signal is also as important as the RMS value. Being a real time process, execution time for the algorithm is another important parameter. Execution time can be minimized by reducing the number of cells and maximum word length for the dictionary. Thus compromising on the accuracy of the results obtained, the execution time can be improved. Table 4.11 shows the execution time for different combinations of word length and number of cells.

Inspection of Table 4.11, noting that the predicted values are separated in time by $?t$ = 0.5 ms, shows that only for trial 5 is the processing time less than the prediction interval.

Table 4.11 Execution time comparison

Trial	Maximum word length	Predicted points	Predicted interval (ms)	Number of cells	Average execution time per point (ms)
1	5	100	10	Variable (33)	9910
2	4	100	10	Variable (33)	5270
3	4	100	10	Variable (4)	160
4	4	200	20	Variable (5)	180
5	4	500	50	Variable (3)	46
6	4	500	50	Variable (7)	206
7	4	500	50	Variable (7)	208
8	4	500	50	10	826
9	4	500	50	10	712
10	4	500	50	10	680

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

This thesis proposed an innovative technique, called Symbolic Dynamics, for power system load modeling. Salient features of this approach can be listed as follows:

- The symbolic dynamic model is aimed at highly varying loads such as electric arc furnaces, steel rolling mills and other loads used in the steel industry where load current does not have a well-behaved waveshape. Symbolic Dynamics is best suited for cases in which a physical model is either inconvenient, inaccurate, inappropriate or unavailable.
- The method is best suited for loads which have a rich set of data but perhaps a poor model.
- Symbolic Dynamics utilizes only the time series historical data of voltage and/or current.
- Bus voltage and/or load current signals are descretized to represent them by *symbols*.
- States of a time varying signal can be represented by *words*, which are combinations of symbols.
- A *symbolic dynamic* dictionary, formed by collecting all the words, characterizes a complete signal.
- In this approach it is assumed that characteristics of the future value of load current replicate the historical data set in some manner.

- Implementing a probabilistic algorithm which utilizes the maximum information available from the historical data set, it is possible to estimate the future value of the load current.
- By predicting the future value of load current, it is possible to condition the actual load current and thus the model can be used as a power conditioner.
- The model can also be used to compare two signals of any type, periodic or aperiodic, and there is no dependence on sinusoidal waveshapes. Noise gives less trouble in this comparison than in traditional methods.
- The symbolic dynamic approach does not contain frequency domain or any other transform.
- The model is simple and contains no complicated formula or mathematical equation. Also, complex numbers are not used.
- The accuracy of a symbolic dynamic model is highly dependent on parameters such as word length, descretization details, and data sample length. However typical accuracy for an AC electric arc furnace is shown in Table 5.1.

Table 5.1 Error in the proposed model when tested for EAF data

Average % error in	Load Modeling	Power Conditioning
RMS load current	6.9	6.48
Average Power	7.62	
CSI		4.3
KS significance level		6.7

5.2 Recommendations

The following are some recommendations for the future work in this area:

- A multidimensional symbolic dynamic dictionary can be used to represent different parameters together. For example, a symbol in this new concept may represent the value of current, voltage, and frequency in one particular state. Another possibility is the use of derivative information in the symbolic dynamic dictionary.
- Frequency domain parameters might be added to the time domain analysis shown in this thesis. For example, some frequency components of load current may be used as model states.
- Validity of the proposed model can be checked by implementing it on the historical data set of other varying loads such as a steel rolling mill.
- Alternative methods of discretizing a signal can be considered to eliminate the difficulties associated with shifting and scaling the signal.
- State estimation techniques for the bad data rejection can be implemented while forecasting the future value of load current.
- The model can be modified to apply in load forecasting.
- To improve the performance of the model for the power conditioning application, coding of the model in some other programming languages should be considered.
- The model can be used in designing a microprocessor controlled active power filter.
- Because Symbolic Dynamics makes no use of a physical load model, and it does not utilize frequency domain information, consideration should be given to the

integration of physical information into the symbolic dynamic approach. This is represented in Figure 5.1.

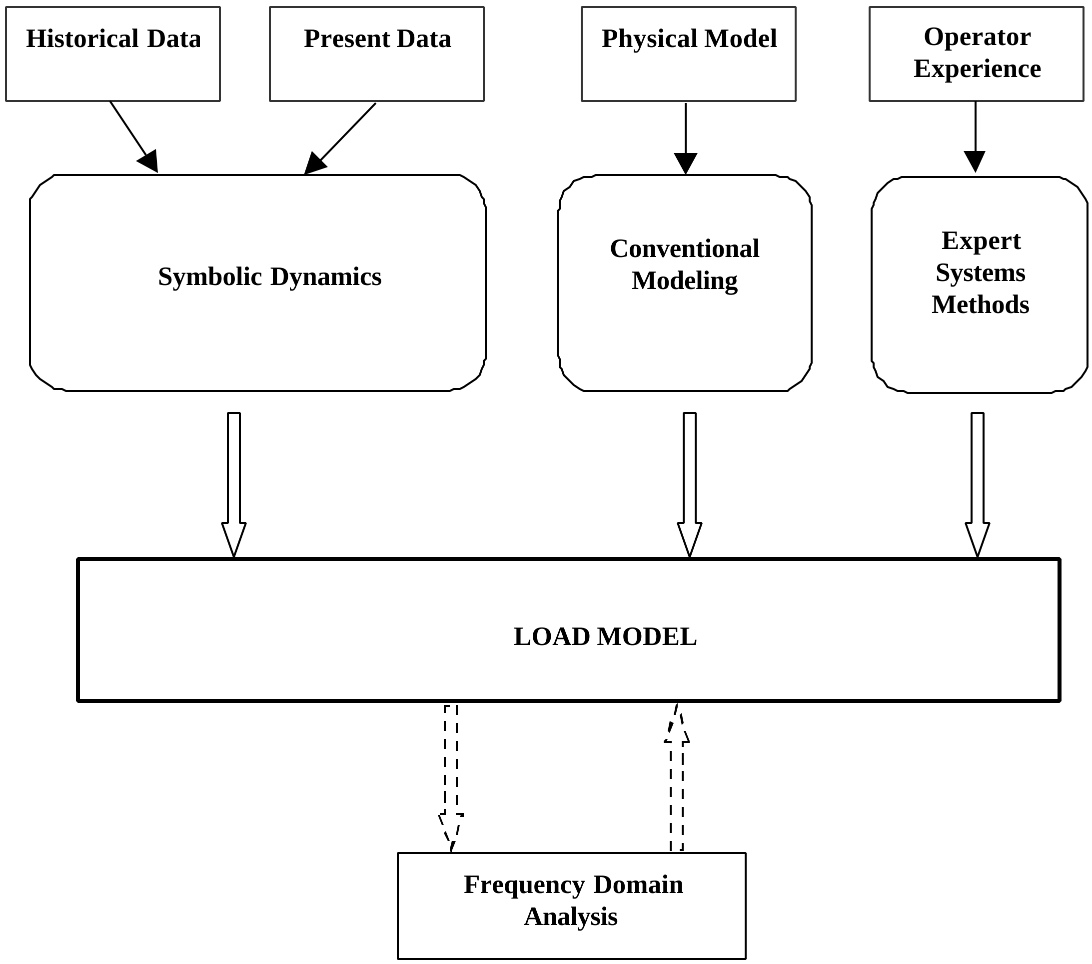


Figure 5.1 Pictorial representation of alternative approaches to power system load modeling

REFERENCES

[1] M. S. Chen, “Determining load characteristics for transient performances,” vol. 3: Procedure for modeling power system loads, *Electric Power Research Institute El-849, Project 849-3, final report*, Palo Alto, CA, May 1979.

[2] Y. Wang, N. C. Pahalawaththa, “Power system load modeling,” *Proceedings of International Conference on Power System Technology*, POWERCON '98, vol. 1 pp. 725 –729.

[3] Q. H. Wu, J. Y. Wen, “Measurement based power system load modeling by learning,” *UKACC International Conference on Control '98*, vol. 1, pp. 871 –876.

[4] P. Ju, E. Handschin, D. Karlsson, “Nonlinear dynamic load modeling: model and parameter estimation,” *IEEE Transactions on Power Systems*, vol. 11, issue 4, Nov. 1996, pp. 1689 –1697.

[5] Ma Da-Qiang, Ju Ping, “A novel approach to dynamic load modeling,” *IEEE Transactions on Power Systems*, vol. 4, no. 2, May 1989.

[6] D. N. Salem, “A Markov approach to power system load modeling,” Masters Thesis, Arizona State University, Tempe AZ, Dec. 2000.

[7] M. Takeda, T. Aritsuka, “Practical applications of active filters for power conditioning in distribution networks,” *Proceedings of 8th International Conference on Harmonics and Quality of Power*, 1998, vol. 1, pp. 304 –309.

[8] J. Nastran, R. Cajhen, M. Seliger, P. Jereb, “Active power filter for nonlinear AC loads,” *IEEE Transactions on Power Electronics*, vol. 9, issue 1, Jan. 1994, pp. 92 –96.

[9] M. Carpita, M. Marchesoni, "Experimental study of a power conditioning system using sliding mode control," *IEEE Transactions on Power Electronics*, vol. 11, issue 5, Sept. 1996, pp. 731 –742.

[10] H. S. Patel, R. G. Hoft, "Generalized techniques of harmonic elimination and voltage control in thyristor inverters: Part I- Harmonic elimination," *IEEE Transaction on Industrial Application*, vol. IA-9, no. 3, pp. 310-317, 1973.

[11] G. S. Buja, G. B. Indri, "Optimal pulsewidth modulation for feeding AC motors," *IEEE Transaction on Industrial Application*, vol. IA-13, no. 1, pp. 38-44, 1977.

[12] S. R. Bowes, "New sinusoidal pulse width-modulated inverter," *Proceedings of IEE*, vol. 122, no. 11, pp. 1279-1285, 1975.

[13] Hao Bai-lin, Elementary symbolic dynamics and chaos in dissipative systems, World Scientific Publishing Co., Singapore, 1989.

[14] Chai Wah Wu, L. O. Chua, **"**Symbolic dynamics of piecewise-linear maps," *IEEE Transactions on Circuits and Systems II: Analog and digital signal processing*, vol. 41, issue 6, June 1994, pp. 420 –424.

[15] P. Peleties, R. DeCarlo, **"**An example of switched system analysis via symbolic dynamics and Petri nets," *Proceedings of the 32nd IEEE Conference on Decision and Control*, 1993, vol. 1, pp. 300 –305.

[16] A. Azzouz, M. Hasler, 'Orbits of the R-L-diode circuit," *IEEE Transactions on Circuits and Systems*, vol. 37, issue 11, Nov. 1990, pp. 1330 –1338.

[17] D. L. Isaacson, R. W. Madsen, Markov chains theory and applications, John Wiley and Sons, Inc., New York, 1976.

[18] R. B. Fancher, A. E. Hulse, "Analyzing the value of equipment testing in a power distribution system," *IEEE Transmission and Distribution Conference*, 1999, vol. 2, pp. 592 –596.

[19] A. Csenki, "Reliability analysis of recovery blocks with nested clusters of failure points," *IEEE Transactions on Reliability*, vol. 42, issue 1, March 1993, pp. 34 –43.

[20] M. Hollander, D. A. Wolfe, <u>Nonparametric statistical methods</u>, John Wiley and Sons, Inc., New York,1999.

[21] N. T. Sindhushayana, B. Marcus, M. D. Trott, "Homogeneous shifts," *Proceedings of the 34th IEEE Conference on Decision and Control*, 1995, vol. 3, pp. 3265 –3270.

[22] N. Mohan, T. M. Undeland, W. P. Robbins, <u>Power electronics: converters, applications, and design,</u> John Wiley and Sons, Inc., New York, 1995.

[23] G. T. Heydt, <u>Electric power quality</u>, Stars in a Circle Publications, W. Lafayette, Indiana, 1991.

[24] Alistair Sinclair, <u>Algorithms for random generation and counting: A Markov chain approach</u>, Birkhäuser, Boston, 1993.

[25] Håkan Andersson, Tom Britton, <u>Stochastic epidemic models and their statistical analysis</u>, Springer-Verlag, New York, 2000.

[26] B. P. Kitchens, <u>Symbolic dynamics: one-sided, two-sided, and countable state Markov shifts</u>, Springer-Verlag, New York, 1998.

[27] D. Lind, B. Marcus, <u>An introduction to symbolic dynamics and coding</u>, Cambridge University Press, Cambridge, 1995.

[28] P. Kundur, <u>Power system stability and control</u>, McGraw-Hill, New York, 1993.

APPENDIX A

MATLAB CODE FOR THE PROPOSED WORK

A.1 Dictionary Formation

In this program, a time series signal was generated synthetically, storing some arbitrary integer values in a row vector x1. This row vector x1 is the input for forming a dictionary. A word length of 3 symbols was selected to form a dictionary in this code.

```
% Program to create a dictionary from a given time series signal x1
***********************************************************
clear
x1=[1,4,1,3,2,1,4,5,2,1,2,3,2,4,5]; % Synthetic time series %signal
x=x1'
%find signal length
nw=3;
nn=size(x);
n=nn(1,1);
%calculate dictionary of all words.  The dictionary is d
f=0;
%w=word length
for w=1:nw
   for s=1:n-w+1
      c=zeros(w,1);
      c(1:w)=x(s:s+w-1);
     f=f+1;
      d(f,1:w)=(c(1:w))';
   end
end
d;
f;
```

```
%count the occurrences of each word, call this kount
kount=ones(f,1);
%delete duplicate rows of dictionary and kount
r=1;
while r<=f-1
   rr=r+1;
   while rr<=f
            if abs(d(rr,:)-d(r,:))<0.0000001
                       d(rr,:)=[];
        kount(r)=kount(r)+1;
        kount(rr)=[];
        rr=rr-1;
        f=f-1;
     end
     rr=rr+1;
  end
  r=r+1;
end
%Print results, first nw cols is the word, last column
%is the kount
y=[d,kount]
s=sum(kount); % total occurance of words in x.
frac_occ=kount/s;
Dictn=[y,frac_occ];% first 3 col. are words 4th one is occurance,5th
one is fractional occurance of each word.
```

A.2 Comparing two dictionaries

Input data for the program are stored in two data files. One set of data belongs to the original current data and the other set is regenerated current. The program forms two dictionaries from these two data sets and compares them by calculating the common signal index (CSI).

```
% Program to compare two dictionaries X and Y and obtain a common
signal index (CSI)
************************************************************
%clear variables
clear
%Reading actual current data from a data file.
Points=2000;
fid=fopen('h:\exp.dat','rt');
if(fid==(-1))
   disp('Error in opening the given input data file');
end
n=1;
while n~=(Points+1)
   x1(n)=fscanf(fid,'%f',1);
   n=n+1;
end
% Reading regenerated current data from a data file.
Points=2000;
fid=fopen('h:\tri.dat','rt');
if(fid==(-1))
   disp('Error in opening the given input data file');
end
n=1;
while n~=(Points+1)
   y1(n)=fscanf(fid,'%f',1);
   n=n+1;
```

```
end
%Calculation of scaling function
N=8;
minx=min(x1);
maxx=max(x1);
mat1=[minx 1;maxx 1];
mat2=[1.01;N+0.99];
mat3=inv(mat1)*mat2;
A=mat3(1);
B=mat3(2);
miny=min(y1);
maxy=max(y1);
mat11=[miny 1;maxy 1];
mat22=[1.01;N+0.99];
mat33=inv(mat1)*mat2;
C=mat33(1);
D=mat33(2);
   x=fix(A*x1'+B); % Scaling function.
   y=fix(C*y1'+D); % Scaling function.
   %set max word length
nw=3;
%find signal length
nn=size(x);
n=nn(1,1);
%calculate dictionary d for original signal.
f=0;
%w=word length
for w=1:nw
   for s=1:n-w+1
```

```
      c=zeros(w,1);
      c(1:w)=x(s:s+w-1);
     f=f+1;
      d(f,1:w)=(c(1:w))';
   end
end
d;
f;
%count the occurrences of each word, call this kount
kount=ones(f,1);
%delete duplicate rows of dictionary and kount
r=1;
while r<=f-1
   rr=r+1;
   while rr<=f
            if abs(d(rr,:)-d(r,:))<0.0000001
                       d(rr,:)=[];
        kount(r)=kount(r)+1;
        kount(rr)=[];
        rr=rr-1;
        f=f-1;
     end
     rr=rr+1;
  end
  r=r+1;
end
%Print results, first nw columns are the word, last column
%is the kount
[d,kount];
```

```
s=sum(kount); % total occurance of words in x.

%Y Dictionary (For regenerated data set)

%set max word length

nw1=nw;

%find signal length

nn1=size(y);

n1=nn1(1,1);

%calculate dictionary of all words.  The dictionary is d

f1=0;

%w=word length

for w1=1:nw1

   for s1=1:n1-w1+1

      c1=zeros(w1,1);

      c1(1:w1)=y(s1:s1+w1-1);

     f1=f1+1;

      d1(f1,1:w1)=(c1(1:w1))';

   end

end

d1;

f1;

%count the occurrences of each word, call this kount

kount1=ones(f1,1);

%delete duplicate rows of dictionary and kount

r1=1;

while r1<=f1-1

   rr1=r1+1;

   while rr1<=f1

            if abs(d1(rr1,:)-d1(r1,:))<0.0000001

        d1(rr1,:)=[];
```

```
        kount1(r1)=kount1(r1)+1;
        kount1(rr1)=[];
        rr1=rr1-1;
        f1=f1-1;
     end
     rr1=rr1+1;
  end
  r1=r1+1;
end
%Print results, first nw columns are the word, last column
%is the kount
[d1,kount1];
s1=sum(kount1); % total occurance of words in Y.
%comparison of X & Y
      k=zeros(f1,1);
      frx=zeros(f1,1);
      fry=zeros(f1,1);
for r2=1:f1
   for rr2=1:f
      if abs(d1(r2,:)-d(rr2,:))<0.000001
         k(r2)=1; % k(n) is 1 if nth signal in Y is also %present in X.
If not, k(n) is 0.
        frx(r2)=kount(rr2);
         fry(r2)=kount1(r2);
      end
   end
end
frx1=frx/s;
fry1=fry/s1;
```

```
fr1=frx1+fry1;
fr3=frx1.*fry1;
fr1=fr1+(fr1==0)*eps;
c1=fr3./fr1; % common signal index for each word.
z=[d1,k,frx,fry];
p=1;
while p<=f1
      if z(p,nw)==0 %remove the rows which are not common in both X & Y
   z(p,:)=[];
   f1=f1-1;
   p=p-1;
end
p=p+1;
end
csi1=sum(c1) % common signal index for comparing the two %signals.
```

A.3 Forecasting a signal using Model-1

Input to this program is fifty data points on a sinusoidal signal, contaminated with third and fifth harmonics. With this data set, a dictionary is formed and the signal is regenerated using model 1.

```
% A program to forecast a signal using Model 1.
***********************************************************
%clear variables
clear
% Form a harmonic signal synthetically by taking 50 data %points.
x1=linspace(0,50,51);
```

```
y1=sin(x1*pi/12.5)+(1/3)*sin(3*x1*pi/12.5)+(1/5)*sin(5*x1*pi/12.5);
y=y1'+2.0
 %set max word length
nw=5;
%find signal length
nn=size(y);
n=nn(1,1);
% clear final word length
fwl=zeros(50,1);
%calculate dictionary of all words.  The dictionary is d
f=0;
%w=word length
for w=2:nw
   for s=1:n-w+1
      c=zeros(w,1);
      c(1:w)=y(s:s+w-1);
     f=f+1;
      d(f,1:w)=(c(1:w))';
   end
end
%count the occurrences of each word, call this kount
kount=ones(f,1);
%zero duplicate rows of dictionary and kount
for r=1:f-1
   for rr=r+1:f
      if (r <= f) & (rr <= f) & (d(r,:)==d(rr,:))
                d(rr,:)=zeros(1,nw);
        kount(r)=kount(r)+1;
        kount(rr)=0;
```

```
     end
   end
end
%remove zero rows from d and unused rows from kount
fff=f;
for r=1:f
   while (r <= fff) & (max(abs(d(r,:))) <= 0.001)
      fff=fff-1;
      d(r,:)=[];
      kount(r)=[];
   end
end
s=sum(kount); % total occurance of words in x.
fr=kount/s;
wd=d ;  % working dictionary
wf=fr;
[wd,wf];
kstop=50;
k=1;
while (k<=kstop)
   cwf=cumsum(wf(:,1)); % cumulative fractional occurrence.
   sz=size(wd);
   ff=sz(1);
r=rand;
if (r<=cwf(1)) % If random no. is less than any of the cwf %entry,
select first word as the seed word.
   seedwd=wd(1,:);
else
for i=1:ff-1
```

```
   if (r>=cwf(i))&(r<cwf(i+1)) % If random no. lies between % ith and
(i+1)th cwf entry, select (i+1)th word as a seed % word.
      seedwd=wd(i+1,:);
      break;
      end
          end
       end
      seedword(k,:)=seedwd;
lastdig=seedwd(nw);     % checking the last digit of the seed %word
t=0;
while (lastdig==0)  % select a nonzero last digit
   lastdig=seedwd(nw-1);
   nw=nw-1;
   t=t+1;
end
fwl(k)=nw;
nw=nw+(1*t);
lastdig;
j=1;
mini_fr=[];
mini_d=[];
for i=1:fff
   if(d(i,1)==lastdig) % words beginning with the last
% digit
      mini_fr(j,:)=fr(i,:);
      mini_d(j,:)=d(i,:);
      j=j+1;
   end
end
```

```
if isempty(mini_d)==1   %If no word begins with the %lastdigit start
from the beginning.
   mini_d=d;
   mini_fr=fr;
end
  mini_d;  % words beginning with the last digit.
  mini_fr;    % fractional occurances of the all words %beginning with
the last digit.
sumf=sum(mini_fr(:,1));
mod_fr=mini_fr/sumf;
wd=mini_d;
wf=mod_fr;
k=k+1;
end
seedword;
fwl;
result1=nonzeros(seedword(1,:));
result2=nonzeros(seedword(2:kstop,2:nw)');
result=[result1',result2']' % Forecasted signal in the form % of a
column matrix.
```

A.4 Forecasting a signal using Model 2, case 1 (Load Modeling)

Input to this program is historically recorded current data of electric arc furnace. Future value of current is predicted using model 2, without updating the historical data set. The code is used for load modeling applications. 5000 data of EAF current are stored in a data file out of which 3000 data are used as a historical data set and 2000 data are predicted and compared with the rest of the data.

```
% A program to forecast a signal for load modeling  application
**********************************************************
clear
% Reading data from the file.
Points=5000;
fid=fopen('h:\Ib.dat','rt');
if(fid==(-1))
   disp('Error in opening the given input data file');
end
n=1;
while n~=(Points+1)
a(n)=fscanf(fid,'%f',1);
   n=n+1;
end
% Calculation of the scaling function
N=10;
minx=min(a);
maxx=max(a);
mat1=[minx 1;maxx 1];
mat2=[1.01;N+0.99];
mat3=inv(mat1)*mat2;
A=mat3(1);
B=mat3(2);
aa1=fix(A*a+B);
x=aa1(:,1:3000);  % Historical data set
I_hist=x'-B;  %  Historical  data  set,  without  any  shift,  for  %  the
purpose of RMS value comparison.
I=aa1';
```

```
x11=x';
nn=size(x11);
n=nn(1,1);
nw=5;
Iact=I(3001:5000)-B % Expected value of future current,
% without any shift, for the purpose of RMS value comparison.
%calculate dictionary of all words.  The dictionary is d
f=0;
%w=word length
for w=1:nw
   for s=1:n-w+1
      c=zeros(w,1);
      c(1:w)=x11(s:s+w-1);
     f=f+1;
      d(f,1:w)=(c(1:w))';
   end
end
% Count the occurrence of each word, call this kount
kount=ones(f,1);
% delete duplicate rows of dictionary and kount
r=1;
while r<=f-1
   rr=r+1;
   while rr<=f
      if abs(d(r,:)-d(rr,:))<0.000001
        d(rr,:)=[];
        kount(r)=kount(r)+1;
        kount(rr)=[];
        rr=rr-1;
```

```
        f=f-1;
     end
     rr=rr+1;
  end
  r=r+1;
end
% Print results, first nw columns is the word, last column
% is the kount
y=[d,kount];
s=sum(kount); % total occurrence of words in x.
frac_occ=kount/s;
Dictn=[y,frac_occ];     % first nw columns are words, next
% column is occurrence, next one is fractional occurrence
% of each word.
p=1;
while(p<=2000)    % Predicting 2000 current data points.
ss=size(x);
temp=ss(1,2);
minx=[];
k=1;
for j=1:f
      nx=ss(1,2); % Last symbol of historical data set.
   i=nw-1;
   ii=0;
tt=0;
% Checking the sequence of symbols.
while i>=1
   if(d(j,i+1)~=0)
         if abs(x(nx)-d(j,i))<0.000001
```

```
            tt=1;
            nx=nx-1;
            i=i-1;
            ii=ii+1;
                   if(i==0)
            nx=ss(1,2);
         end
         if((nx<=temp-1)&(i<=nw-2)&(x(nx+1)~=d(j,i+1)))
              tt=0;
              end
        else
           i=i-1;
           ii=0;
          nx=ss(1,2);
       end
   else
      i=i-1;
      end
   end   % end of while
   if((tt~=0)&(ii>=1))
      minx(k,:)=d(j,:);
      minfr(k,:)=frac_occ(j,:);
      len(k,:)=ii;    % length of a sequence, found in the % historical
data.
      k=k+1;
   end
end      % end of for
kk=1;
if(isempty(minx))
```

```
   for jj=1:f
      if(d(jj,2:nw)==0)
         minx(kk,:)=d(jj,:);
         minfr(kk,:)=frac_occ(jj,:);
         len(kk,:)=1;
      end
      kk=kk+1;
      end
            end
min=[minx,minfr,len];  %  Mini  dictionary  of  all  the  matching  %
symbols/sequences
wtfr=minfr.*(len.^2); % Weight the fractional occurrence of % each word
above by the square of its length.
sm=sum(wtfr(:,1));
newfr=wtfr/sm;
cfr=cumsum(newfr);
sz=size(minx);
f1=sz(1);
r=rand;
if (r<=cfr(1))% Select a new word from the mini dictionary.
   newwd=minx(1,:);
   wdlen=len(1,:);
else
for i=1:f1-1
   if (r>=cfr(i))&(r<cfr(i+1))
      newwd=minx(i+1,:);
      wdlen=len(i+1,:);
      break;
      end
```

```
         end
      end
     newwd;
      wdlen;
      mm=wdlen+1;
      emp=any(newwd(:,2:nw));
      if(emp==0)
         newI(p)=newwd(1);
      else
         newI(p)=newwd(mm);
         end %end of if-else
             x=[x,newI(p)];
% New value of current, newI is the last symbol of the new % word,
selected from the mini dictionary.
clear ss k temp nx minx minfr len wdlen min wtfr sm newfr cfr sz newwd
      p=p+1;
   end % end of while
   Ipred=newI'-B % predicted current data set
   Iact=I(3001:5000)-B; % expected current
   RMSpred=sqrt(mean(Ipred.^2)) %RMS of predicted data
   Avgpred=mean(Ipred) %Average value of predicted data
   RMSact=sqrt(mean(Iact.^2))%RMS value of actual data
   Avgact=mean(Iact) % Average value of actual data
   Error_act=((RMSpred-RMSact)/RMSact)*100
  RMShist=sqrt(mean(I_hist.^2)) %RMS of historical data
  Error_hist=((RMSpred-RMShist)/RMShist)*100
  Avghist=mean(I_hist) % Average value of historical data
```

A.5 Forecasting a signal using Model 2, case 2 (Power conditioning)

This model is used for power conditioning application. The historical data set, in this code, is updated each time a new current value is predicted. Predicted current is conditioned by injecting appropriate current as described in Figure (4.4).

```
***********************************************************
clear
% Reading the data from the file.
Points=600;
fid=fopen('h:\Ia2.dat','rt');
if(fid==(-1))
   disp('Error in opening the given input data file');
end
n=1;
while n~=(Points+1)
a(n)=fscanf(fid,'%f',1);
n=n+1;
end
% Calculation of scaling function.
N=10;
minx=min(a);
maxx=max(a);
mat1=[minx 1;maxx 1];
mat2=[1.01;N+0.99];
mat3=inv(mat1)*mat2;
A=mat3(1);
B=mat3(2);
aa1=fix(A*a+B);
```

```
x=aa1(:,1:100);

I_hist=x'-B; % Historical data set, without any shift, for % the

purpose of RMS value comparison.

I=aa1';

for t=1:100

II(t)=1*sin(2*pi*60*0.0001*t);

end

% II is a synthetic current source with unity amplitude and % points

separated by the same sampling frequency as in

% historical data set.

IFFT=fft(II); % FFT of the above current signal.

fac=abs(IFFT(2))% Finding a factor to obtain the true FFT.

ph1=angle(IFFT(2)) % Phase angle of the fundamental.

hisFFT=fft(a(1:100)); % FFT of the historical data set.

M=fac*abs(hisFFT(2)) % True magnitude of the fundamental

% component of the historical data set.

ph=angle(hisFFT(2)) % Phase angle of the fundamental

% component of the historical data set.

x11=x';  % Historical data set.

nn=size(x11);

n=nn(1,1);

nw=5;

%calculate dictionary of all words.  The dictionary is d

f=0;

%w=word length

for w=1:nw

   for s=1:n-w+1

      c=zeros(w,1);

      c(1:w)=x11(s:s+w-1);
```

```
     f=f+1;
      d(f,1:w)=(c(1:w))';
   end
end
%count the occurrences of each word, call this kount
kount=ones(f,1);
% Forecasting the future value of current.
p=1;
while(p<=500)
nn=size(x);
n=nn(1,2);
s=n+1;
f1=0;
for w=1:nw
         c1=zeros(w,1);
         s=s-1;
      c1(1:w)=x11(s:s+w-1);
     f1=f1+1;
d1(f1,1:w)=(c1(1:w))';
   end
%count the occurrences of each word, call this kount
kount1=ones(f1,1);
% updated dictionary
d=[d;d1];
kount=[kount;kount1];
%delete duplicate rows of dictionary and kount
r=1;
f=f+f1;
while r<=f-1
```

```
   rr=r+1;
   while rr<=f
      if abs(d(rr,:)-d(r,:))<0.0000001
                d(rr,:)=[];
        kount(r)=kount(r)+1;
        kount(rr)=[];
        rr=rr-1;
        f=f-1;
     end
     rr=rr+1;
  end
  r=r+1;
end
%Print results, first nw cols is the word, last col
%is the kount
y=[d,kount];
sm1=sum(kount); % total occurrence of words in x.
frac_occ=kount/sm1;
Dictn=[y,frac_occ];% first 3 col. are words 4th one is %occurance,5th
one is fractional occurrence of each word.
d;
minx=[];
k=1;
for j=1:f
   nx=n;
   i=nw-1;
   ii=0;
tt=0;
while i>=1
```

```
   if(d(j,i+1)~=0)
         if(x(nx)==d(j,i))
           tt=1;
                    nx=nx-1;
         i=i-1;
         ii=ii+1;
                  if(i==0)
                     nx=n;
                  end
        if((nx<=n-1)&(i<=nw-2)&(x(nx+1)~=d(j,i+1)))             tt=0;
             end
       else
          i=i-1;
          ii=0;
         nx=n;
              end
   else
      i=i-1;
          end
   end   %end of while
   if((tt~=0)&(ii>=1))
      minx(k,:)=d(j,:);
      minfr(k,:)=frac_occ(j,:);
      len(k,:)=ii;  % length of a sequence, found in the
 % historical data.
      k=k+1;
    end
end
kk=1;
```

```
if(isempty(minx))
   for jj=1:f
      if(d(jj,2:nw)==0)
         minx(kk,:)=d(jj,:);
         minfr(kk,:)=frac_occ(jj,:);
         len(kk,:)=1;
      end
      kk=kk+1;
      end
               end
min=[minx,minfr,len];  %  Mini  dictionary  of  all  the  matching  %
symbols/sequences
wtfr=minfr.*(len.^2);% weighted fractional occurrence
sm=sum(wtfr(:,1));
newfr=wtfr/sm; % modifying wtfr so that it add to 1.
cfr=cumsum(newfr); % cumulative fractional occurrence
sz=size(minx);
   f2=sz(1);
r=rand;
if (r<=cfr(1))% Select a new word from the mini dictionary.
   newwd=minx(1,:);
   wdlen=len(1,:);
else
for i=1:f2-1
   if (r>=cfr(i))&(r<cfr(i+1))
      newwd=minx(i+1,:);
       wdlen=len(i+1,:);
      break;
      end
```

```
          end
       end
       newwd;
       wdlen;
       mm=wdlen+1;
       emp=any(newwd(:,2:nw));
       if(emp==0)
          newI(p)=newwd(1);
       else
          newI(p)=newwd(mm);
      end
% New value of current, newI is the last symbol of the new % word,
selected from the mini dictionary.
          sinus(p)=M*sin((2*pi*60*0.0001*p)+ph);
% Sinus is a sinusoidal source formed by taking the
% fundamental component of the historical data set.
          junk(p)=((newI(p)-B)/A)-sinus(p); % amount of
% unwanted components in the new current value.
          good(p)=a(100+p)-junk(p); % injecting the
% unwanted components in phase opposition, into the actual % future
current.
           x=[x,I(100+p)];
       x11=[x11;I(100+p)];
       newI(p);
       clear ss k n nx minx minfr len min wtfr sm sm1 newfr cfr sz
newwd d1 f2 kount1 frac_occ
    p=p+1;
    end % end of while
    filt1=good'/1000;
```

```
 Iact=newI'-B; % actual current
 Iex=a(101:600)'    %expected current
 RMSex=sqrt(mean(Iex.^2));
 RMSfilt1=sqrt(mean(filt1.^2));
 filt=filt1*(RMSex/RMSfilt1) % conditioned current.
plot(1:length(Iex),Iex,'g-.',1:length(filt),filt,'r-')
RMSact=sqrt(mean(Iact.^2))/10
RMSex=sqrt(mean(Iex.^2))
Error=((RMSact-RMSex)/RMSex)*100
```

APPENDIX B

ILLUSTRATIVE EXAMPLE OF KOLMOGOROV-SMIRNOV TEST

B.1 An illustrative example of Kolmogorov-Smirnov test

In this appendix, the Kolmogorov-Smirnov test for the comparison of two time series is given. Reference [20] gives all the details of the theory of this test.

Suppose X and Y are two sets of data shown in Table B.1. The two data sets are distribution free, which means no other information is available about the data. The aim is to check the degree of similarity between the two data sets using KS test.

Table B.1 Data sets under consideration

X	Y
1	1
5	1
7	1
0	2
2	8
3	3
1	4
2	8
5	9
8	1

B.2 Solution of the example

The following applies to the foregoing example:

Number of elements in $X = m = 10$

Number of elements in $Y = n = 10$

Greatest common divisor of m and $n = d = 10$

Z can be defined as *Z(i) = Z(i+1)* for ordered value of combined population of *X* and *Y*. Parameters *Z(i)*, $F_{10}(Z(i))$, $G_{10}(Z(i))$, and ½$F_{10}(Z(i))$-$G_{10}(Z(i))$ ½ are calculated in Table B.2

Table B.2 Calculation of functions for KS test

i	*Z(i)*	$F_{10}(Z(i))$	$G_{10}(Z(i))$	½$F_{10}(Z(i))$-$G_{10}(Z(i))$ ½
1	0	1/10	0/10	1/10
2	1	3/10	4/10	1/10
3	1	3/10	4/10	1/10
4	1	3/10	4/10	1/10
5	1	3/10	4/10	1/10
6	1	3/10	4/10	1/10
7	1	3/10	4/10	1/10
8	2	5/10	5/10	0
9	2	5/10	5/10	0
10	2	5/10	5/10	0
11	3	6/10	6/10	0
12	3	6/10	6/10	0
13	4	6/10	7/10	1/10
14	5	8/10	7/10	1/10
15	5	8/10	7/10	1/10
16	7	9/10	7/10	2/10
17	8	10/10	9/10	1/10
18	8	10/10	9/10	1/10
19	8	10/10	9/10	1/10
20	9	10/10	10/10	0

The KS test proceeds as,

$$J = (mn/d) * \max_{(-\infty<t<\infty)} \{|F_m(t) - G_n(t)|\}$$

= (10x10/10) * (2/10)

= 2

$$J^* = (mn/N)^{1/2} * \max_{i=1,2...N} \{|F_m(Z(i)) - G_n(Z(i))|\} = d/(mnN)^{1/2} * J$$

= 10/√(10x10x20) * 2 ($N = m+n = 20$)

= 0.4472

The value of a obtained from the table A.11 in [20] (repeated as Table B.3 here) is,

$$a = Q(0.447) = 0.9882.$$

Which means the lowest significance level for which the two data sets are not identical, is 98.82%. Thus the two populations have a very high degree of similarity.

Table B.3 Value of α for the KS test (taken from [20])

For a given s, the table entry is

$$Q(s) = 1 - \sum_{k=-\infty}^{\infty} (-1)^k e^{-2k^2 s^2} = P_0(J^* \geq s)$$

s	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.3	1.0000	1.0000	1.0000	.9999	.9998	.9997	.9995	.9992	.9987	.9981
0.4	.9972	.9960	.9945	.9926	.9903	.9874	.9840	.9800	.9753	.9700
0.5	.9639	.9572	.9497	.9415	.9325	.9228	.9124	.9013	.8896	.8772
0.6	.8643	.8508	.8368	.8222	.8073	.7920	.7764	.7604	.7442	.7278
0.7	.7112	.6945	.6777	.6609	.6440	.6272	.6104	.5936	.5770	.5605
0.8	.5441	.5280	.5120	.4962	.4806	.4653	.4503	.4355	.4209	.4067
0.9	.3927	.3791	.3657	.3527	.3399	.3275	.3154	.3036	.2921	.2809
1.0	.2700	.2594	.2492	.2392	.2296	.2202	.2110	.2024	.1939	.1857
1.1	.1777	.1701	.1626	.1555	.1486	.1420	.1356	.1294	.1235	.1177
1.2	.1123	.1070	.1019	.0970	.0924	.0879	.0836	.0794	.0755	.0717
1.3	.0681	.0646	.0613	.0582	.0551	.0522	.0495	.0469	.0444	.0420
1.4	.0397	.0375	.0354	.0335	.0316	.0298	.0288	.0266	.0250	.0236
1.5	.0222	.0209	.0197	.0185	.0174	.0164	.0154	.0145	.0136	.0127
1.6	.0120	.0112	.0105	.0098	.0092	.086	.0081	.0076	.0071	.0066
1.7	.0062	.0058	.0054	.0050	.0047	.0044	.0041	.0038	.0035	.0033
1.8	.0031	.0029	.0027	.025	.0023	.0021	.0020	.0019	.017	.0016
1.9	.0015	.0014	.0013	.0012	.0011	.0010	.0009	.0009	.0008	.0007
2.0	.0007	.0006	.0006	.005	.0005	.0004	.0004	.0004	.0004	.0003
2.1	.0003	.0003	.0003	.0002	.0002	.0002	.0002	.0002	.0001	.0001
2.2	.0001	.0001	.0001	.0001	.0001	.0001	.0001	.0001	.0001	.0001
2.3	.0001	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000

APPENDIX C

EAF AND REAL DATA RECORDING

C.1 Real data used in the proposed work

In Chapter 4, real EAF data were used to test the model. The real data were obtained from an electric arc furnace. These data are the recorded values of per phase voltage and current with a sampling frequency of 10 kHz. Table C.1 shows the sample values of EAF voltage and current for all the phases. An overall CT ratio of 400 and PT ratio of 8750 are utilized to obtain the actual current and voltage values respectively. Figures C.1, C.2, and C.3 show 60 cycles of EAF phase A, phase B, and phase C current respectively. These current values represent secondary side CT current in amps. A schematic of the arc furnace is shown in Figure C.4.

(Time is shown in ‘points’. One point = 0.1 ms)

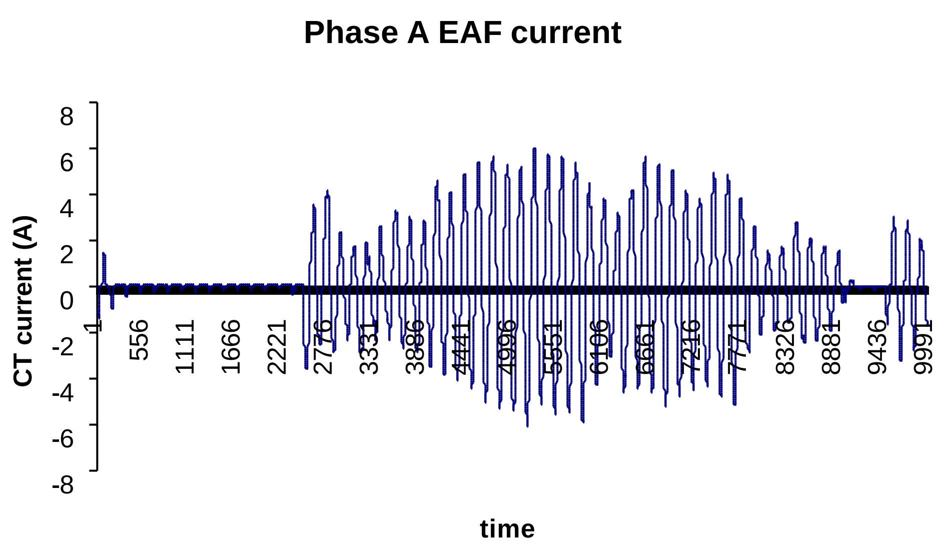
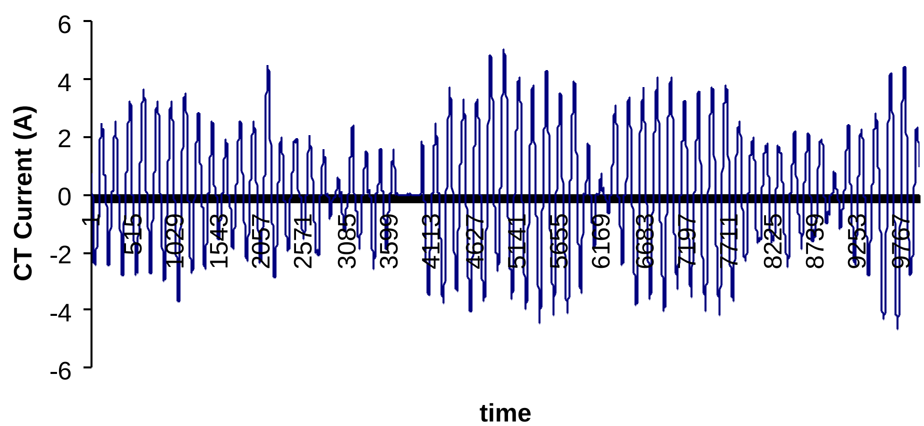


Figure C.1 A sample of EAF current for phase A

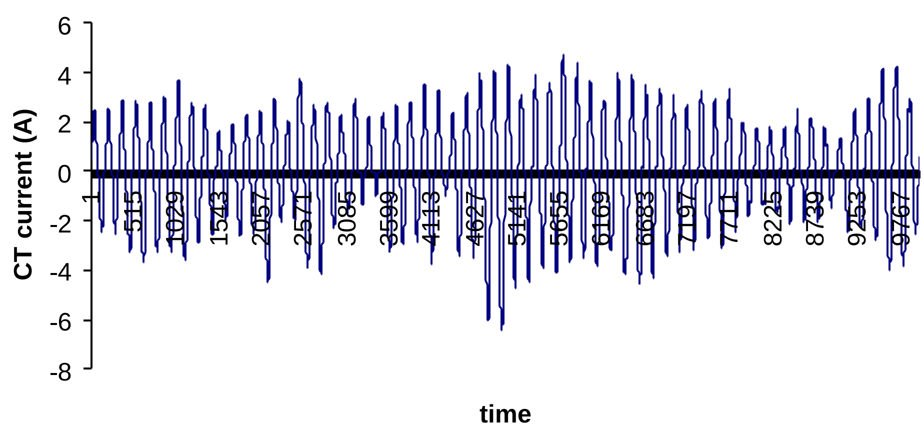
Phase B EAF current



(Time is shown in ‘points’. One point = 0.1 ms)

Figure C.2 A sample of EAF current for phase B

Phase C EAF current



(Time is shown in ‘points’. One point = 0.1 ms)

Figure C.3 A sample of EAF current for phase C

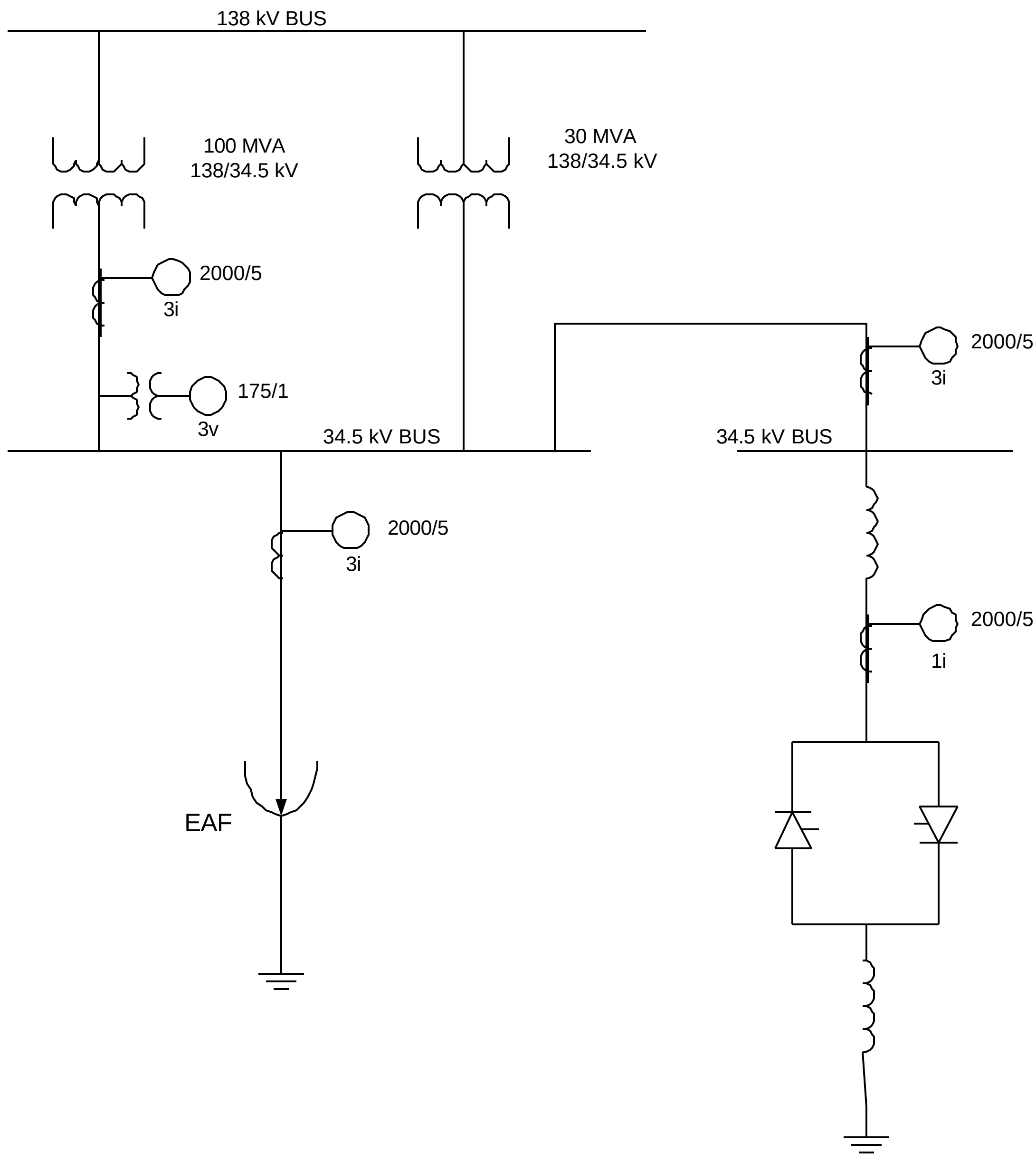


Figure C.4 Electric arc furnace schematic

Table C.1 Sample data sheet for 3-phase EAF current and voltage

I_A (A)	I_B (A)	I_C (A)	V_{AN} (V)	V_{BN} (V)	V_{CN} (V)
-1.118	0.669	0.483	-2.944	0.352	2.632
-1.104	0.601	0.542	-2.896	0.239	2.7
-1.099	0.527	0.61	-2.842	0.122	2.759
-1.089	0.444	0.688	-2.783	0.015	2.812
-1.079	0.356	0.771	-2.725	-0.103	2.866
-1.06	0.264	0.845	-2.666	-0.22	2.915
-1.05	0.181	0.903	-2.603	-0.337	2.969
-1.05	0.151	0.942	-2.524	-0.474	3.022
-1.079	0.161	0.952	-2.441	-0.596	3.071
-1.108	0.2	0.952	-2.368	-0.713	3.11
-1.147	0.239	0.947	-2.28	-0.835	3.14
-1.196	0.293	0.942	-2.188	-0.957	3.169
-1.26	0.356	0.947	-2.095	-1.074	3.198
-1.333	0.405	0.952	-1.978	-1.191	3.193
-1.387	0.444	0.962	-1.841	-1.304	3.169
-1.406	0.469	0.962	-1.733	-1.416	3.169
-1.401	0.449	0.991	-1.631	-1.509	3.164
-1.362	0.342	1.064	-1.523	-1.606	3.149
-1.294	0.2	1.128	-1.411	-1.709	3.14
-1.206	0.054	1.182	-1.294	-1.812	3.13
-1.089	-0.112	1.235	-1.177	-1.904	3.101
-0.962	-0.298	1.289	-1.06	-2.021	3.096
-0.815	-0.493	1.323	-0.938	-2.163	3.11
-0.649	-0.688	1.362	-0.811	-2.261	3.091
-0.483	-0.874	1.401	-0.684	-2.339	3.032
-0.337	-1.06	1.431	-0.552	-2.417	2.983
-0.225	-1.23	1.494	-0.42	-2.49	2.925
-0.132	-1.392	1.567	-0.293	-2.563	2.866
-0.059	-1.538	1.646	-0.166	-2.627	2.808
-0.015	-1.675	1.738	-0.039	-2.69	2.744
0.005	-1.792	1.841	0.088	-2.749	2.681
0.015	-1.899	1.938	0.205	-2.808	2.612
0.024	-1.997	2.021	0.322	-2.856	2.549
0.029	-2.075	2.1	0.444	-2.905	2.48
0.029	-2.139	2.168	0.552	-2.954	2.412
0.029	-2.192	2.222	0.664	-2.998	2.344
0.024	-2.236	2.266	0.776	-3.032	2.271
0.024	-2.271	2.3	0.889	-3.071	2.192
0.024	-2.29	2.319	1.001	-3.11	2.114
0.015	-2.3	2.334	1.108	-3.135	2.036
0.01	-2.329	2.368	1.221	-3.159	1.948
CT ratio - 1:400			PT ratio - 1:87500		