4-7-2016

Mining Dynamic Recurrences in Nonlinear and Nonstationary Systems for Feature Extraction, Process Monitoring and Fault Diagnosis

Yun Chen

Mining Dynamic Recurrences in Nonlinear and Nonstationary Systems

for Feature Extraction, Process Monitoring and Fault Diagnosis

by

Yun Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Co-Major Professor: Hui Yang, Ph.D.
Co-Major Professor: Alex Savachkin, Ph.D.
Ali Yalcin, Ph.D.
Yicheng Tu, Ph.D.
Kandethody M. Ramachandran, Ph.D.

Date of Approval:
April 5, 2016

Keywords: Sensor Network, Multiscale Analysis, Spatiotemporal Model, Computer Simulation and Optimization, Healthcare Informatics

DEDICATION

The dissertation is dedicated to my family, especially my parents, Jinjie Chen and Muxia Hu, who have been so supportive and considerate over the years.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to Dr. Hui Yang for his continuous support and guidance during this process. I have enjoyed working with him and learning from his experiences. I really appreciate him for providing me the opportunity to work in the Complex Systems Monitoring, Modeling and Analysis Laboratory and introducing this exciting research field to me.

Secondly, I am extremely thankful to the members of my committee, Dr. Alex Savachkin, Dr. Ali Yalcin, Dr. Yicheng Tu and Dr. Kandethody M. Ramachandran. Deep thanks go to them for the constructive suggestions they proposed for my dissertation and the valuable advices for my future.

Thirdly, my deep gratitude also goes to Gloria Hanshaw, Elizabeth Conrad, Catherine Burton and Norma Paz for their consistent help and support in my graduate study. And thank my colleagues and friends for their help in my study and life.

I gratefully acknowledge the financial support from NSF (CMMI-1454012, CMMI-1266331, CMMI-1617148, CMMI-1619648, IIP-1447289 and IOS-1146882).

Finally, I also would like to thank my family for their support during these years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Real-time sensing brings the proliferation of big data that contains rich information of complex systems. It is well known that real-world systems show high levels of nonlinear and nonstationary behaviors in the presence of extraneous noise. This brings significant challenges for human experts to visually inspect the integrity and performance of complex systems from the collected data. My research goal is to develop innovative methodologies for modeling and optimizing complex systems, and create enabling technologies for real-world applications. Specifically, my research focuses on *Mining Dynamic Recurrences in Nonlinear and Nonstationary Systems for Feature Extraction, Process Monitoring and Fault Diagnosis.* This research will enable and assist in (i) sensor-driven modeling, monitoring and optimization of complex systems; (ii) integrating product design with system design of nonlinear dynamic processes; and (iii) creating better prediction/diagnostic tools for real-world complex processes.

My research accomplishments include the following.

(1) *Feature Extraction and Analysis:*

I proposed a novel multiscale recurrence analysis to not only delineate recurrence dynamics in complex systems, but also resolve the computational issues for the large-scale datasets. It was utilized to identify heart failure subjects from the 24-hour heart rate variability (HRV) time series and control the quality of mobile-phone-based electrocardiogram (ECG) signals.

(2) *Modeling and Prediction:*

I proposed the design of stochastic sensor network to allow a subset of sensors at varying locations within the network to transmit dynamic information intermittently, and a new approach

of sparse particle filtering to model spatiotemporal dynamics of big data in the stochastic sensor network. It may be noted that the proposed algorithm is very general and can be potentially applicable for stochastic sensor networks in a variety of disciplines, e.g., environmental sensor network and battlefield surveillance network.

(3) *Monitoring and Control:*

Process monitoring of dynamic transitions in complex systems is more concerned with aperiodic recurrences and heterogeneous types of recurrence variations. However, traditional recurrence analysis treats all recurrence states homogeneously, thereby failing to delineate heterogeneous recurrence patterns. I developed a new approach of heterogeneous recurrence analysis for complex systems informatics, process monitoring and anomaly detection.

(4) *Simulation and Optimization:*

Another research focuses on fractal-based simulation to study spatiotemporal dynamics on fractal surfaces of high-dimensional complex systems, and further optimize spatiotemporal patterns. This proposed algorithm is applied to study the reaction-diffusion modeling on fractal surfaces and real-world 3D heart surfaces.

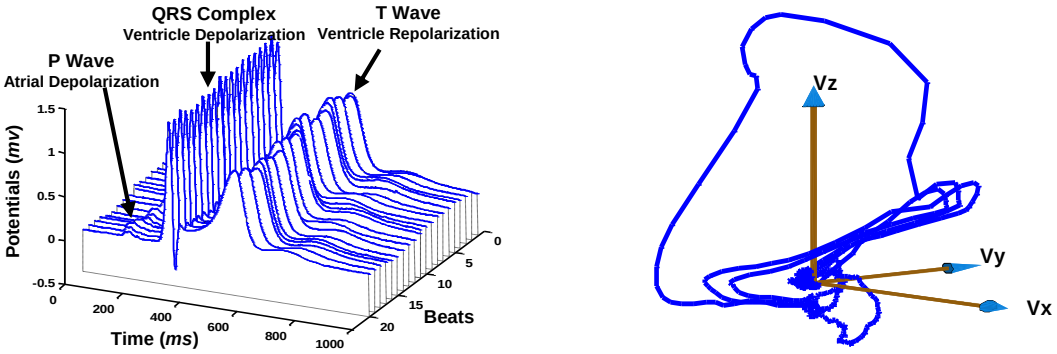
CHAPTER 1: INTRODUCTION AND BACKGROUND

In this chapter, motivations for selecting this research topic are presented. Then the literature review about nonlinear dynamics and recurrence quantification analysis will be introduced. It is followed by the major thrust and scope of research. In the end, the organization of this dissertation is shown.

1.1 Research Motivations

Nonlinear dynamics arise whenever multifarious entities of a system cooperate, compete, or interfere. Effective monitoring and control of nonlinear dynamics will increase system quality and integrity, thereby leading to significant economic and societal impacts. For example, heart disease is responsible for 1 in every 4 deaths in the United States, amounting to an annual loss of $448.5 billion [1]. Realizing a better quality of cardiac operations will reduce healthcare costs and improve the health of our society. Figure 1.1a shows nonlinear waveforms of 1-lead electrocardiogram (ECG) signals when human heart maintains blood circulation through *orchestrated depolarization and repolarization* of cells. It is common to observe the near-periodic patterns but with hidden temporal variations between heart cycles in these physiological signals. Figure 1.1a shows some common characteristics of ECG signals: 1) Within one cycle, the signal waveform at different segments change significantly. The reason is that different segments often correspond to different stages of cardiac operations. 2) Between cycles, the signals are similar to each other but with variations. Near-periodical beatings of human heart provide nourishments to all parts of body and maintain vital living organs.

As complex physiological systems evolve in time, dynamics deal with change. Whether the system settles down to the steady state, undergoes incipient changes, or deviates into more complicated variations, it is dynamics that help analyze system behaviors. Figure 1.1b shows an example of the ECG phase space constructed from multi-lead ECG signals using the Takens' embedding theorem [2]. As multiple sensors are deployed at various locations, distributed sensing provides multi-directional views of nonlinear dynamics in the underlying processes. Traditional linear methodologies focus on the analysis of *time-domain signals*, and attempt to understand a system's behavior by breaking it down into parts and then combining all constituent parts that have been examined separately. This idea underlies such methods as principal component analysis (PCA), Fourier analysis, and factor analysis. These methods encounter difficulties in capturing nonlinear, nonstationary and high-order variations. The breakthrough in nonlinear theory came with Poincaré's geometric thinking of dynamical systems [3, 4], *which focuses on geometric analysis of nonlinear trajectories in the phase space* (see Figure 1.1b).



(a) one-lead ECG in aligned cycles

(b) ECG trajectories in the 3-dimensional phase space

Figure 1.1 Examples of physiological signals. (a) Aligned ECG cycles, (b) ECG trajectories.

Physiological sensing brings the proliferation of measurements of process dynamics (e.g., action potentials, ECG signals, echocardiogram). The challenge now is to harness and exploit nonlinear complexity underlying sensing signals for quality and integrity improvements in cardiac

operations. However, multi-sensing capabilities are not fully utilized to extract information about nonlinear dynamics in the *phase-space domain*. Particularly, nonlinear dynamical systems defy understanding based on the traditional reductionist's approach, in which one attempts to understand a system's behavior by combining all constituent parts that have been analyzed separately. For example, clinicians had thought that drugs that significantly reduce arrhythmic behaviors in isolated cardiac cells would also do so in the heart until the concept was proven wrong by the failure of two large clinical trials [5]. In order to cope with system complexity and increase information visibility, modern healthcare systems are investing in advanced physiological sensing and patient monitoring, thereby giving rise to big data. Realizing the full potential of big data for healthcare intelligence requires fundamentally new methodologies to harness and exploit complexity. However, available nonlinear dynamics techniques are either not concerned with healthcare objectives or fail to effectively analyze big data to extract useful information for improving healthcare services. There is an urgent need to develop analytical methodologies that fully utilizing nonlinear dynamics and chaos principles for advancing healthcare services with exceptional features such as personalization, responsiveness, and superior quality.

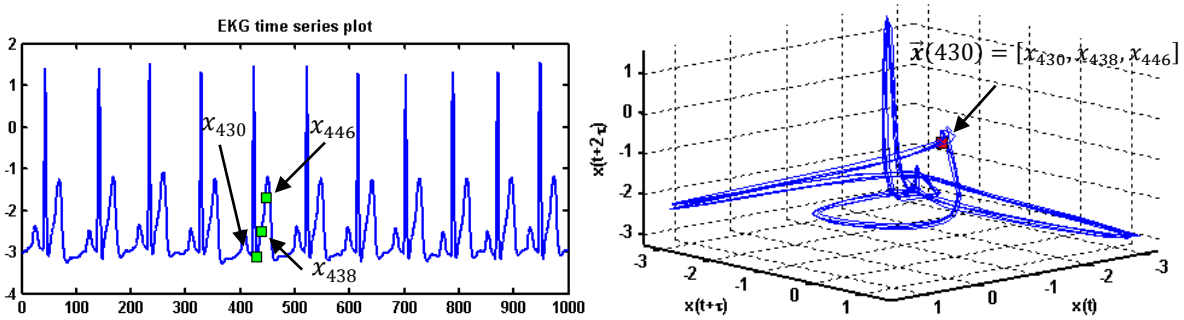
Over the past few decades, the theory of nonlinear dynamics has emerged as a powerful technique in the design of superconducting circuits [6], chatter control in mechanical systems [7], laser stabilization [8], precise fabrication of nanomaterials [9], as well as information security [10]. In addition, several investigations into characterization and modeling of physiological systems, from the cellular level to the system level, have begun to adapt nonlinear dynamics principles.

1.2 Research Background

1.2.1 Nonlinear Dynamics

Nonlinear dynamics theory has emerged as an important methodology for complex systems modeling and analysis. The basic idea is to model the state evolution of underlying processes by a

set of nonlinear differential equations, i.e., $\dot{\boldsymbol{X}} = \frac{d\boldsymbol{X}}{dt} = F(\boldsymbol{X}, \boldsymbol{\theta}), F \in \mathbb{R}^n \rightarrow \mathbb{R}^n$, where $\boldsymbol{X}$ is a multi-dimensional state variable, F is the nonlinear function, and $\boldsymbol{\theta}$ is model parameters. Thus, the solution, i.e., $\boldsymbol{X} = f(\boldsymbol{X}(0), t)$, generates a trajectory representing the flow of state evolution for a given initial condition $\boldsymbol{X}(0)$. When there is a small perturbation in $\boldsymbol{\theta}$ or $\boldsymbol{X}(0)$, the dynamics of a nonlinear process undergo abrupt changes and reveal complex characteristics, including chaos, recurrences, fractals and bifurcations. Notably, linear systems often attribute irregular behaviors of the system to random external inputs, but nonlinear systems can produce very chaotic data with purely deterministic equations and without stochastic inputs.



(a) (b)

Figure 1.2 An example of time delay reconstruction: (a) ECG time series, (b) lag-reconstructed ECG attractor.

Much of the complexity in real-world systems is known to emerge from the underlying nonlinear stochastic dynamics. The exhibited signals from complex systems are often chaotic in nature with irregular behaviors. However, dynamics manifests in the vicinity an attractor **A** (e.g., ECG attractor shown in Figure 1.1b), an invariant set defined in an *m*-dimensional state space. Takens' delay embedding theorem [11] shows that system dynamics can be adequately reconstructed by using the time-delay coordinates of the individual measurements because of the high dynamic coupling existing in physical systems. For the time series $\boldsymbol{X} = \{x_1, x_2, \cdots, x_N\}^T$, state vector $\vec{\boldsymbol{x}}$ (Figure 1.2b) is reconstructed using a delay sequence of $\{x_i\}$ as $\vec{\boldsymbol{x}}(i) =$

$[x_i, x_{i+\tau}, \cdots, x_{i+\tau(m-1)}]$, where m is the embedding dimension and τ is the time delay. Figure 1.2 shows an example of time delay reconstruction of 3-dimensional ECG state attractor from the 1-dimensional ECG time series. The optimal embedding dimension m suffice to unfold the attractor is determined by false nearest neighbor method [12]. In addition, mutual information [13] is used to minimize both linear and nonlinear correlations for the choice of optimal time delay τ.

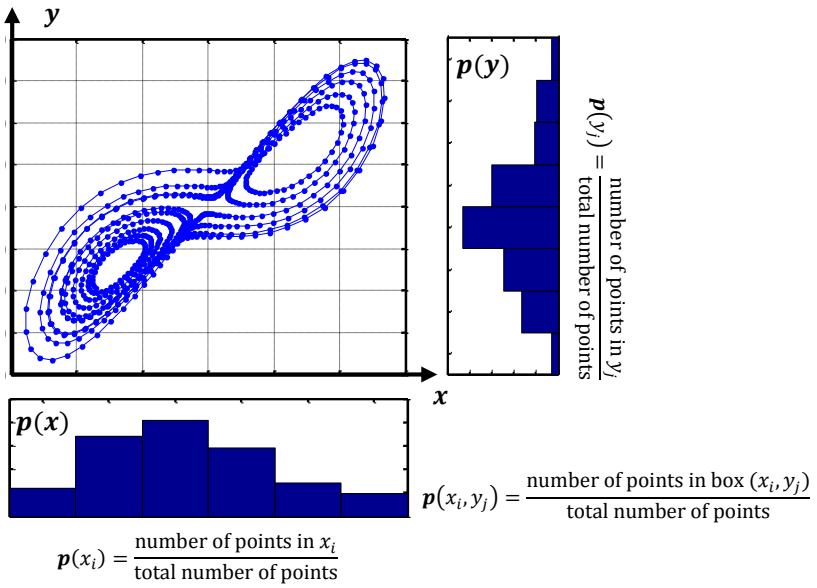


Figure 1.3 An illustration for the computation of mutual information.

If the time delay τ is too small, the attractor will be *restricted to the diagonal* of the reconstructed phase space. However, if the time delay is too large, reconstructed attractor no longer represents the true dynamics. In the literature, there are two traditional approaches for the selection of time delay τ. The first approach is to increase the τ value and then visually inspect that which τ gives the most spread out attractor. The disadvantage of visual inspection is that it only achieves satisfactory results for simple systems. The second approach is autocorrelation function (delay τ)

$$r_\tau = \frac{\sum_{i=1}^{N-\tau}(x_i - \bar{x})(x_{i+\tau} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Optimal τ is required to minimize the linear independence that is the value when the autocorrelation function first passes through 0. Yet, autocorrelation is a second-order quantity evaluating merely linear dependency among data. Notably, mutual information quantifies both linear and nonlinear dependency between two variables x_i and y_j, which is defined as:

$$I(x,y) = \sum_{i,j} p(x_i, y_j)\, log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

where $p(x,y)$ is the joint probabilistic distribution, $p(x)$ and $p(y)$ are marginal probabilities. Figure 1.3 shows the practical implementation to compute the mutual information. In the scatter plot of two variables x and y, the histogram is shown for each variable. Marginal probabilities $p(x_i)$ and $p(y_j)$ are computed as the number of points in x_i and y_j divided by the total number of points in the 2-dimensional space. The joint probability $p(x_i, y_j)$ is computed as the number of points in box (x_i, y_j) divided by the total number of points in the space. Optimal τ is selected to minimize the general dependency between variables that is the first local minimum of Mutual Information function.

The method of false nearest neighbor (FNN) was first proposed by Kennel et al. to determine the minimal embedding dimension m suffice to reconstruct system dynamics [12]. In other words, FNN method is to reconstruct the abstractor in the m-dimensional space that preserves dynamical properties of complex systems in the original phase space. Most importantly, the minimal dimension needs to guarantee the diffeomorphism of reconstruction without any information being lost but without adding unnecessary information. Suppose a m-dimensional attractor is projected to the lower dimensional space (m' dimension and $m' < m$). Due to this projection, the topological structure of the m-dimensional attractor is no longer preserved. Some states are projected into neighborhoods of other states, but they are not true neighbors in the higher dimensional space.

These states are called "false neighbors". An optimal dimension for time-delayed embedding is the smallest dimension that minimizes the number of "false neighbors". However, a larger dimension than the optimum leads to excessive computation when investigating the dynamical properties. "Noise" will populate and dominate the extra dimension of the space where no dynamics is operating. The basic idea of FNN is to measure the distances between a state and its nearest neighbors as this dimension increases. This distance should not change if the states are really nearest neighbors.

For a given time series $\boldsymbol{X} = \{x_1, x_2, \cdots, x_N\}^T$, we calculate the change of distances between neighboring states when the embedding dimension is increased from m to $\mathrm{m} + 1$. If the embedding dimension is high enough, then the fraction of false neighbors is zero, or at least sufficiently small. The state vector in m-dimensional space is

$$\boldsymbol{x}(i) = \left(x_i, x_{i+\tau}, \cdots, x_{i+\tau(m-1)}\right)$$

Let's denote the rth nearest neighbor of $\boldsymbol{x}(i)$ by $\boldsymbol{x}^{(r)}(i)$, then the Euclidean distance between $\boldsymbol{x}(i)$ and its neighbor is

$$R_m^2(\mathrm{i,r}) = \sum_{k=0}^{m-1} \left(x_{i+k\tau} - x_{i+k\tau}^{(r)}\right)^2$$

If the embedding dimension is increased from m to $\mathrm{m} + 1$, the $(\mathrm{m} + 1)$th coordinate is added to each state vector $\boldsymbol{x}(i)$. Therefore, the distance between $\boldsymbol{x}(i)$ and the rth nearest neighbor that we identified in the mth dimension is

$$R_{m+1}^2(\mathrm{i,r}) = R_m^2(\mathrm{i,r}) + \left(x_{i+m\tau} - x_{i+m\tau}^{(r)}\right)^2$$

Then the FNN criterion (i.e., relative change in the distances between neighbors) is

$$\left(\frac{R_{m+1}^2(\mathrm{i,r}) - R_m^2(\mathrm{i,r})}{R_m^2(\mathrm{i,r})}\right)^{1/2} = \frac{\left|x_{i+m\tau} - x_{i+m\tau}^{(r)}\right|}{R_m(i,r)} > R_{tol}$$

where R_{tol} is the threshold. We will now examine the relative change in the distance as a way to see if the states are not really close together when increased to a higher-dimensional space.

1.2.2 Recurrence Quantification Analysis

Recurrence (i.e., approximate repetitions of a certain event) is one of the most common phenomena in natural and engineering systems. For examples, the human heart is near-periodically beating to maintain vital living organs, and manufacturing machines are cyclically forming sheet metals during production. Real-time sensing brings the proliferation of big data (i.e., dynamic, nonlinear, nonstationary, high dimensional) from complex processes. This provides an unprecedented opportunity for data-driven characterization and modeling of nonlinear recurrence behaviors towards system informatics and control. However, most of existing approaches adopt linear methodologies for analyzing dynamic recurrences. Traditional linear methods interpret the regular structure, e.g., dominant frequencies in the signals. They have encountered certain difficulties to capture the nonlinearity, nonstationarity and high-order variations. For example, Fourier analysis does not provide the temporal localization of frequency components, and assume spectral components exist at all times (i.e., stationarity). Instead, system diagnostics and process control are more concerned with aperiodic recurrences and nonlinear recurrence variations.

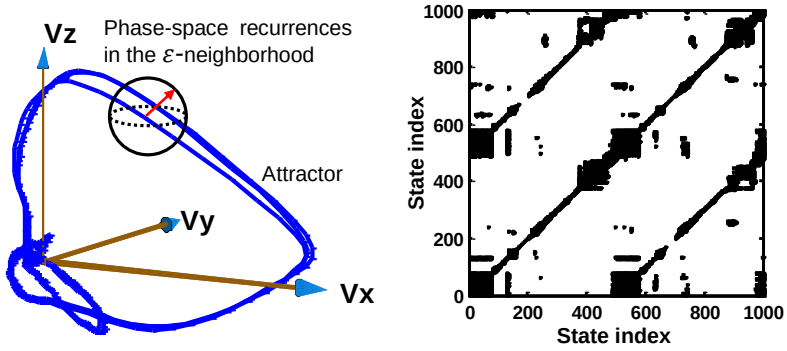


Figure 1.4 (a) An example of ECG trajectories in the 3-dimensional phase space. (b) The recurrence plot characterizes the proximity of two states $\vec{\boldsymbol{x}}(i)$ and $\vec{\boldsymbol{x}}(j)$, i.e., $R(i,j) \coloneqq \Theta(\varepsilon - \|\vec{\boldsymbol{x}}(i) - \vec{\boldsymbol{x}}(j)\|)$, where Θ is the Heaviside function and $||\cdot||$ is a distance measure.

Therefore, nonlinear recurrence methodologies are urgently needed to handle the underlying complexity in the big data. Poincaré recurrence theorem shows that if a dynamical system has the measure preserving transformation, its trajectories eventually reappear in the ε-neighborhood of former states [4]. The methodology of nonlinear recurrence analysis is emerged from the theory of nonlinear dynamics, and characterizes recurrence behaviors in the high-dimensional state space. The recurrence plot was introduced by Eckmann *et al.* in the late 1980's [14] to characterize the proximity of states in the phase space. Mathematically, the recurrence plot is defined as: $\boldsymbol{R}(i,j) = \Theta(\varepsilon - \|\vec{\boldsymbol{x}}(i) - \vec{\boldsymbol{x}}(j)\|)$, where Θ is the Heaviside function, ε the neighborhood size and $||\cdot||$ is a distance measure. For the lag-reconstructed phase space $\vec{\boldsymbol{x}}(i) = \left(x_i, x_{i+\tau}, \cdots, x_{i+\tau(M-1)}\right), i = 1, \cdots, N - \tau(M-1)$, which is lag-reconstructed from a time series, the computation of recurrence plot will be

$$\boldsymbol{R}(i,j) = \Theta\left[\varepsilon - \sqrt{\sum_{m=0}^{M-1}\left(x_{i+m\tau} - x_{j+m\tau}\right)^2}\right]$$

It is worth mentioning that if the neighborhood size ε is too small, there will be few recurrence points in the plot. Thus, we can hardly learn anything about recurrence structures. However, if ε is too large, almost every point is a neighbor of every other point. In the literature, there are several "rules of thumb" for the selection of ε: (1) a small percentage of the maximum diameter of the state space; (2) a fixed scale region in the recurrence rate; (3) fix the number of neighbors for every point; (4) take into account the standard deviation of the observational noise. An optimal choice of ε facilitates the characterization of recurrence structures and dynamical properties of complex systems.

As shown in Figure 1.4, recurrence plot captures topological relationships in the state space as a 2D image. If two states are located close to each other in the m-dimensional state space (e.g.,

3D space in Figure 1.4a), the color code is black (Figure 1.4). If they are located farther apart, the color is white. The structure of a recurrence plot has distinct topology and texture patterns (Figure 1.4b). The ridges locate the nonstationarity and/or the switching between local behaviors. The parallel diagonal lines indicate the near-periodicity of system behaviors.

Recurrence quantification analysis (RQA) measures intriguing structures and patterns in the recurrence plot, including small-structures (e.g., small dots, vertical and diagonal lines), chaos-order transitions, as well as chaos-chaos transitions. A comprehensive review on recurrence quantifiers is reported in Marwan et al., 2007 [15]. Here, we will present several examples of recurrence quantifiers as follows:

(1) Recurrence rate (RR) - a measure of the density of recurrence points in the RP,

$$\mathrm{RR} = \frac{1}{N^2}\sum_{i,j=1}^{N} R(i,j)$$

where N is the number of states in the attractor.

(2) Determinism (DET) - the percentage of recurrence points which form the diagonal lines,

$$\mathrm{DET} = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=1}^{N} lP(l)}$$

where $P(l)$ is the histogram of diagonal line length.

(3) Entropy (ENT) - Shannon information entropy for the probability distribution of the diagonal line lengths $P(l)$

$$ENT = -\sum_{l=l_{min}}^{N} p(l)\ln p(l)$$

(4) Laminarity (LAM) - the percentage of recurrence points which form vertical lines,

$$LAM = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=1}^{N} vP(v)}$$

where $P(v)$ is the histogram of vertical line lengths;

(5) Trapping time (TT) - the average length of vertical structures,

$$\mathrm{TT} = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=v_{min}}^{N} P(v)}$$

Recurrence quantification analysis goes beyond the visual inspection in the RP and provides complexity measures of system dynamics. If we compute the RQA measures in small windows (sub-matrices) along the line of identity (LOI) of the RP, time-dependent behaviors of system dynamics will be quantified. Recurrence quantification analysis have successful applications in various disciplines, e.g., physiology [16-19], biology [20], economy [21], manufacturing [22], geophysics [23], and neuroscience [19, 24].

1.3 Research Objectives

Real-time sensing brings the proliferation of big data that contains rich information of complex systems. It is well known that real-world systems show high levels of nonlinear and nonstationary behaviors in the presence of extraneous noise. This brings significant challenges for human experts to visually inspect the integrity and performance of complex systems from the collected data. My research goal is to develop innovative methodologies for modeling and optimizing complex systems, and create enabling technologies for real-world applications. The study includes feature extraction, process monitoring and fault diagnosis from nonlinear dynamic principles. In cope with the main research objectives, the research tasks are divided into the following:

(1) Design a self-organized approach to effectively estimate and predict the quality of telemedicine ECG recordings.

(2) Investigate space-time dynamics (e.g., reaction, diffusion and propagation) on the fractal geometry and explore the geodesic structure of the high-dimensional surface.

(3) Integrate multivariate statistical control charts with heterogeneous recurrence analysis to simultaneously monitor two or more related quantifiers.

(4) Develop efficient and real-time particle filtering algorithms that forecast the evolving dynamics in the stochastic sensor network.

1.4 Research Contributions

This proposed research will enable and assist in (1) sensor-driven modeling, monitoring and optimization of complex systems; (2) integrating product design with system design of nonlinear dynamic processes; and (3) creating better prediction/diagnostic tools for real-world complex processes. It combines the nonlinear dynamics, predictive modeling, signal processing and computer simulation. Specific research contributions are as follows:

(1) Proposed a novel multiscale recurrence analysis to not only delineate recurrence dynamics in complex systems, but also resolve the computational issues for the large-scale datasets.

(2) Proposed the design of stochastic sensor network to allow a subset of sensors at varying locations within the network to transmit dynamic information intermittently, and a new approach of sparse particle filtering to model spatiotemporal dynamics of big data in the stochastic sensor network.

(3) Developed a new approach of heterogeneous recurrence analysis for complex systems informatics, process monitoring and anomaly detection.

(4) New fractal-based simulation to study spatiotemporal dynamics on fractal surfaces of high-dimensional complex systems, and further optimize spatiotemporal patterns.

1.5 Dissertation Organization

This chapter presents the research motivations and background in this area, as well as research objectives and research contributions. Rest of the dissertation is organized as follows:

(1) Chapter 2: Multiscale recurrence analysis of long-term nonlinear and nonstationary time series. A multiscale recurrence analysis approach is established for the specific heart rate variability time series. Implementation details, validations, results and conclusions are described.

(2) Chapter 3: Self-organized neural network for the quality control of 12-lead ECG signals. This chapter presents my efforts to integrate multiscale recurrence analysis with self-organizing map for controlling the ECG signal quality. The evaluation performances for both training set A and test set B are presented.

(3) Chapter 4: Sparse design for modeling and estimating space-time dynamics in stochastic sensor networks. The new approach of sparse particle filtering is presented to model spatiotemporal dynamics of big data in the stochastic sensor network. Three scenarios of stochastic sensor network (i.e., spatially, temporally, and spatiotemporally dynamic networks) were simulated to demonstrate the effectiveness of sparse particle filtering to support the design of stochastic sensor networks.

(4) Chapter 5: Heterogeneous recurrence quantification of dynamic transitions in continuous nonlinear processes. The heterogeneous recurrence analysis for online monitoring and anomaly detection in nonlinear dynamic processes are presented. Experimental results on simulation studies of auto-regressive and Lorenz model and real-world case studies (UPM sensor and ECG signals) are detailed.

(5) Chapter 6: Simulation modeling and pattern characterization of nonlinear spatiotemporal dynamics on fractal surfaces. Novel methods and tools for the simulation modeling and pattern recognition of spatiotemporal dynamics on fractal surfaces of complex systems are presented in this chapter. Simulation modeling of spatiotemporal dynamics on fractal surfaces and heart surfaces are described.

(6) Chapter 7: Conclusions and future work. This chapter presents research contributions, general conclusions and future work.

CHAPTER 2: MULTISCALE RECURRENCE ANALYSIS OF LONG-TERM NONLINEAR AND NONSTATIONARY TIME SERIES[1]

Recurrence analysis is an effective tool to characterize and quantify the dynamics of complex systems, e.g., laminar, divergent or nonlinear transition behaviors. However, recurrence computation is highly expensive as the size of time series increases. Few, if any, previous approaches have been capable of quantifying the recurrence properties from a long-term time series, while which is often collected in the real-time monitoring of complex systems. This paper presents a novel multiscale framework to explore recurrence dynamics in complex systems and resolve computational issues for a large-scale dataset. As opposed to the traditional single-scale recurrence analysis, we characterize and quantify recurrence dynamics in multiple wavelet scales, which captures not only nonlinear but also nonstationary behaviors in a long-term time series. The proposed multiscale recurrence approach was utilized to identify heart failure subjects from the 24-hour time series of heart rate variability (HRV). It was shown to identify the conditions of congestive heart failure with an average sensitivity of 92.1% and specificity of 94.7%. The proposed multiscale recurrence framework can be potentially extended to other nonlinear dynamic methods that are computationally expensive for large-scale datasets.

2.1 Introduction

With the advancement of microprocessors, miniature sensors, wired and wireless digital networks, contemporary monitoring devices collect enormous amount of datasets exhibited from

[1] This chapter is previously published in [3]. Permission is included in Appendix A.

complex systems. It is well known that real-world systems show high level of nonlinear and nonstationary behaviors in the presence of extraneous noises. This brings significant challenges for human experts to visually inspect the integrity and performance of complex systems from the collected data. It is necessary to develop a method in nonlinear dynamics that can not only explore the system dynamics but also address the computational issues of large-scale datasets.

Conventional frequency-domain analysis and linear statistical approaches tend to have limitations to capture the nonlinear and nonstationary behaviors in the long-term time series. For example, the fast Fourier transformation (FFT) is efficient for the large-scale computation of transform from time domain to frequency domain. Such transform provides additional information different from the original representation of time series, which will not be available otherwise. However, Fourier analysis does not provide the temporal localization of these frequency components, and assumes that spectral components exist at all times (i.e., stationarity). As opposed to this assumption, nonstationarity is usually the case in real-world complex systems. There is a need to identify the time varying spectral components across scales and separate the system nonstationary behaviors in the time-frequency domain. Furthermore, linear statistical methods, e.g., regression models or analysis of variance (ANOVA), have certain difficulties to capture the nonlinearity, nonstationarity and high-order variations. As a result, linear methods lead to less realistic characterization and estimation of time series from nonlinear systems. Nonlinear dynamic methods are necessary to gain a better understanding and interpretation of complex dynamical behaviors, albeit computationally expensive for a long-term time series. Examples of such expensive computations in nonlinear dynamics may include recurrence quantification analysis and false nearest neighbor searching, etc.

Few, if any, previous recurrence-related investigations were aimed at analyzing nonlinear and nonstationary behaviors in the long-term time series. Because recurrence quantification

analysis (RQA) needs to search nearest neighbors in the state pace, the computational complexity is a square increase (i.e., o $\left(\frac{n(n-1)}{2}\right)$) with respect to the size of time series *n*. In this present paper, we developed a novel multiscale recurrence approach to not only explore recurrence dynamics but also resolve the computational issues for the large-scale datasets. Wavelet transform decomposes the nonstationary time series into various frequency bands for effectively separating the system transient, intermittent and steady behaviors. For example, the narrower wavelet function will capture the high-frequency transient behaviors in a fine-grained time resolution, and the wider one will characterize the low-frequency steady behaviors in a better frequency resolution. As such, a nonstationary time series consisting of slowly varying trends with occasional abruptions will be resolved into multiple non-overlapping frequency bands, thereby reducing the complexity of analysis in each scale. In addition, the dyadic subsampling in discrete wavelet transforms leads to the reduction of sample size, which is desirable in the analysis of long-term time series. Because the filtering removes half of the spectral components in the signal and makes half the number of samples redundant, such a subsampling operation does not lose any information. For example, wavelet packet decomposition will decompose a long-term time series of size *n* into 2^k subseries of length ($n/2^k$) at the *k*th level. These shorter subseries will make some expensive nonlinear computations, e.g., recurrence quantification analysis, not only plausible but also more effective under the stationary assumptions in multiple wavelet scales. Hence, multiscale recurrence analysis effectively addresses the issues of nonstationarity and large datasets through time-frequency decomposition. In each wavelet scale, recurrence analysis is utilized to further quantify the dynamics of nonlinear systems, including recurrence rate, determinism, laminarity etc. Such a multiscale framework as we are proposing can be potentially extended and integrated into other nonlinear dynamic approaches.

The structure of paper is organized as follows: Section 2.2 provides the background of wavelet packet decomposition and recurrence methods. Section 2.3 introduces the research methodology of multiscale recurrence analysis, feature selection, and classification. Materials and experimental design are presented in Section 2.4. Section 2.5 shows the implementation results. Section 2.6 discusses and concludes the present study.

2.2 Background

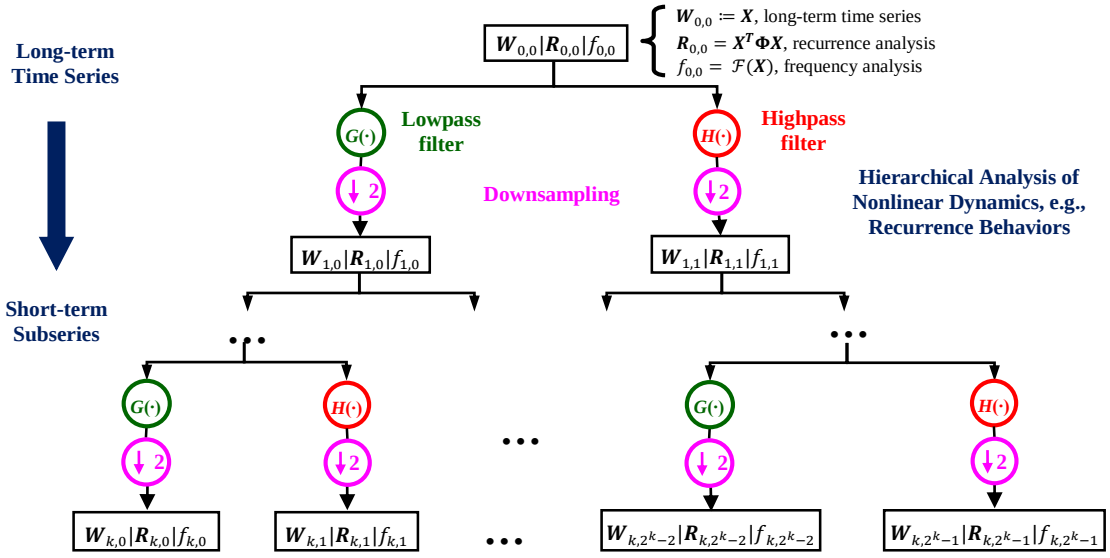


Figure 2.1 Flow diagram illustrating the WPD of a long-term time series $\boldsymbol{X}$, as well as the hierarchical analysis of recurrence and frequency behaviors. The subband coding (i.e., lowpass filter $G(\cdot)$ and highpass filter $H(\cdot)$) and dyadic subsampling processes decompose a long-term time series $\boldsymbol{X}$, defined as $\boldsymbol{W}_{0,0}$, into the k^{th} level subseries that are denoted as $\boldsymbol{W}_{k,n}, n = 0, \cdots, 2^k - 1$, thus facilitate the analysis of recurrence dynamics.

Multiscale analysis is of fundamental importance to solve engineering and physics problems that have important characteristics at multiple scales, e.g., spatial, temporal and/or frequency scales. In meteorology and material sciences, many previous investigations have studied the system physics using information from multiple spatial scales and model such correlations between large and small spatial areas. Examples of successful applications include the prediction of weather evolution [25] and the control of nanomaterial growth [26]. In the field of signal

processing, wavelets are widely used to elucidate the multiscale information of sensor signals in the time-frequency domain [27]. We previously investigated the compactness of biomedical signal representation using continuous wavelet transform (CWT) [28]. While the CWT maintains the same length as the original signals in each scale, the wavelet coefficients are highly redundant across scales. Discrete wavelet transform (DWT) introduces both the wavelet function and scaling function for an efficient pyramid decomposition of space V_j into an approximation space V_{j+1} and a detail space W_{j+1}. The approximation space V_{j+1} is, then, divided iteratively in the next level.

Wavelet packet decomposition (WPD) further extends the DWT by dividing not only the approximation space but also the detail space W_{j+1} in each level. This provides a better resolution in both time and frequency scales. As shown in Figure 2.1, the time series $\boldsymbol{X} = \{x_1, x_2, \cdots, x_N\}^T$, denoted as $\boldsymbol{W}_{0,0}$, is passed through the lowpass filter $G(\cdot)$ and highpass filter $H(\cdot)$ and followed by the dyadic subsampling process in each level. This subband coding is repeated to produce the kth level coefficient sets that are denoted as $\boldsymbol{W}_{k,n}, n = 0, \cdots, 2^k - 1$. The redundancy is removed because each set $\boldsymbol{W}_{k,n}$ is of length $N/2^k$ and the total length in level k is the same as the original time series $\boldsymbol{X}$. The first level of WPD is described in matrix notation as:

$$\mathcal{W}_1 \boldsymbol{X} = \begin{bmatrix} \mathcal{B}_1 \\ \mathcal{A}_1 \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} \boldsymbol{W}_{1,1} \\ \boldsymbol{W}_{1,0} \end{bmatrix} \tag{1}$$

where $\mathcal{W}_1$ is the wavelet matrix with two submatrices $\mathcal{B}_1$ and $\mathcal{A}_1$ of size $\frac{N}{2} \times N$, $\mathcal{B}_1 \mathcal{B}_1^T = \mathcal{A}_1 \mathcal{A}_1^T = \boldsymbol{I}_{\frac{N}{2}}$ and $\mathcal{B}_1 \mathcal{A}_1^T = \mathcal{A}_1 \mathcal{B}_1^T = \boldsymbol{0}_{\frac{N}{2}}$. The time series $\boldsymbol{X}$ can be reconstructed from wavelet decomposition coefficients $\boldsymbol{W}_{1,1}$ and $\boldsymbol{W}_{1,0}$ as

$$\boldsymbol{X} = \begin{bmatrix} \mathcal{B}_1^T & \mathcal{A}_1^T \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_{1,1} \\ \boldsymbol{W}_{1,0} \end{bmatrix} \tag{2}$$

Similarly, the k^{th} level of WPD is calculated with the following two rules: (i) If n in $\boldsymbol{W}_{k-1,n}$ is even, the low-pass filter $G(\cdot)$ is used to obtain $\boldsymbol{W}_{k,2n}$ and the high-pass filter $H(\cdot)$ for $\boldsymbol{W}_{k,2n+1}$ as

in the following equation: $\boldsymbol{W}_{k,2n} = \mathcal{A}_k \boldsymbol{W}_{k-1,n}$ and $\boldsymbol{W}_{k,2n+1} = \mathcal{B}_k \boldsymbol{W}_{k-1,n}$. (ii) If *n* is odd, we use $H(\cdot)$ to obtain $\boldsymbol{W}_{k,2n}$ and $G(\cdot)$ to obtain $\boldsymbol{W}_{k,2n+1}$ as follows: $\boldsymbol{W}_{k,2n} = \mathcal{B}_k \boldsymbol{W}_{k-1,n}$ and $\boldsymbol{W}_{k,2n+1} = \mathcal{A}_k \boldsymbol{W}_{k-1,n}$. The matrices $\mathcal{B}_k$ and $\mathcal{A}_k$ are of size $\frac{N}{2^k} \times \frac{N}{2^{k-1}}$ and also satisfy

$$\mathcal{B}_k \mathcal{B}_k^T = \mathcal{A}_k \mathcal{A}_k^T = \boldsymbol{I}_{\frac{N}{2^k}} \text{ and } \mathcal{B}_k \mathcal{A}_k^T = \mathcal{A}_k \mathcal{B}_k^T = \boldsymbol{0}_{\frac{N}{2^k}} \tag{3}$$

The time series $\boldsymbol{X}$ can be reconstructed from 2^k coefficient sets at level *k*, namely $\boldsymbol{W}_{k,n}, n = 0, \cdots, 2^k - 1$ as

$$\boldsymbol{X} = \mathcal{W}_k^T \begin{bmatrix} \boldsymbol{W}_{k,2^k-1} \\ \vdots \\ \boldsymbol{W}_{k,0} \end{bmatrix} \tag{4}$$

where $\mathcal{W}_k$ is the k^{th} wavelet matrix.

For example, Figure 2.2 shows the process of WPD for a time series $\boldsymbol{X} = \{2,4,3,7,4,6,5,5\}^T$ using Haar wavelet. The lowpass and highpass filters of Haar wavelet are $\{1/\sqrt{2}, 1/\sqrt{2}\}$ and $\{1/\sqrt{2}, -1/\sqrt{2}\}$ respectively. In the first level, the time series $\boldsymbol{X}$ is decomposed into two sub-series as follows: $\boldsymbol{W}_{1,0} = \left\{\frac{(2+4)}{\sqrt{2}}, \frac{(3+7)}{\sqrt{2}}, \frac{(4+6)}{\sqrt{2}}, \frac{(5+5)}{\sqrt{2}}\right\}^T = \left\{3\sqrt{2}, 5\sqrt{2}, 5\sqrt{2}, 5\sqrt{2}\right\}^T$ is the running average of original time series and $\boldsymbol{W}_{1,1} = \left\{\frac{(2-4)}{\sqrt{2}}, \frac{(3-7)}{\sqrt{2}}, \frac{(4-6)}{\sqrt{2}}, \frac{(5-5)}{\sqrt{2}}\right\}^T = \left\{-\sqrt{2}, -2\sqrt{2}, -\sqrt{2}, 0\right\}^T$ is the running difference. Similarly, the k^{th} level decomposition is achieved with two aforementioned rules, i.e., (i) If *n* in $\boldsymbol{W}_{k-1,n}$ is even, the low-pass filter $\{1/\sqrt{2}, 1/\sqrt{2}\}$ is used to obtain $\boldsymbol{W}_{k,2n}$ and the high-pass filter $\{1/\sqrt{2}, -1/\sqrt{2}\}$ for $\boldsymbol{W}_{k,2n+1}$. (ii) If *n* is odd, we use $\{1/\sqrt{2}, -1/\sqrt{2}\}$ to obtain $\boldsymbol{W}_{k,2n}$ and $\{1/\sqrt{2}, 1/\sqrt{2}\}$ to obtain $\boldsymbol{W}_{k,2n+1}$. It may be noted that Haar wavelet is not only orthonormal but also has the unit energy. This allows the perfect reconstruction of original time series from wavelet decomposition coefficients. In addition, signal energy is preserved in the multiple levels of wavelet decomposition (see Figure 2.2). In practice, there exist many types of

wavelets including Haar wavelet, Daubechies wavelet, Meyer wavelet, and Symlet wavelet etc. Entropy is the most widely used criterion to select an optimal wavelet that yields compact signal representations [28]. For more background on the WPD see, for example, [29].

level	**Haar Wavelet Decomposition**								**Energy**
$k=0$	$\boldsymbol{W}_{0,0}=\boldsymbol{X}=\{2,4,3,7,4,6,5,5\}^T$								**180**
$k=1$	$\boldsymbol{W}_{1,0}=\{3\sqrt{2},5\sqrt{2},5\sqrt{2},5\sqrt{2}\}^T$				$\boldsymbol{W}_{1,1}=\{-\sqrt{2},-2\sqrt{2},-\sqrt{2},0\}^T$				**180**
$k=2$	$\boldsymbol{W}_{2,0}=\{8,10\}^T$		$\boldsymbol{W}_{2,1}=\{-2,0\}^T$		$\boldsymbol{W}_{2,2}=\{1,-1\}^T$		$\boldsymbol{W}_{2,3}=\{-3,-1\}^T$		**180**
$k=3$	$\boldsymbol{W}_{3,0}=\{9\sqrt{2}\}^T$	$\boldsymbol{W}_{3,1}=\{-\sqrt{2}\}^T$	$\boldsymbol{W}_{3,2}=\{-\sqrt{2}\}^T$	$\boldsymbol{W}_{3,3}=\{-\sqrt{2}\}^T$	$\boldsymbol{W}_{3,4}=\{0\}^T$	$\boldsymbol{W}_{3,5}=\{\sqrt{2}\}^T$	$\boldsymbol{W}_{3,6}=\{-\sqrt{2}\}^T$	$\boldsymbol{W}_{3,7}=\{-2\sqrt{2}\}^T$	**180**

Figure 2.2 An example of wavelet packet decomposition using Haar wavelet.

However, most existing methods focused on the recurrence dynamics in a single scale. This present work will investigate the recurrence properties of complex systems in the wavelet lenses as well as how they are linked across scales. It is well known that recurrence of states is an important characteristic in the dynamical systems. Recurrence plot is a graphical tool to visualize the recurrence behaviors in the state space $\boldsymbol{x}(i)=\left(x_i,x_{i+\tau},\cdots,x_{i+\tau(M-1)}\right), i=1,\cdots,N-\tau(M-1)$, which is lag-reconstructed from a time series $\boldsymbol{X}=\{x_1,x_2,\cdots,x_N\}^T$. The embedding dimension M is determined by the false nearest neighbor algorithm [12], and the time delay parameter τ is estimated with mutual information method [13]. The unthresholded recurrence plot represents a symmetric distance matrix between the state vectors. It may be noted that the Euclidean distance is used in this present investigation.

$$UR(i,j)=\|\boldsymbol{x}(i)-\boldsymbol{x}(j)\|=\left(\sum_{m=0}^{M-1}\left(x_{i+m\tau}-x_{j+m\tau}\right)^2\right)^{1/2} \tag{5}$$

The thresholded recurrence plot is mathematically expressed by a symmetric matrix $\boldsymbol{R}(i,j)$ characterizing the proximity of the state vectors.

$$R(i,j) = \Theta(\varepsilon - \|\boldsymbol{x}(i) - \boldsymbol{x}(j)\|) \tag{6}$$

where $i,j = 1,\cdots,N-\tau(M-1)$, $\Theta(x)$ is the Heaviside function and the ε defines recurrence of states in the embedded space. Recurrence quantification analysis (RQA) not only characterizes small-structures (e.g., small dots, vertical and diagonal lines) in the recurrence plots but also quantifies the system chaos-order transitions, chaos-chaos transitions and laminar phases etc [30-32]. The recurrence quantifiers include recurrence rate (RR), determinism (DET), maximal length of diagonal structures (LMAX), entropy (ENT), laminarity (LAM) and trapping time (TT). For more background on the recurrence quantification analysis see, for example, [15].

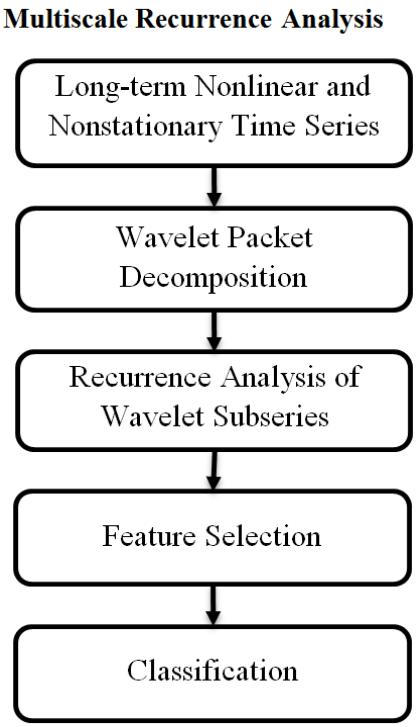


Figure 2.3 Flow chart of the research methodology used.

2.3 Research Methodology

This present paper analyzes recurrence dynamics in multiple wavelet scales and explores the cross-scale links of recurrence characteristics. In addition, we aim to improve the efficiency of recurrence computation for a long-term time series. Few, if any, previous recurrence methods have

demonstrated the possibility of analyzing the large datasets, e.g., the 24-hour heart monitoring time series. With the use of wavelet transform, we delineated the recurrence property not only in various wavelet scales but also the correlation of recurrence behaviors across scales.

As shown in Figure 2.3, the long-term nonlinear and nonstationary time series is firstly decomposed into the short-term subseries in multiple scales using wavelet packets. Then, we quantify the recurrence dynamics in the wavelet subseries instead of the original time series. Thus, the nonstationary behaviors will be resolved into non-overlapped wavelet scales to facilitate the characterization of nonlinear dynamics. Furthermore, the shorter subseries will make the computation of recurrence plots more efficient. Finally, feature selection and classification algorithms will optimally search the space of recurrence features for a parsimonious subset pertinent to abnormal behaviors of the investigated complex systems.

2.3.1 Multiscale Recurrence Analysis

Multiscale recurrence analysis integrates recurrence quantification analysis into the framework of wavelet subband coding. Given a time series $\boldsymbol{X} = \{x_1, x_2, \cdots, x_N\}^T$, the wavelet packet decomposition generates the 2^k sets of coefficients as $\boldsymbol{W}_{k,0}^T, \cdots, \boldsymbol{W}_{k,2^k-1}^T$ in the k^{th} level decomposition. The unthresholded recurrence plot $UR_{0,0}(i,j)|_{i,j=1,\cdots,N}$ of time series $\boldsymbol{X} \coloneqq \boldsymbol{W}_{0,0}$ with no embedding is expressed as

$$UR_{0,0}^2(i,j) = \left|x_i - x_j\right|^2 = \begin{bmatrix} x_i & x_j \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \boldsymbol{X}^T \Phi_{i,j} \boldsymbol{X} \tag{7}$$

where $i, j = 1, \cdots, N$, and $\Phi_{i,j}$ is a $N \times N$ positive semidefinite matrix having 1 in its ii^{th} and jj^{th} elements, -1 in the ij^{th} and ji^{th} elements and 0 otherwise. After the 1st level of WPD, the time series $\boldsymbol{X}$ can be reconstructed by wavelet coefficients $\boldsymbol{W}_{1,1}$ and $\boldsymbol{W}_{1,0}$ as shown in equation (2). Therefore, the original $UR_{0,0}(i,j)|_{i,j=1,\cdots,N}$ can be represented as the function of 1st level wavelet subseries:

$$UR_{0,0}^{2}(i,j) = \boldsymbol{X}^{T}\Phi_{ij}\boldsymbol{X} = \begin{bmatrix}\boldsymbol{W}_{1,1}^{T} & \boldsymbol{W}_{1,0}^{T}\end{bmatrix}\begin{bmatrix}\mathcal{B}_1\\ \mathcal{A}_1\end{bmatrix}\Phi_{i,j}\begin{bmatrix}\mathcal{B}_1^{T} & \mathcal{A}_1^{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{W}_{1,1}\\ \boldsymbol{W}_{1,0}\end{bmatrix} = \begin{bmatrix}\boldsymbol{W}_{1,1}^{T} & \boldsymbol{W}_{1,0}^{T}\end{bmatrix}\Psi_{i,j}^{1}\begin{bmatrix}\boldsymbol{W}_{1,1}\\ \boldsymbol{W}_{1,0}\end{bmatrix} \tag{8}$$

where $\Psi_{i,j}^{1} = \begin{bmatrix}\mathcal{B}_1\\ \mathcal{A}_1\end{bmatrix}\Phi_{i,j}\begin{bmatrix}\mathcal{B}_1^{T} & \mathcal{A}_1^{T}\end{bmatrix}$ is determined based on wavelet matrices $\mathcal{B}_1$ and $\mathcal{A}_1$, i.e., the chosen wavelet and scaling functions. Iteratively, the original unthresholded recurrence plot $UR_{0,0}(i,j)|_{i,j=1,\cdots,N}$ can be represented by the k^{th} level of wavelet subseries.

$$UR_{0,0}^{2}(i,j) = \boldsymbol{X}^{T}\Phi_{i,j}\boldsymbol{X} = \begin{bmatrix}\boldsymbol{W}_{k,2^k-1}^{T} \cdots \boldsymbol{W}_{k,0}^{T}\end{bmatrix}\mathcal{W}_k\Phi_{i,j}\mathcal{W}_k^{T}\begin{bmatrix}\boldsymbol{W}_{k,2^k-1}\\ \vdots\\ \boldsymbol{W}_{k,0}\end{bmatrix}$$

$$= [\boldsymbol{W}_{k,2^k-1}^{T} \cdots \boldsymbol{W}_{k,0}^{T}]\Psi_{i,j}^{k}\begin{bmatrix}\boldsymbol{W}_{k,2^k-1}\\ \vdots\\ \boldsymbol{W}_{k,0}\end{bmatrix} \tag{9}$$

where $\Psi_{i,j}^{k} = \mathcal{W}_k\Phi_{i,j}\mathcal{W}_k^{T}$ is determined by the k^{th} wavelet matrix $\mathcal{W}_k$.

Similarly, in the case of embedded state space $\boldsymbol{x}(i) = \left(x_i, x_{i+\tau}, \cdots, x_{i+\tau(M-1)}\right),\ i = 1, \cdots, N-\tau(M-1)$, the original unthresholded recurrence plot $UR_{0,0}(i,j)|_{i,j=1,\cdots,N}$ is calculated as follows:

$$UR_{0,0}^{2}(i,j) = \sum_{m=0}^{M-1}\left|x_{i+m\tau} - x_{j+m\tau}\right|^2 = \sum_{m=0}^{M-1}\left(\boldsymbol{X}^{T}\Phi_{i+m\tau,j+m\tau}\boldsymbol{X}\right) = \boldsymbol{X}^{T}\left(\sum_{m=0}^{M-1}\Phi_{i+m\tau,j+m\tau}\right)\boldsymbol{X}$$

$$= [\boldsymbol{W}_{k,2^k-1}^{T} \cdots \boldsymbol{W}_{k,0}^{T}]\mathcal{W}_k\left(\sum_{m=0}^{M-1}\Phi_{i+m\tau,j+m\tau}\right)\mathcal{W}_k^{T}\begin{bmatrix}\boldsymbol{W}_{k,2^k-1}\\ \vdots\\ \boldsymbol{W}_{k,0}\end{bmatrix} = [\boldsymbol{W}_{k,2^k-1}^{T} \cdots \boldsymbol{W}_{k,0}^{T}]\Sigma_{i,j}^{k}\begin{bmatrix}\boldsymbol{W}_{k,2^k-1}\\ \vdots\\ \boldsymbol{W}_{k,0}\end{bmatrix} \tag{10}$$

where $\Sigma_{i,j}^{k} = \mathcal{W}_k\left(\sum_{m=0}^{M-1}\Phi_{i+m\tau,j+m\tau}\right)\mathcal{W}_k^{T} = \sum_{m=0}^{M-1}\Psi_{i+m\tau,j+m\tau}^{k}$ is also determined based on the k^{th} wavelet matrix $\mathcal{W}_k$. Hence, the unthresholded recurrence plot $UR_{0,0}(i,j)|_{i,j=1,\cdots,N}$ of original time series $\boldsymbol{X}$ is reconstructed with the k^{th} level of wavelet coefficients $\boldsymbol{W}_{k,n}(n = 0,1,\cdots,2^k-1)$.

Furthermore, wavelet coefficients $\boldsymbol{W}_{k,n}$ are closely related to the k^{th} level of unthresholded recurrence plots $UR_{k,n}(i,j)$, because many previous investigations demonstrated that a time series can be reproduced from its recurrence plots [33-36]. Such an inverse problem can be solved in the following two ways: one is to reconstruct the distance matrices from thresholded recurrence plots and then use multidimensional scaling or matrix eigendecomposition to reproduce the time series [33-35], and the other one is to use the numerical algorithms to reconstruct the time series from the recurrence plots [36]. Both approaches showed the preservation of dynamical properties in the recurrence plots. For example, if we have the thresholded recurrence plot $R_{k,n}(i,j)$ of wavelet coefficients $\boldsymbol{W}_{k,n}$, then the recurrence distance matrix $UR_{k,n}(i,j)$ can be reproduced from the $R_{k,n}(i,j)$ using a weighted graph as follows.

Let $\boldsymbol{z}(i)$ denote a state at the time index $i(1 \le i \le n)$ and define $\mathcal{H}_i \coloneqq \{j|R(i,j) = 1\}$ as the set of time indices to which i is close. A graph is constructed such that two states $\boldsymbol{z}(i)$ and $\boldsymbol{z}(j)$ are linked by an edge only if $R(i,j) = 1$ preserves, and the weight on this edge is

$$\mathcal{w}(i,j) = 1 - \frac{\left|\mathcal{H}_i \cap \mathcal{H}_j\right|}{\left|\mathcal{H}_i \cup \mathcal{H}_j\right|}$$

Then the distance matrix $UR(i,j)$ is reproduced with the shortest distance between two states $\boldsymbol{z}(i)$ and $\boldsymbol{z}(j)$ in the weighted graph $\mathcal{w}(i,j)$.

Moreover, the time series can be reconstructed from the distance matrix $UR(i,j)$ using the method of matrix eigendecomposition. If the origin is added as the first state, i.e., $\boldsymbol{z}(0) = \boldsymbol{0}$, the Gram matrix, i.e., $\mathcal{G}(i,j) \coloneqq \boldsymbol{z}(i) \cdot \boldsymbol{z}(j)(1 \le i,j \le n)$, can be constructed from the distance matrix entirely as follows:

(1) For $i = j$, $\mathcal{G}(i,i) = \boldsymbol{z}(i) \cdot \boldsymbol{z}(i) = \|\boldsymbol{z}(i) - \boldsymbol{z}(0)\|^2 = UR^2(i,0)$.

(2) For $i \neq j$, since

$$UR^2(i,j) = [\boldsymbol{z}(i) - \boldsymbol{z}(j)] \cdot [\boldsymbol{z}(i) - \boldsymbol{z}(j)] = \boldsymbol{z}(i) \cdot \boldsymbol{z}(i) + \boldsymbol{z}(j) \cdot \boldsymbol{z}(j) - 2\boldsymbol{z}(i) \cdot \boldsymbol{z}(j)$$

$$= UR^2(i,0) + UR^2(j,0) - 2G(i,j)$$

Then $\mathcal{G}(i,j)$ is expressed as

$$\mathcal{G}(i,j) = \frac{1}{2}[UR^2(i,0) + UR^2(j,0) - UR^2(i,j)]$$

It may be noted that the Gram matrix is a square matrix and can be decomposed as: $\mathcal{G} = U\Lambda U^T$, where Λ is a diagonal matrix with the eigenvalues of $\mathcal{G}$ and U is a matrix of corresponding eigenvectors. In addition, the positive semidefinite Gram matrix $\mathcal{G}$ is written as:

$$\mathcal{G} = U\sqrt{\Lambda}\sqrt{\Lambda}U^T = U\sqrt{\Lambda}\left(U\sqrt{\Lambda}^T\right)^T = U\sqrt{\Lambda}\left(U\sqrt{\Lambda}\right)^T$$

Based on the theory of isometry (i.e., if two sets of states share the same Gram matrix, then they are isometric) [37], the original time series is isometric to $U\sqrt{\Lambda}$ and there is no loss of dynamical or geometric information in the recurrence plots [34]. Hence, $UR_{0,0}(i,j)$ is closely related to the k^{th} level of wavelet coefficients $\boldsymbol{W}_{k,n}$ and their unthresholded recurrence plots $UR_{k,n}(i,j)$. The dynamical properties in the original recurrence plots are preserved after the wavelet packet decomposition. The recurrence dynamics pertinent to the original time series are further delineated by computing the recurrence plots from the wavelet subseries $\boldsymbol{W}_{k,0} \cdots \boldsymbol{W}_{k,2^k-1}$.

The proposed framework of wavelet recurrence analysis effectively decomposes the long-term time series into short-term subseries, and thus reduces the computational complexity. It may be noted that many previous approaches adjusted the threshold ε for an optimal recurrence plot in the presence of noises [38]. The wavelet approach is more robust to observational noises because it decomposes the system behaviors into different frequency bands. For example, noises will be separated into the high-frequency band and long-term trend will go into the low-frequency band. When there is a mixture of noise, nonlinear and nonstationary behaviors inherent to the original

time series $\boldsymbol{X}$, the proposed wavelet decomposition separates the mixture of information into various wavelet scales. This further reduces the complexity of nonlinear dynamics within each scale. For example, as shown in Figure 2.4, the time series in the original scale has a mixture of nonlinear and nonstationary dynamics, including sinusoids, fractional Brownian motion, Lorenz time series and high frequency noises. The recurrence plot in the original scale shows obscure patterns and can hardly reveal the mixed underlying dynamics. Wavelet packet decomposition separates the highly nonlinear and nonstationary series into four wavelet subseries (see Figure 2.4), whose recurrence plots show distinct patterns of nonlinear and nonstationary behaviors. Therefore, the multiscale recurrence analysis not only improves the computational efficiency of recurrence analysis but also further delineates the system dynamics with the time-frequency decompositions.

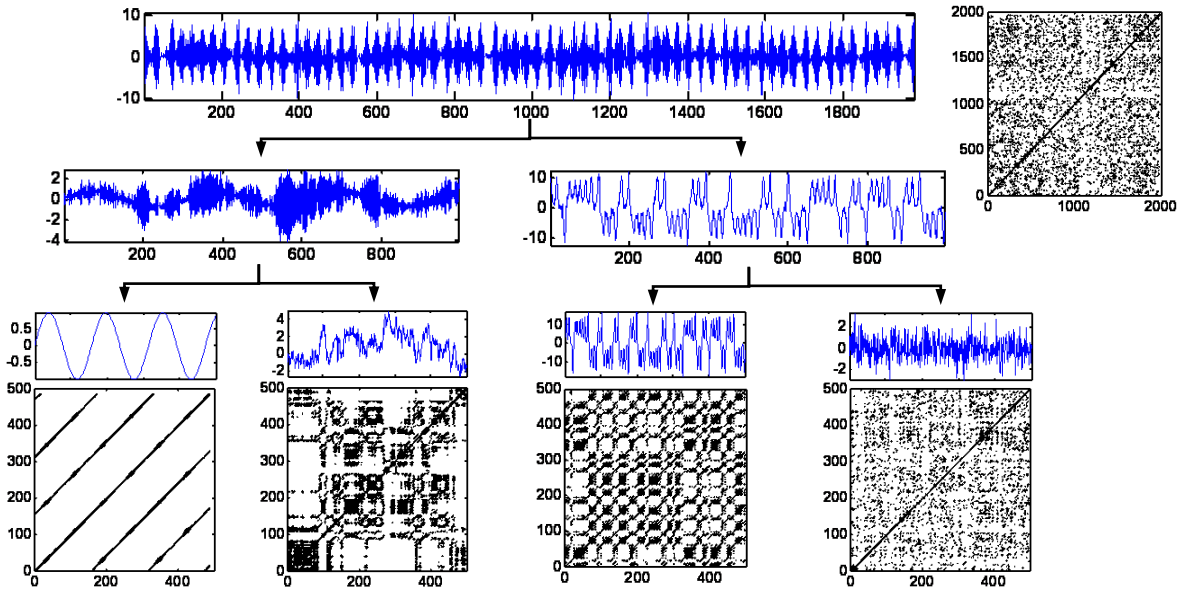


Figure 2.4 Illustration of the multiscale recurrence analysis of a mixed nonlinear and nonstationary time series.

2.3.2 Feature Selection

Because six RQA quantifiers, namely RR, DET, LMAX, ENT, LAM, TT are extracted for each of the wavelet subseries, the k^{th} level wavelet packet decomposition will lead to a high-

dimensional feature space (i.e., 6×2^k number of feature). As a result, this may bring the issues of "curse of dimensionality" for the classification models, e.g., increased model parameters and overfitting problems. Here, a feature selection strategy is proposed to optimally choose a subset of features that are correlated with the process variations, e.g., pathological variations of autonomous nervous control functions. It may be noted that the features selected are dependent on the applications because the behaviors of real-world systems vary in different fields, e.g., plasma physics, machine cutting, and cardiac arrhythmia. A further analysis of chosen feature subset will yield a deeper understanding of the abnormal variations in the underlying processes.

1. Start with empty feature set $\boldsymbol{S}_0 = \{\emptyset\}$
2. Select the additional feature $s^+ = arg\max_{s \notin \boldsymbol{S}_\ell} \Gamma(\boldsymbol{S}_\ell + s)$
   if the objective of discriminant function $\Gamma(.)$ is maximized
3. Update $\boldsymbol{S}_{\ell+1} = \boldsymbol{S}_\ell + s^+;\ \ell = \ell + 1$
4. If (ℓ == desired subset size) end;
   Else Go to 2.

Figure 2.5 Pseudocode of the sequential forward feature selection algorithm.

As shown in Figure 2.5, the sequential forward feature selection is utilized to wrap the discriminant models into the objective function. Starting from an empty feature subset, the additional feature s^+ is selected when it maximizes the objective function $\Gamma(\boldsymbol{S}_\ell + s^+)$ in terms of the predictive accuracy. This process is repeated until it reaches the desired subset size. The step of feature selection cannot only surmount the aforementioned classification complexity and overfitting problems, but also provide faster and more cost-effective models with the optimal feature subset [39].

2.3.3 Classification

Classification models associate the input feature pattern $\boldsymbol{s}$ to one of the $\mathcal{K}$ classes of process conditions, $\mathcal{C}_1, \dots, \mathcal{C}_{\mathcal{K}}$. To construct the classification model, the whole dataset $\mathcal{D}$ is partitioned into the training dataset $\mathcal{D}_1 = \{\langle y(i), \boldsymbol{s}(i)\rangle | i = 1, \cdots, N_1\}$ and testing dataset $\mathcal{D}_2 = \{\langle y(i), \boldsymbol{s}(i)\rangle | i = N_1 + 1, \cdots, N_1 + N_2\}$, where N_1 and N_2 are the size of training and testing datasets, $y(i)$ takes values in the output set $\mathcal{C}_1, \dots, \mathcal{C}_{\mathcal{K}}$, $\boldsymbol{s}(i) = \{s_{i1}, s_{i2}, \dots, s_{i\ell}\}$ is the set of ℓ selected features for the i^{th} record in $\mathcal{D}$.

One instance-based learning K-Nearest-Neighbor (KNN) model and two perceptron-based learning models (i.e., logistic regression (LR) and artificial neural network (ANN)) are considered in this investigation. The KNN rule is an intuitive method that classifies unlabeled examples based on nearest training examples in the feature space. For a given feature point $\boldsymbol{s}$ from the testing dataset $\mathcal{D}_2$, find the k "closest" feature examples $\boldsymbol{s}_{(r)}, r = 1, \cdots, k$ in the training dataset $\mathcal{D}_1$and assign $\boldsymbol{s}$ to the class that appears most frequently within the *k*-subset.

The LR model can be regarded as a single-layer perceptron network when the transfer function is chosen to be a logistic function. Generally, the feature vector $\boldsymbol{s}$ is added an unitary element (i.e., $\boldsymbol{s}' = \{1, s_1, s_2, \dots, s_\ell\}$) in order to use the simple dot product $\boldsymbol{\alpha} \cdot \boldsymbol{s}'$ for a linear combination of vector components instead of the more cumbersome notation $\boldsymbol{\alpha} \cdot \boldsymbol{s} + \alpha_0$. Then the LR model calculates the class membership probability of this feature point $\boldsymbol{s}$ as:

$$p(\boldsymbol{s}) = \frac{e^{\boldsymbol{\alpha} \cdot \boldsymbol{s}'}}{e^{\boldsymbol{\alpha} \cdot \boldsymbol{s}'} + 1} = \frac{1}{1 + e^{-(\boldsymbol{\alpha} \cdot \boldsymbol{s}')}}$$

The parameters $\boldsymbol{\alpha}$ are determined based on the feature sets in the training dataset $\mathcal{D}_1$, usually by maximum-likelihood estimation. It may be noted that the LR model is a common generalized linear model for binomial regression. The hyperplane of feature vector $\boldsymbol{s}$ satisfying the equation $\boldsymbol{\alpha} \cdot \boldsymbol{s}' = \boldsymbol{0}$ (i.e., the class membership probability is 0.5) forms the decision boundary between two classes.

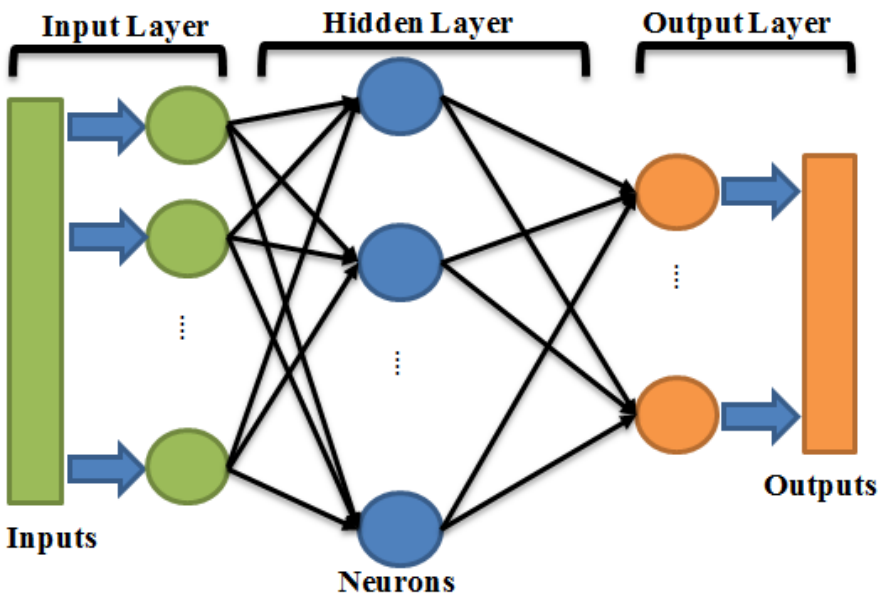


Figure 2.6 The structure diagram of a multilayer feed-forward artificial neural network.

The multilayer perceptron network (ANN) further introduces a number of hidden layers between input and output layers. A feed-forward ANN is shown in Figure 2.6, the network is trained with the training dataset $\mathcal{D}_1$ to determine input-output mapping. The network weights are then optimized for the classifications of the testing dataset $\mathcal{D}_2$. In this present investigation, we used a neural network with two hidden layers. Hyperbolic tangent sigmoid transfer function (tansig) is used in two hidden layers and a linear transfer function (purelin) is used in the output layer. Both hidden layers have 40 neurons and the output layer contains 2 neurons representing the classification targets. The mean squared normalized error (MSNE) is used as the performance metric of the neural network model.

2.4 Materials and Experimental Design

In this present paper, the proposed multiscale recurrence approach is implemented and validated with a real-world case study. The case study is to investigate multiscale recurrence properties of heart rate variability (HRV) between healthy control and heart failure subjects.

2.4.1 Databases

The HRV database contains the 24-hour heart rate time series that are gathered from 54 healthy control (HC) subjects and 29 congestive heart failure (CHF) patients, available in the PhysioNet [40]. Heart failure is caused by a loss of cardiac ability to supply sufficient blood flows to the body. As a result, central nervous system controls the heart rate to compensate heart failure by maintaining blood pressure and perfusion, e.g., increasing the sympathetic activity. However, the neural control of heart rate dynamics reveals highly complex variations [41] that are not only nonlinear and nonstationary but also with long-range correlations, at time scales ranging from seconds to minutes to hours. This is vastly different from acute cardiac events that may be associated with only a segment of ECG signals. Therefore, long-term time series are necessary to delineate the complex long-range dependence behaviors in multiple scales for the identification of heart failures. The 54 recordings of normal sinus rhythm are from the population of 30 men (aged from 28.5 to 76) and 24 women (aged from 58 to 73). The 29 recordings of heart failure are from 8 men, 2 women and 21 unknown genders (aged from 34 to 79). The original ECG recordings for all the subjects were digitized at 128 samples per second.

The heart rate time series is preprocessed to eliminate erroneously large intervals and outliers due to missed beat detections following the same procedure as in [41]. The preprocessing procedures include (a) a moving-window average filter, and (b) increment smoothing. For the 5 consecutive points in a moving window, the central point is removed if it is greater than twice the local mean calculated from the other four points. There is no interpolation in this moving-window average filter. The second step calculates differences between adjacent elements in the time series. If the successive increments has opposite sign with amplitudes > 3×standard deviation of increment series, both increments will be replaced by the interpolated value in between. The new heart rate time series is, then, reconstructed from the post-processed series of increments.

2.4.2 Experimental Design

Computer experiments were designed to evaluate the performances of multiscale recurrence analysis. It may be noted that the cutoff distance ε in recurrence plots was chosen to be 5% of the maximal distance of state space [42]. Six recurrence statistics, namely RR, DET, LMAX, ENT, LAM and TT are exacted to quantify the nonlinear dynamic behaviors in each of the wavelet subseries. Therefore, a total of 6×2^k recurrence features are exacted for the k^{th} level wavelet packet decomposition. In the case study of HRV, k is chosen from 6 to 9 for all subjects to explore the optimal decomposition level that captures the frequency ranges of disease variations. The total length of each recording is pruned to be 76000 data points to keep computational consistency for all subjects. Three classification models, namely KNN, LR and ANN are used to associate the multiscale recurrence features with the underlying disease processes.

2.4.3 Cross Validation and Ensemble Voting Classification

To reduce the bias and overfitting in the performance evaluation of classification models, both *K*-fold cross-validation and random subsampling were utilized in this investigation. As shown in Figure 2.7, *K*-fold cross-validation partitions the total dataset $\mathcal{D}$ into *K* folds, in which *K*-1 folds are used for the training purpose and the rest one fold for testing. After completion of all *K* folds, three performance statistics, i.e., sensitivity, specificity and correct rate, are computed from the testing datasets. The correct rate is the number of subjects (i.e., either disease or control) correctly identified in the testing datasets. Sensitivity measures the proportion of actual positives which are correctly identified as such, and the specificity measures the proportion of negatives which are correctly identified. In other words, sensitivity is the indictor which gives the percentage of disease subjects that are identified as having the condition, and specificity is the percentage of healthy control cases that are identified as not having the condition.

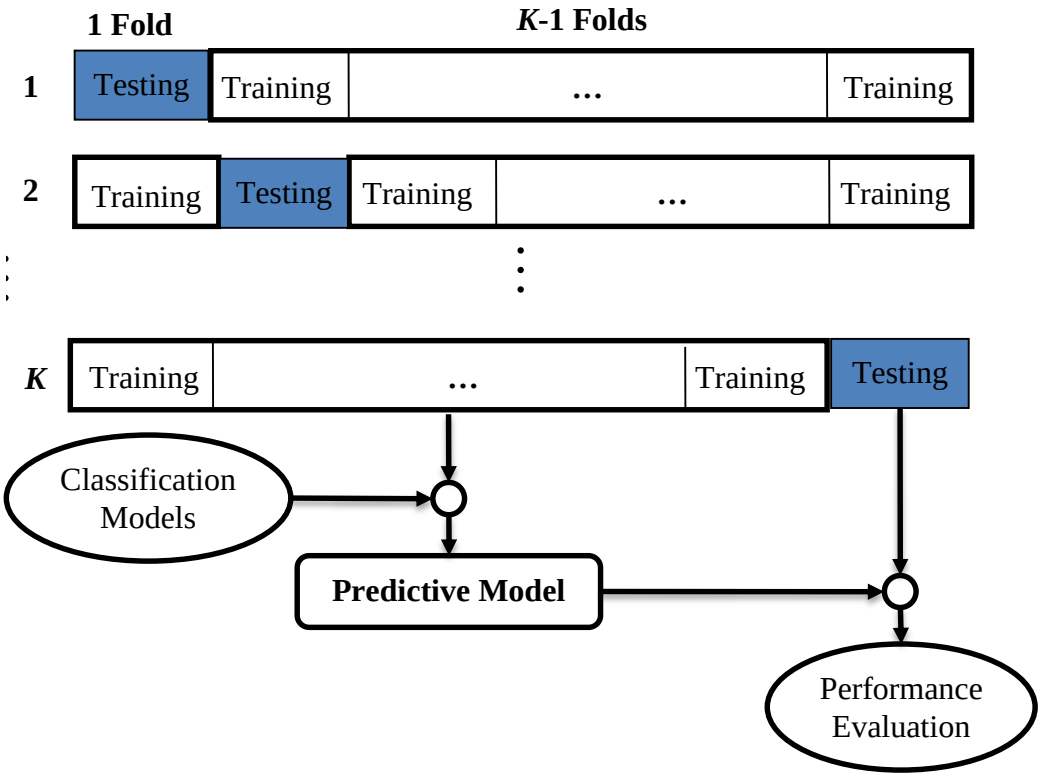


Figure 2.7 The diagram of K-fold cross-validation. The whole dataset is partitioned into *K* equal-sized folds. One is used as the validation subset, and the rest *K*-1 folds as training subsets. The partitions of *K* folds are randomly replicated for 100 times in our experiment with the random subsampling method.

The estimate of true performance is obtained as the average of those *K* error rates on the testing samples [43]. In addition, random subsampling method will randomly create the *K*-fold partitions and replicate such *K*-fold cross-validation experiments for 100 times to obtain the probability distribution of performance statistics. This procedure was repeated from 2-fold to 10-fold for the identification of heart failures. This integration of *K*-fold cross-validation and random subsampling methods prevents the biases from the inequitable selection of training datasets [43].

However, the class sizes in the training datasets are often not equal. The classification models derived, in general, will favor the larger (majority) class. This will affect the sensitivity and specificity for classifying the testing datasets. In the case study, the HRV (54 HCs and 29 CHFs) database has imbalanced class sizes. Thus, an ensemble-based classification approach is

used to statistically bootstrap the minority class with random replacements [44]. For *i*-th *K*-fold cross validation ($i = 1,2,\cdots,K$), *K*-1 folds are denoted as *i*-th training dataset $\mathcal{D}_1^{(i)}$ and the rest one fold as *i*-th testing set $\mathcal{D}_2^{(i)}(=\mathcal{D}-\mathcal{D}_1^{(i)})$. Then, the training dataset $\mathcal{D}_1^{(i)}$ is augmented by bootstrapping additional samples form minority class in $\mathcal{D}_1^{(i)}$ so as to reconstruct the balanced training dataset. The classification models are trained with this augmented and balanced training set. This procedure is replicated *n* times to obtain *n* different bootstrapped training datasets $\mathcal{D}_1^{(i)}$. The final prediction is based on the majority voting from *n* classifiers trained using the *n* different sets of bootstrapped training datasets. This ensemble voting classification will provide more balanced estimates of sensitivity and specificity.

2.5 Experimental Results

2.5.1 Feature Selection

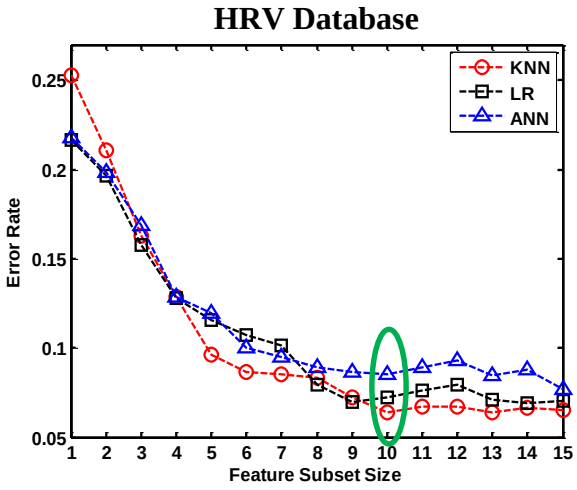


Figure 2.8 Variations of classification error rates versus the size of feature subset for the HRV database. The green circle marks an optimal feature subset size of 10.

As aforementioned, six typical RQA are extracted from the recurrence plot of each wavelet subseries. In total, a large set of recurrence quantifiers are extracted for both cases, i.e., $\sum_{k=6}^{9} 6 \times 2^k = 5760$ features. The method of sequential feature selection is used to optimally

choose a subset of features that are closely correlated with the disease variations. As shown in Figure 2.8, the classification error rates are decreased when the optimal features are sequentially added into the classification models. It may be noted that the error rate oscillates rather than decreases for a larger set of features than 10. Thus, the optimal size of feature subset is selected as 10 for the HRV case study, which also provide a consistent amount of features for building the classification model with reduced complexity.

Furthermore, Table 2.1 shows the statistical evaluation of the individual feature using the unpaired *t*-test and Kolmogorov-Smirnov (KS) test. In the unpaired *t*-test, the smaller *p*-value indicates more evidences to reject the null hypothesis, i.e., the feature has the same distribution between HC and CHF groups. In the KS test, a larger KS statistic shows that this feature has more distinct cumulative distribution functions between the HC and CHF groups. As shown in the Table I, both statistical tests agree on the fact that most of the features selected are significant, because their *p*-values are <0.05 and KS statistic >0.3. However, 1-dimensional statistical test does not account for the feature dependence in the high-dimensional space.

Table 2.1 1-Dimensional unpaired *t*-test and KS test for selected features

Statistic Tests	**Test Statistics**									
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Unpaired *t*-test (*p*-value)	6.2e-04	3.5e-03	0.012	8.9e-04	3.1e-03	0.030	8.5e-03	0.037	0.020	3.5e-03
Two-sample KS test (KS statistic)	0.613	0.372	0.324	0.467	0.425	0.343	0.430	0.343	0.375	0.433

2.5.2 Ensemble Voting Classification

Therefore, three types of classification models were constructed to evaluate the combinatorial effects of multi-dimensional recurrence features. As shown in Figure 2.9, the box plot is used to visualize the distributions of classification performance statistics (i.e., sensitivity,

specificity and correct rate) that are computed from 100 random replications of the *K*-fold cross-validation. The red line in the middle of boxplot represents the median, the blue box shows the lower quartile and upper quartile of performance distributions, and the black dash lines represent the most extreme values within 1.5 times the interquartile range. Outliers are shown as the red crosses in the box plot.

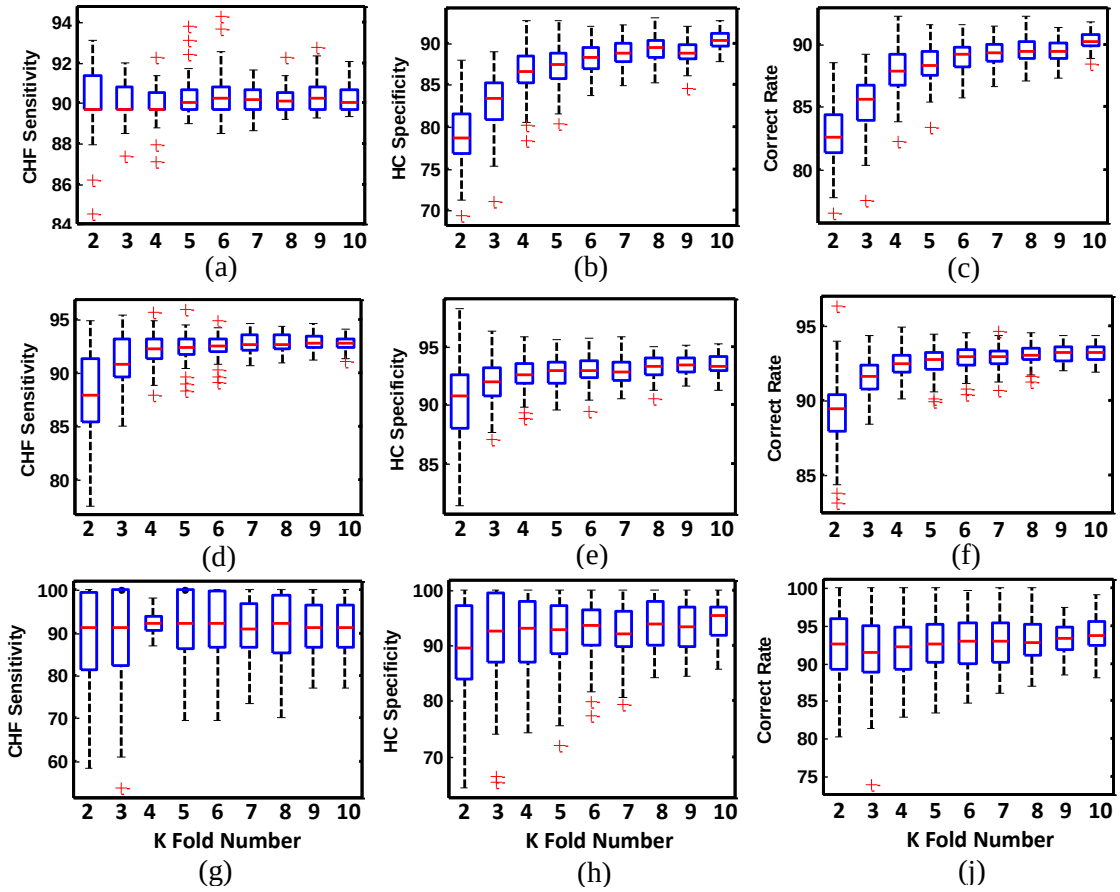


Figure 2.9 Performance results of HRV database for three alternative classification models. (a) sensitivity, (b) specificity and (c) correct rate of KNN model; (d) sensitivity, (e) specificity and (f) correct rate of LR model; (g) sensitivity, (h) specificity and (j) correct rate of ANN model. Note: *The K-fold number is varied from 2-fold to 10-fold, and there are 100 random replicates for each K-fold cross-validation.*

Figure 2.9 (a), (b) and (c) shows the sensitivity, specificity and correct rate respectively for the performances of KNN models. Figure 2.9 (a) demonstrates an average sensitivity of 90% with small deviations (<1.5%). Figure 2.9 (b) shows that the median specificity yields a stable

increasing trend from 79.2% to 90.2% with small deviations (<3.8%). The median correct rate is shown to be 90.2% for the KNN model in the 10-fold cross validation (see Figure 2.9 (c)). The experimental results of KNN models show that the features extracted from the multiscale recurrence analysis are significant between CHF and HC subjects.

Figure 2.9 (d), (e) and (f) present the classification results using LR models. The use of logistic functions are found to further improve the sensitivity to 92.8% and specificity to 93.4%, both with small deviations (<0.8%). The correct rates of LR models are shown to increase from 89.0% to 93.2% with small deviations (<2.2%) when the *K*-fold number is varied from 2-fold to 10-fold (see Figure 2.9 (f)). The ANN models are shown to yield the best classification performances (see Figure 2.9 (g), (h) and (j)). Because the ANN model is a self-adaptive method that can learn the nonlinear and nonstationary relationships in the datasets, the median sensitivity is around 92.1% and the median specificity is around 94.7%. The correct rates of ANN models are around 93.8% for the identification of heart failure subjects in the HRV database.

2.6 Discussion and Conclusions

Few previous investigations have demonstrated the capability to characterize and quantify recurrence dynamics in a long-term time series from the complex systems, e.g., the 24-hour heart rate time series. This paper presented a multiscale recurrence framework to not only explore recurrence dynamics in the complex systems but also resolve the computational issues for the large-scale datasets. Wavelet transform decomposes the nonstationary time series into various frequency bands for effectively separating the system transient, intermittent and steady behaviors. Further, the dyadic subsampling in wavelet transformations enable the computation of recurrence plots from the long-term time series. Also, recurrence analysis becomes more effective under the stationary assumptions in the multiple wavelet scales. Theoretically, we have not only

characterized the recurrence dynamics in multiple wavelet scales but also how the recurrence behaviors are correlated across scales.

In this present paper, the proposed multiscale recurrence approach was utilized to identify heart failure subjects from the 24-hour HRV time series. Computer experiments demonstrated that the proposed approach yields better performances by delineating the nonlinear and nonstationary behaviors in multiple wavelet scales. The ANN classification models were shown to identify the conditions of congestive heart failure with an average sensitivity of 92.1% and specificity of 94.7%.

In summary, the proposed multiscale recurrence approach is shown to effectively capture the nonlinear and nonstationary behaviors in a long-term time series. The wavelet packet decomposition facilitates the exploration of recurrence dynamics in the complex systems. It is anticipated that this new approach can also be applied to other biological and engineering systems with long-term monitoring signals. The multiscale framework we proposed can be potentially "transplanted" to other nonlinear dynamic approaches to characterize system dynamics and improve computational efficiency.

CHAPTER 3: SELF-ORGANIZED NERUAL NETWORK FOR THE QUALITY CONTROL OF 12-LEAD ECG SIGNALS[2]

Telemedicine is very important for the timely delivery of health care to cardiovascular patients, especially those who live in the rural areas of developing countries. However, there are a number of uncertainty factors inherent to the mobile-phone-based recording of electrocardiogram (ECG) signals such as the personnel with minimal training and other extraneous noises. PhysioNet organized a challenge in 2011 to develop efficient algorithms that can assess the ECG signal quality in telemedicine settings. This paper presents our efforts in this challenge to integrate multiscale recurrence analysis with self-organizing map for controlling the ECG signal quality. As opposed to directly evaluate the 12-lead ECG, we utilize an information-preserving transform, i.e., Dower Transform, to derive the 3-lead vectorcardiogram (VCG) from 12-lead ECG in the first place. Secondly, we delineate the nonlinear and nonstationary characteristics underlying the 3-lead VCG signals into multiple time-frequency scales. Further, a self-organizing map is trained, in both supervised and unsupervised ways, to identify the correlations between signal quality and multiscale recurrence features. The efficacy and robustness of this approach are validated using real-world ECG recordings available from PhysioNet. The average performance was demonstrated to be 95.25% for the training dataset and 90.0% for the test dataset with unknown labels.

3.1 Introduction

Recent advancements in telemedicine have shown the trend of using smart phones and other portable computing devices to gather electrocardiogram (ECG) recordings remotely from

[2] This chapter is previously published in [80]. Permission is included in Appendix A.

patients. The ECG signals are recorded on the body surface to track the continuous dynamic details of cardiac functioning. Such valuable real-time information is usually unavailable in static and discrete clinical laboratory tests, for e.g., computer imaging, chest x-ray, and blood enzyme test. However, some ECG recordings, especially those measured by people with minimal training, are distorted by a combination of human errors, noises and artifacts. Examples of these uncertainty factors include patient motion artifacts, poor skin-electrode contact and misplaced electrodes. These issues worsen the quality of ECG recordings and hamper the use of them for the diagnostic purposes. Therefore, it is imperative to perform quality control of the telemedicine ECG signal acquisition, thereby improving the accuracy of cardiovascular diagnostics.

However, noises, artifacts and human errors are inevitable in the data acquisition. ECG noises and artifacts can be within the frequency band of interest and can manifest similar morphologies as ECG itself [45]. Noises often are long-term random fluctuations from either the environment or the devices. Noises contaminate the quality of the signals with undesired disturbances that weaken the useful information in the signals, thereby reducing the signal-to-noise ratio (SNR). Artifacts usually refer to the short-term transient interruptions that are not natural. Examples of ECG noises and artifacts include electrode-skin contact problems, device malfunctions, baseline wander, power-line interference, electrode pop or contact noise and patient-electrode motion artifacts [46, 47]. In addition, human errors are caused by people who recorded the signals. Examples of human errors include misplacement of ECG electrodes, incorrect setting of the ECG system and wrong input of patients' clinical information. For a complete review of noises and artifacts within ECG signals see, for example, [45].

PhysioNet organized a challenge in 2011 to develop efficient algorithms for controlling the ECG signal quality in telemedicine settings [40, 48]. It may be noted that some researchers focused

on the time-domain approaches [49, 50] to extract features pertinent to the signal quality, e.g., amplitudes, spikes, variance, skewness of 12-lead ECG signals. The frequency-domain features are also adopted, including high-frequency (above 50 Hz) and low-frequency (below 17.5 Hz) noises [51, 52]. Most researchers integrated time-domain with frequency-domain approaches for the investigation of ECG quality-related issues [53-63]. In particular, Xia *et al* considered the frequency analysis within short segments of each lead to explore how the frequency features vary over time [55].

It is noticed that unacceptable ECG signals (i.e., poor quality) often show significant nonstationary behaviors, e.g., intermittent abruptions and transient noises due to aforementioned uncertainty factors. The ECG signal was subjected to a variety of tests designed to show evidences that the cardiac signals are nonlinear and nonstationary and the presence of deterministic chaos [64, 65]. Also, cardiac signals have been reported to show nonstationarity and long-range dependence in terms of power law correlations [66]. Due to the high-level nonlinearity and nonstationarity, traditional methods, e.g., time domain analysis and linear statistical approaches, tend to have limitations to extract effective features pertinent to the quality of ECG signals. For examples, time-domain approaches cannot capture the nonlinear steady dynamics, for e.g., Hurst exponent or recurrence properties [17, 43]. Frequency-domain analysis does not provide the temporal localization of these frequency components, and assumes that spectral components exist at all times (i.e., stationarity) [3, 28]. It is generally agreed that ECG artifacts rarely show the linear and stationary behaviors. For example, skin-electrode contact problems cause transient and sharp potential changes or even saturations at full scale deflection (FSD) levels. Therefore, nonlinear dynamic methods are necessary to characterize and quantify the quality-related features hidden in the telemedicine ECG signals. In addition, signal representations in time or frequency domain

alone are not adequate for identifying the time-varying spectral components across scales. Wavelet analysis provides better characterization of nonlinear and nonstationary features of ECG signals in the time-frequency domain [28, 67].

The objective of this present study is to develop classification algorithms that can effectively estimate and predict the quality of telemedicine ECG recordings. Few, if any, previous approaches have integrated wavelets with nonlinear dynamics to explore the nonlinear and nonstationary characteristics of ECG signals, and further investigate how they are correlated with the quality of ECG signals. It may be noted that our previous investigations developed an approach of multiscale recurrence analysis to explore the nonlinear and nonstationary behaviors of long-term time series [3, 43]. Furthermore, we expended this developed multiscale recurrence analysis to quantify quality-related markers underlying the ECG signals. In each wavelet scale, recurrence analysis is utilized to further quantify the nonlinear dynamics of ECG signals, including recurrence rate, determinism, laminarity etc. The new approach of multiscale recurrence analysis captures not only the steady and intermittent characteristics but also transient dynamics in the ECG signals. Such a multiscale framework as we are proposing can be potentially extended and integrated into other nonlinear dynamic approaches.

The structure of paper is organized as follows: after a brief dataset description, the methodology and experimental design are presented in Section 3.2. Section 3.3 shows the implementation results. Section 3.4 discusses and concludes the present study.

3.2 Methodology and Experimental Design

This paper integrates multiscale recurrence analysis with self-organizing map for the quality control of ECG signals. The flow diagram of methodology and experimental design is illustrated in Figure 3.1. As opposed to directly use the 12-lead ECG, we firstly derived the 3-lead

vectorcardiogram (VCG) from 12-lead ECG (i.e., 8 independent leads V1-V6, I and II) with the inverse Dower transformation matrix (see Figure 3.1 (a) and (b)). The VCG signals monitor the space-time cardiac electrical activities along 3 orthogonal planes of the human body, namely, frontal, transverse, and sagittal planes. Secondly, the 3-lead VCG is decomposed into multiple wavelet scales, and recurrence features are extracted from the 3-dimensional phase space constructed by signal components in the same scale (see Figure 3.1 (b) and (c)). Finally, self-organizing map (SOM), in both supervised and unsupervised ways, is utilized to investigate how self-organized patterns of recurrence features are correlated with the quality of ECG signals. The detailed procedures are presented in the following subsections.

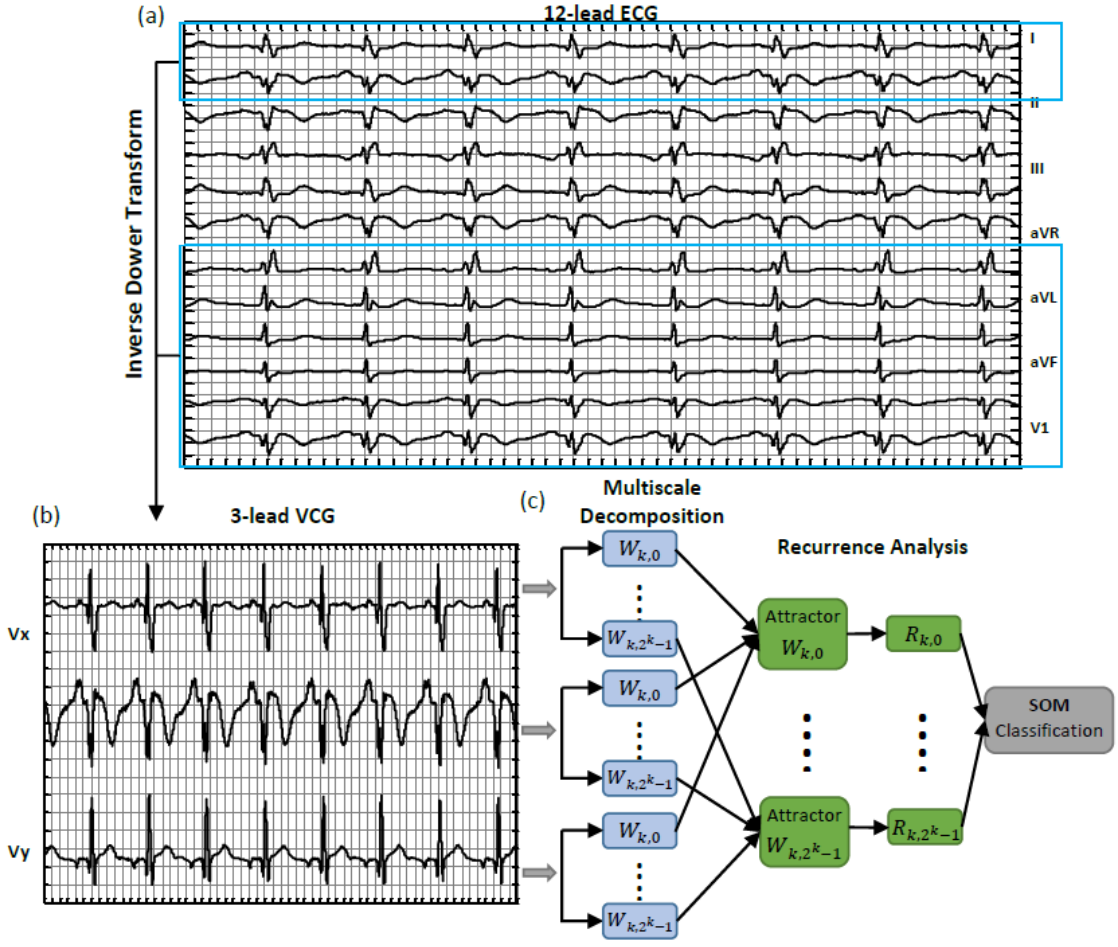


Figure 3.1 Flow diagram of research methodology and experimental design. (a) 12-lead ECG; (b) 3-lead VCG; (c) multiscale recurrence analysis and self-organizing map classification.

3.2.1 Data Description

The real-world ECG dataset is available from the PhysioNet/Computing in Cardiology Challenge 2011 [40, 48]. The challenge is to develop automatic algorithms for the assessment of ECG signal quality, i.e., whether the ECG recording is of adequate quality for diagnostic interpretation. The dataset includes 1500 recordings of standard 12-lead ECG, which is 10-second long with full diagnostic bandwidth (0.05 through 100 Hz). The ECG signals are recorded at the frequency of 500 samples per second, with 16-bit resolution.

The 1500 recordings of 12-lead ECG are examined by a group of human experts. Each human expert assigns a letter and a numerical rating for representing its quality (A-0.95-excellent, B-0.85-good, C-0.75-adequate, D-0.60-poor, or F-0-unacceptable). Then, the numerical ratings are averaged to separate the 1500 recordings into one of the 3 groups, i.e., acceptable, indeterminate and unacceptable. Further, the challenge data are partitioned into two datasets: 1000 recordings are used for training (Set A with known labels of quality groups) and the rest 500 are for testing (Set B with unknown labels). Based on the average rating of human experts, 775 recordings in Set A are acceptable and the remaining 225 recordings are unacceptable. The binary label (acceptable or unacceptable) for each recording of 12-lead ECG in Set A is available for the participants. The quality labels in the test set (Set B) are undisclosed. The scoring is determined by the fraction of correctly classified acceptable and unacceptable ECG recordings in test set B (indetermined recordings are excluded).

3.2.2 Signal Pre-processing

All the ECG recordings are pre-processed with the following procedures before the quality assessment. First, each signal is normalized to the range of [0, 1] and then filtered with a band-pass filter between 1 ~ 120 Hz to remove the baseline wandering and high-frequency noises.

Second, the 1-second segments in the beginning and end of ECG signals are discarded due to the presented transient variations. Third, one ECG recording will be classified as unacceptable if it contains 1 or more straight lines, i.e., ECG electrodes losing connection with the skin during the acquisition. The straight line is defined when there is a constant first order derivative throughout the entire time series. After three pre-processing steps, a total of 769 acceptable and 93 unacceptable recordings are left in training set A, and 429 recordings left in test set B. These remaining data in Set A and Set B will be used in the following steps of transformation, feature extraction, and classification.

3.2.3 Dower Transformation

It is well known that 12-lead ECG records cardiac electrical activities from 12 different directions, and some leads are calculated from others, e.g., lead III can be derived from leads I and II. Dower et al. demonstrated that 12-lead ECG can be linearly transformed to 3-lead vectorcardiogram (VCG) (see Figure 3.1 (a) and (b)), while the transformation accuracy can be improved using separate customized transforms to, for example, age or pathologic conditions [68, 69]. This linear transformation indicates information preservation between 12-lead ECG and 3-lead VCG. Therefore, 3-lead VCG surmounts not only the information loss in one lead ECG but also the redundant information in 12-lead ECG [69].

The inverse Dower transformation matrix derives the 3-lead VCG from 12-lead ECG as follows:

$$VCG = D_{inv} \times ECG$$

where D_{inv} is the inverse dower transformation matrix (3×8) and given by

$$D_{inv} = \begin{bmatrix} -0.172 & -0.074 & 0.122 & 0.231 & 0.239 & 0.194 & 0.156 & -0.010 \\ 0.057 & -0.019 & -0.106 & -0.022 & 0.041 & 0.048 & -0.227 & 0.887 \\ -0.229 & -0.310 & -0.246 & -0.063 & 0.055 & 0.108 & 0.022 & 0.102 \end{bmatrix}$$

ECG is an $8 \times L$ matrix which consists of eight independent leads (V1-V6, I and II) from the 12-lead ECG system, VCG is a $3 \times L$ matrix of XYZ leads, and L is the length of ECG recordings. It may be noted that the linear transformation closely associate 12-lead ECG with 3-lead VCG. In other words, if there are quality-related issues with 12-lead ECG, the transformation will also induce similar issues in 3-lead VCG. However, 3-lead VCG has a lower dimension than 12-lead ECG, thereby more efficient in the computer processing and analysis of quality-related issues.

3.2.4 Multiscale Recurrence Analysis

Cardiac electrical signals, either 12-lead ECG or 3-lead VCG, often show high-level nonlinear and nonstationary behaviors. Conventional time-domain and frequency-domain approaches tend to have limitations to capture the nonlinear and nonstationary characteristics, e.g., steady, intermittent and transient dynamics. Time-domain approaches cannot capture the nonlinear steady dynamics, for e.g., Hurst exponent or recurrence properties [17, 43]. Frequency-domain analysis does not provide the temporal localization of these frequency components, and assumes that spectral components exist at all times (i.e., stationarity) [3, 28]. Wavelet is an effective tool for the time-frequency representation. It not only decomposes signals into various frequency bands but also preserves the time-domain information. It may be noted that the narrower wavelet function capture high-frequency transient behaviors in a fine-grained time resolution, while the wider one will characterize the low-frequency steady behaviors in a better frequency resolution. As such, a nonstationary signal consisting of slowly varying trends with occasionally abruptions is resolved into multiple non-overlapping frequency bands, thereby reducing the complexity to identify quality-related features in each scale.

Wavelet analysis includes continuous wavelet transformation (CWT), discrete wavelet transformation (DWT) and wavelet packet decomposition (WPD). In CWT, sub-signals in wavelet

scales maintain the same length as the original signal, resulting in redundant information. DWT introduces both the wavelet function and scaling function for dividing the original signal into the approximations and details [29, 67]. WPD is similar to DWT, but it further divides not only the approximations but also the details in each wavelet scale. This provides a better resolution in both time and frequency scales. As illustrated in Figure 3.2, it may be noted that the original signal, denoted as $\boldsymbol{W}_{0,0}$, is firstly decomposed into 2 sub-signals. Then, each sub-signal is iteratively decomposed to 2 more wavelet sub-signals, thereby producing 2k sub-signals, i.e., $\boldsymbol{W}_{k,n}, n = 0, \cdots, 2^k - 1$, in the kth level. The redundancy is removed because each set $\boldsymbol{W}_{k,n}$ is of length $L/2^k$ and the total length in level k is the same as the original signal $\boldsymbol{W}_{0,0}$. For more background on WPD see, for example, [29].

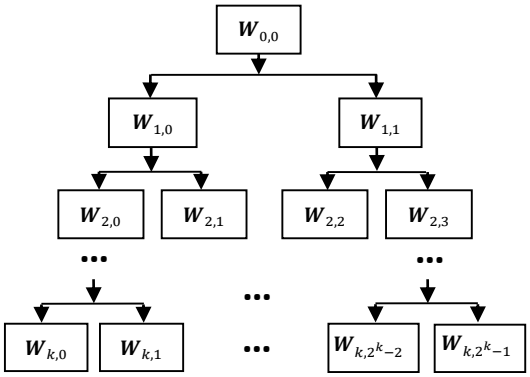


Figure 3.2 Tree structure of wavelet packet decomposition.

Within each wavelet scale, recurrence analysis is utilized to quantify the underlying dynamics of nonlinear systems [15]. We have previously utilized recurrence quantification analysis (RQA) methods to characterize and quantify the cardiovascular dynamics [43, 70]. With the use of wavelets, RQA was shown to quantify the chaos to order and chaos to chaos variations in multiple frequency scales. However, few existing studies have considered the recurrence analysis for the quality control of ECG signals in telemedicine. Recurrence plots (RPs) were introduced by Eckmann et al. to visualize the recurrences of system states in the phase space [14].

As shown in Figure 3.3, the recurrence plot captures topological relationships existing in the transformed 3-lead VCG vector space. Recurrence patterns will be quantified to identify ECG quality markers from a single-scale view as well as the multi-scale perspectives. As shown in Figure 3.3 (c), the unthresholded recurrence plot (UTRP) delineates the distances of every state $\vec{x}(i)$, the state vector realized at time i, to all the others in the phase space, i.e., $UTRP_{i,j} \coloneqq \|\vec{x}(i) - \vec{x}(j)\|$, where $\|\cdot\|$ is the Euclidean norm. While the thresholded recurrence plot (TRP) (see Figure 3.3 (d)), $TRP_{i,j} \coloneqq \mathbf{\Theta}(\varepsilon - \|\vec{x}(i) - \vec{x}(j)\|)$, only uses dark points to represent the recurrence of states when the distance is below a threshold ε, where Θ is the Heaviside function. The state $\vec{\mathrm{x}}(\mathrm{i})$ refers to $\vec{\mathrm{x}}(\mathrm{i}) = [Vx(\mathrm{i}), Vy(\mathrm{i}), Vz(\mathrm{i})]$, which is a vector composed of 3-lead VCG at the time index i. Analogously, the state vector $\vec{\mathrm{x}}(j)$ is at a different time index j. The texture patterns in recurrence plots reveal information of the 3-lead VCG signals (see Figure 3.3 (c) and (d)). For example, the diagonal structures represent the near-periodic patterns and the vertical structures show the nonstationary behaviors.

Furthermore, six RQA features are extracted to statistically quantify the patterns in recurrence plots, including recurrence rate (RR), determinism (DET), maximal length of diagonal structures ($LMAX$), entropy (ENT), laminarity (LAM) and trapping time (TT) [30-32]. Recurrence rate measures the percentage of dark points in the TRP (the pairs of points whose distance is less than the threshold), and it is calculated as $RR \coloneqq \frac{1}{N^2}\sum_{i,j=1}^{N} TRP_{i,j}$, where N is the size of recurrence plots. Recurrence rate characterizes the global near-periodicity of signals. Determinism (DET) represents the proportion of recurrent points forming diagonal line structures and is defined as: $DET \coloneqq \frac{\sum_{l=l_{min}}^{N} lp(l)}{\sum_{l=1}^{N} lp(l)}$, where $p(l)$ is the frequency distribution of diagonal line segments of length l. DET measures the repeating or deterministic patterns in the signals. Linemax ($LMAX$) is the

length of the longest diagonal line segment in the TRP, $LMAX \coloneqq \max(\{l_i\}_{i=1}^{N_l})$, where $N_l = \sum_{l \geq l_{min}} p(l)$ is the total number of diagonal lines. $LMAX$ is related to the inverse of the largest positive lyapunov exponent. The shorter the $LMAX$ is, the more chaotic (less stable) the signal. Entropy (ENT) is the Shannon information entropy of frequency distribution of diagonal line segments, and is calculated as $ENT \coloneqq -\sum_{l=l_{min}}^{N} p(l)\ln[p(l)]$. The aforementioned 4 RQA features are based on diagonal structures to identify chaos-order transitions [71]. However, two vertical-structure features, i.e., LAM and TT, are to identify chaos-chaos transitions [32]. Laminarity (LAM) is analogous to DET except that it measures the vertical line structures. It is given by $LAM \coloneqq \frac{\sum_{v=v_{min}}^{N} vp(v)}{\sum_{v=1}^{N} vp(v)}$, where $p(v)$ is the histogram of vertical lines of length v. Trapping time (TT) provides a measure of how long the system remains in a specific state, and captures the average length of vertical lines as $TT \coloneqq \frac{\sum_{v=v_{min}}^{N} vp(v)}{\sum_{v=v_{min}}^{N} p(v)}$.

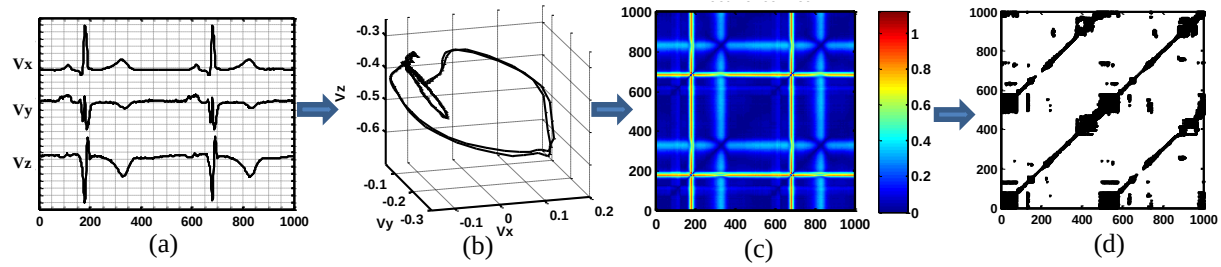


Figure 3.3 Graphical illustration of relationship among (a) 3-lead VCG; (b) VCG trajectory; (c) Unthresholded recurrence plot (UTRP); (d) Thresholded recurrence plot (TRP).

As shown in Figure 3.1 (b) and (c), we not only investigate 3-lead VCG in the original scale but also in multiple wavelet scales. The decomposed wavelet sub-signals in the same scale are used to extract the recurrence patterns. Computer experiments were designed to evaluate the performances of multiscale recurrence features. It may be noted that the threshold ε in recurrence plots was chosen to be 5% of the maximal distance of state space [42]. Six recurrence statistics, namely RR, DET, LMAX, ENT, LAM and TT are exacted to quantify the nonlinear dynamic

behaviors in each of the wavelet subseries. Therefore, a total of 6 ×2k recurrence features will be exacted for the kth level wavelet packet decomposition. In this presented investigation, a length of 4000 input vectors is chosen for all recordings and the level of decomposition k is selected from 2 to 3 to heuristically explore the optimal decomposition level that captures the frequency ranges of quality-related issues.

3.2.5 Self-organizing Map

Self-organizing map (SOM), in both supervised and unsupervised ways, is utilized to investigate how self-organized patterns of recurrence features are correlated with the quality of ECG signals. Self-organizing map automatically organizes data with similar structures close to each other in the output layer of network [72]. The SOM neurons are usually represented on a low-dimensional map (e.g., two-dimensional map), as shown in Figure 3.4. Each neuron i is a vector $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \cdots, w_{id}], i = 1,2, \dots, M$, where M is the number of neurons, and d is the dimension that is the same as the vector of input features $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]$. In each training step, an input feature vector $\boldsymbol{x}$ is presented to SOM neurons. The index c of the best-matching neuron (BMN) is determined by

$$c = arg\min_i \|\boldsymbol{x} - \boldsymbol{w}_i\|$$

The vector of BMN and its neighbors on the hexagonal map are updated by moving towards the input feature vector with the rule of Kohonen update as:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_{ci}(t) \cdot [\boldsymbol{x}(t) - \boldsymbol{w}_i(t)], i = 1,2, \dots, M$$

where $t = 0, 1, 2, \cdots$ is the iteration step of neurons. The neurons in an N-dimensional self-organizing map are initialized so that the initial weights $\boldsymbol{w}_i$ are distributed across the space spanned by the most significant N principal components of the input features. Here $h_{ci}(t)$ is the neighborhood function, which is usually formed as $h_{ci}(t) = h(\|r_c - r_i\|, t)$, where $r_c, r_i \in \Re^2$ are

the locations of neuron c and i. This neighborhood function $h_{ci}(t) \to 0$ when $t \to \infty$, and $h_{ci}(t) \to 0$ when $\|r_c - r_i\|$ increases. Gaussian function is used for $h_{ci}(t)$ in this present investigation as:

$$h_{ci}(t) = \alpha(t) \cdot \exp(-\frac{\|r_c - r_i\|^2}{2[\sigma(t)]^2})$$

where $\alpha(t)$ is the learning-rate factor, and $\sigma(t)$ is the width of the neighborhood kernel function. Both $\alpha(t)$ and $\sigma(t)$ are monotonically decreasing functions of time.

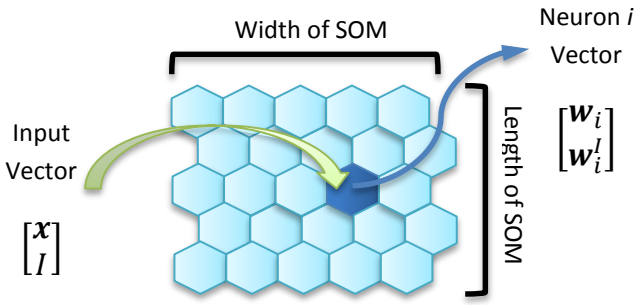


Figure 3.4 Supervised self-organizing map (SOM).

The SOM arranges high-dimensional vectors of input features in a 2-dimensional output space such that similar inputs are mapped onto neighboring SOM regions. Thus, similar patterns of the input data are preserved within the SOM representation space. Supervised SOM integrates the label information as an additional element in the feature vector during the training phase [72, 73]. As shown in Figure 3.4, the input vectors have two parts $\boldsymbol{x}$ and $\boldsymbol{I}$, where $\boldsymbol{x}$ is a sample vector of input features and $\boldsymbol{I}$ indicates the label information of this sample. After the training, each neuron i includes $\boldsymbol{w}_i$ and $\boldsymbol{w}_i^I$, respectively. The label of each neuron is determined by the majority of its associated input vectors' labels. For the testing datasets, the label of best-matching neuron is assigned to an input vector $\tilde{\boldsymbol{x}}$.

The complexity of SOM models is pertinent to the number of neurons used. As the number of neuron increases, it will likely lead to "overfitting" models. Based on the "principles of parsimony", an optimal model is the one with simplest structures but sufficient explanatory power.

Hence, we change the number of neurons and investigate how this will influence model performances. Both the width and length of SOM maps are varied from 3 to 10. An optimal SOM architecture (see Figure 3.4) will be identified for further classification and visualization. To evaluate whether the SOM model is properly trained or not, the metric of quantization error (QE) is computed as

$$QE = \frac{1}{N}\sum_{n=1}^{N}\left\|\boldsymbol{x}_n - \boldsymbol{w}_{c(n)}\right\|$$

where N is the number of recordings, $\boldsymbol{x}_n$ is the input vector of multiscale recurrence features, and $\boldsymbol{w}_{c(n)}$ is the BMN.

The number of neurons in an SOM (i.e. map size) is important to quantify the variations of the input feature vectors. If the map size is too small, it will not capture some important differences underlying the data. However, if the map size is too big, overfitting may occur. In order to choose the optimal map size, a modified Bayesian information criterion (BIC) is designed as

$$BIC = Nln(QE) + pln(N)$$

where p is the map size and N is the number of recordings. The component of $Nln(QE)$ represents the model performance and the component of $pln(N)$ measures the model complexity. In other words, a balance between model performance and complexity will be achieved by minimizing the BIC values.

In this present investigation, the response surface of BIC has higher amplitudes at both small and large map sizes (see Figure 3.5). The blue dash line represents the "valley bottom" of the response surface, where the map sizes are around 20 to 30. An optimal map size 5-by-5 (i.e., the red dot in Figure 3.5) is the minimum of the BIC response surface. Therefore, the SOM with 5-by-5 neurons is used for further classification and visualization.

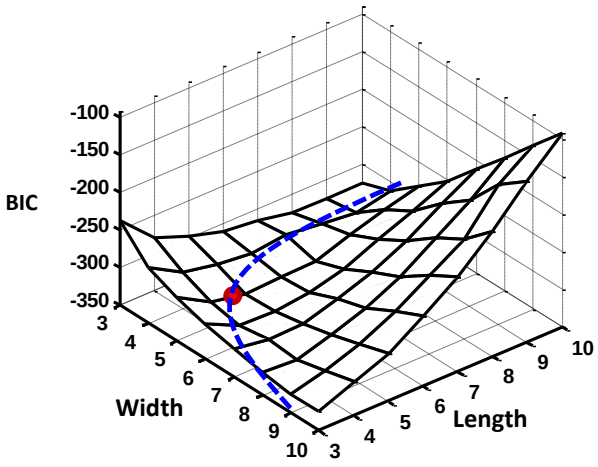


Figure 3.5 BIC response surface for SOM models. The blue dash line represents the "valley bottom" of this surface, and the red dot represents the minimum.

3.2.6 Cross Validation and Ensemble Voting Classification

To reduce the bias and overfitting in classification models, both K-fold cross validation and bootstrapping were utilized in this investigation. K-fold cross validation partitions the total dataset $\mathcal{D}$ into K folds, in which K-1 folds are used for the training purpose and the rest one fold for validation. After completion of all K folds, three performance statistics, i.e., sensitivity, specificity and correct rate, are computed from the validation datasets. The correct rate is the ratio of subjects (i.e., either acceptable or unacceptable) correctly identified in the validation datasets. Sensitivity measures the proportion of actual positives which are correctly identified as such, and the specificity measures the proportion of negatives which are correctly identified. In other words, sensitivity is the indictor which gives the percentage of unacceptable recordings that are identified as unacceptable, and specificity is the percentage of acceptable cases that are identified as acceptable.

However, it may be noted that the training dataset is highly imbalanced, i.e., 769 acceptable and 93 unacceptable recordings. Conventional classification models assume that each class has enough representative cases in the training set. The objective of classification algorithms is to

maximize the overall prediction accuracy. When it comes to a highly imbalanced datasets, classification models tend to favor the majority class and relatively overlook the minority class. Therefore, it will cause a low detection rate of minority case [74], but the prediction of unacceptable recordings is critical in the control of ECG signal quality. Therefore, bootstrapping methods were utilized to reconstruct the balanced datasets. Bootstrapping is a statistical approach that does random sampling with replacement from a dataset. It resamples the training dataset to create a large number of "bootstrapping samples". It is generally agreed that the bootstrapping provides better approximations of the underlying distribution.

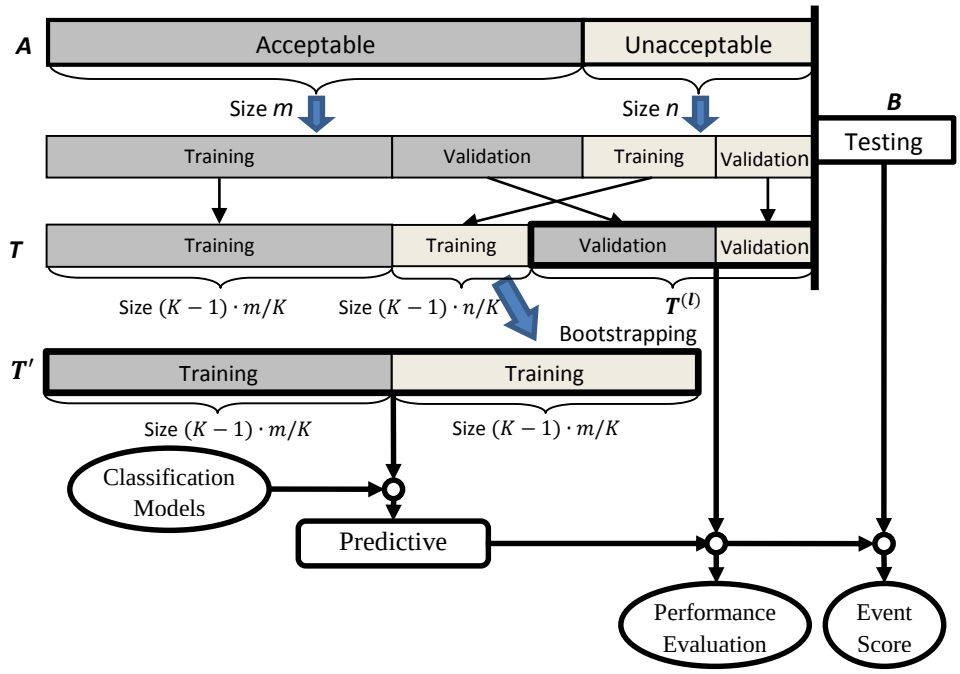


Figure 3.6 The diagram of cross-validation and ensemble voting classification.

As shown in Figure 3.6, dataset A consists of m acceptable and n unacceptable ECG recordings ($m \gg n$). In the first place, the dataset A is randomly partitioned into two subsets: the training subset T and the out-of-bag validation subset $T^{(l)}$. The K-fold cross validation uses $(K-1)$ folds for the training purpose and the rest 1 fold for validation. It may be noted that the partition ratio, i.e., $(K-1)$ training folds vs. 1 validation fold, is the same for acceptable and unacceptable

groups. Furthermore, a balanced training set T is reconstructed with the use of bootstrapping methods. The unacceptable group is enlarged to yield the same size as the acceptable group in the new training set T'. In other words, unacceptable recordings in training set T are resampled with replacement to increase the size from $(K-1)\cdot n/K$ to $(K-1)\cdot m/K$. In addition, the bootstrapping procedure is randomly replicated for R times to avoid biases. For each replicate of the training datasets T', a SOM model will be constructed and trained. A total of R SOM models are yielded for R replicated training datasets. A majority voting mechanism is designed to assign the majority class label to each recording in the validation subset $T^{(l)}$, i.e., a label that appears more than half $(R/2)$ the votes for R SOM models will be assigned to this recording.

3.3 Experimental Results

In the present investigation, multiscale recurrence features were extracted from both Set A and Set B. Supervised SOM models were developed to classify the pre-processed data in Set A (769 acceptable and 93 unacceptable recordings). Self-organizing map is different from other artificial neural networks because it preserves the topological properties of the input space. This makes SOM useful to visualize the high-dimensional data in a low-dimensional map. Because the quality labels for Set B are unknown, it will be used as an independent test dataset (see Figure 3.6). The SOM models will facilitate the comparison of topological properties in multiscale recurrence features extracted from Set A and Set B in the 2-dimensional maps. To optimize the design of SOM models, we have conducted experiments to address the following questions, including (1) How the multiscale recurrence patterns differ from unacceptable to acceptable recordings? (2) How the K-fold cross validation varies the model performance? (3) What are the differences in SOM structures between Set A and Set B? The experimental results are detailed in the following subsections.

3.3.1 Visualization of 12-lead ECG and Recurrence Plots

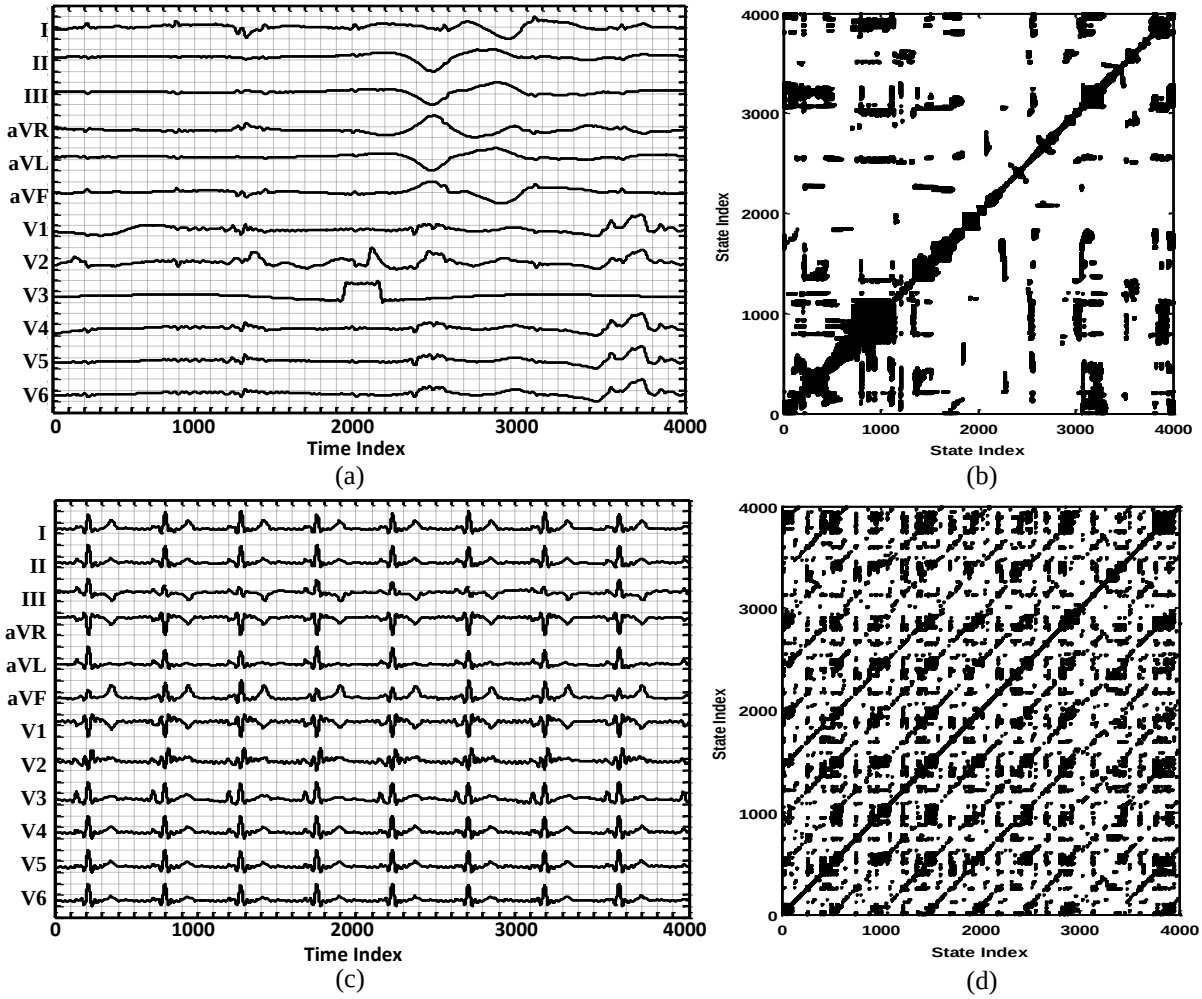
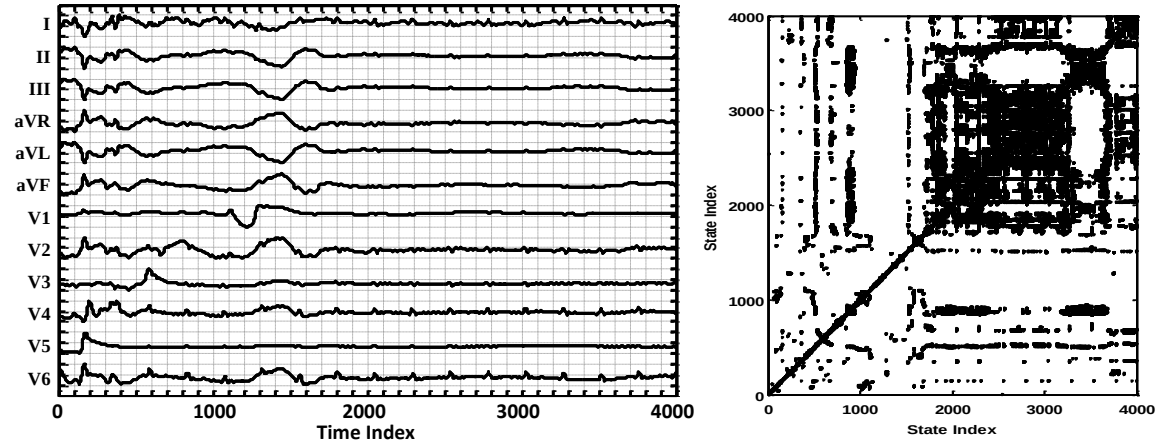


Figure 3.7 Visualization of 12-lead ECG and recurrence plots for the unacceptable recording (#1003574) (a-b) and the acceptable recording (#1005639) (c-d).

Few previous works have investigated how the quality issues of ECG signals relate to the irregular patterns in recurrence plots. In the presence of quality issues, the ECG signals are no longer acceptable for clinical diagnostics because of the distortion of useful information. Figure 3.7 shows the time-domain 12-lead ECG and the pertinent recurrence plots for one unacceptable recording (#1003574) and one acceptable recording (#1005639). As shown in Figure 3.7, recurrence plots have distinct topology and texture patterns for unacceptable and acceptable recordings. The typology includes large-scale patterns that are characterized as homogeneous,

periodic, drift and disrupted. The texture provides a closer inspection of small scale structures, including single dots, diagonal lines and vertical and horizontal lines. It may be noted that the both large-scale and small-scale patterns are difference between unacceptable and acceptable recordings. For the unacceptable recording, Figure 3.7 (a) and (b) show abrupt changes and drifting, non-uniformly distributed TRP patterns. Also, the periodic or quasi-periodic recurrent structures (diagonal lines, checkerboard structures) rarely appear in the TRPs of unacceptable recordings. The longer vertical lines indicate that the state does not change or changes very slowly (i.e., trapped states) for unacceptable recordings. However, for the acceptable recordings, Figure 3.7 (c) and (d) show structured TRP patterns with few abrupt changes and drifts. The discontinuous diagonal (45o) lines indicate the near-periodicity of the system behaviors over given time segments with a period, heart rate, equal to the separations between successive diagonal lines. The vertical lines are shorter than the ones in the TRPs of unacceptable recordings.



(a) (b)

Figure 3.8 Visualization of a recording (# 2428645) labeled as acceptable but classified as unacceptable by the multiscale recurrence analysis. (a) 12-lead ECG. (b) TRP of this recording.

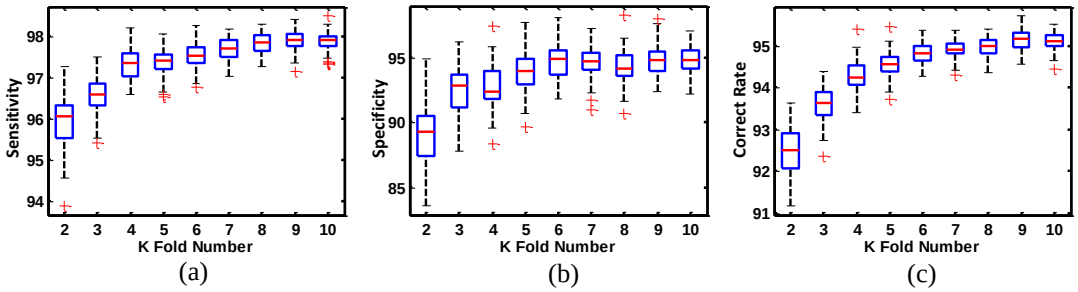
The proposed multiscale recurrence analysis is a data-driven approach that quantifies and extracts the large-scale and small-scale structures from the TRPs. However, there may be a 12-lead ECG labeled as acceptable in the database but having similar recurrence patterns as the

unacceptable recordings. For example, the recording (#2428645) from set A is labeled by human experts as "acceptable", while it has analogous patterns as unacceptable recordings and is classified as "unacceptable" by the multiscale recurrence analysis and self-organizing map models. As shown in Figure 3.8 (a), this ECG recording shows high level of abrupt changes and nonstationary behaviors. The TRP of this recording (see Figure 3.8 (b)) shows non-uniformly distributed point patterns, less quasi-periodic recurrent structures and longer vertical lines that are similar to Figure 3.7 (b). Based on the database descriptions in PhysioNet, all the recordings are labeled as acceptable or unacceptable by human experts. There are also intra/inter rater disagreement on the labels. The classification models, i.e., self-organizing map, automatically organizes recurrence features with similar structures close to each other in the output layer of network.

3.3.2 Ensemble Voting Classification

The SOM classification model with 5×5 neurons is constructed to evaluate the combinatorial effects of multi-dimensional recurrence features. Cross-validation and ensemble voting algorithms are used to estimate the model performance, which is obtained based on the average of those K error rates on the validation samples (see Figure 3.6). In addition, random subsampling method will randomly create the K-fold partitions and replicate such K-fold cross-validation experiments for 100 times to obtain the probability distribution of performance statistics. This integration of K-fold cross-validation and random subsampling methods prevents the biases from the inequitable selection of training datasets [43]. As shown in Figure 3.9, the box plot is used to visualize the distributions of classification performance statistics (i.e., sensitivity, specificity and correct rate) that are computed from 100 random replications of the K-fold cross validation. The red line in the middle of boxplot represents the median, the blue box shows the

lower quartile and upper quartile of performance distributions, and the black dash lines represent the most extreme values within 1.5 times the interquartile range. Outliers are shown as the red crosses in the box plot.



(a) (b) (c)

Figure 3.9 Performance results of supervised SOM with 5-by-5 two-dimensional map. (a) sensitivity, (b) specificity and (c) correct rate. Note: *The K-fold number is varied from 2-fold to 10-fold, and there are 100 random replicates for each K-fold cross-validation.*

As shown in Figure 3.9 (a), (b) and (c), the model performances, i.e., sensitivity (detection rate of unacceptable recordings), specificity (detection rate of acceptable recordings) and correct rate were obtained for the supervised SOM with 5×5 neurons from 2-fold to 10-fold cross validation. Figure 3.9 (a) demonstrates an average sensitivity of 97.85% with small deviations (<0.6%). Figure 3.9 (b) shows that the median specificity yields a stable increasing trend from 89.28% to 94.85% with small deviations (<2.4%). The median correct rate is shown to be 95.18% for the SOM model in the 10-fold cross validation (see Figure 3.9 (c)). The experimental results of supervised SOM models show that multiscale recurrence features are significant between unacceptable and acceptable recordings.

However, it may be noticed that the aforementioned classification performances are only for the training set A after the pre-processing step. Figure 3.10 shows the detailed performances for both the pre-processing step (i.e., straight line detection) and the SOM models. After the pre-processing step, 138 recordings in Set A are separated to be unacceptable in this step (i.e., 132 true unacceptable, 6 false unacceptable). For the rest (i.e., 862 recordings), SOM models yield the

performances of sensitivity (SEN) to be 97.85%, specificity (SPE) to be 94.85% and correct rate (CR) to be 95.18%. After both steps, the overall sensitivity is $\frac{97.85\%*93+132}{225} = 99.11\%$, and the overall specificity is $\frac{94.85\%*769}{775} = 94.12\%$. The total accuracy is $\frac{95.18\%*862+132}{1000} = 95.25\%$.

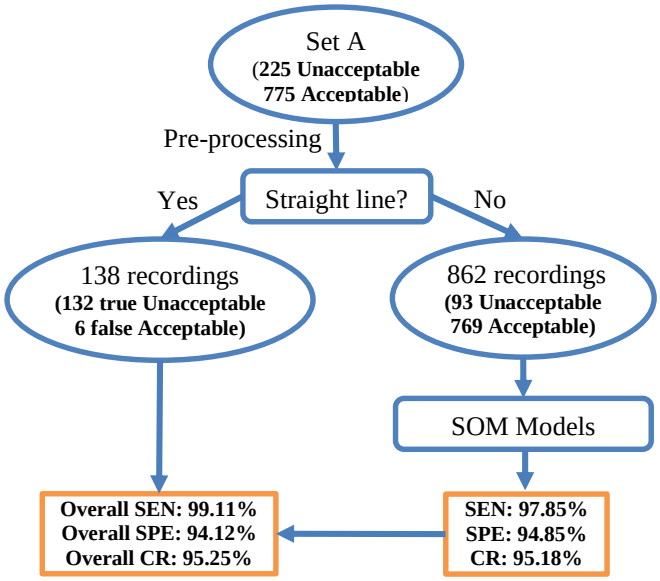


Figure 3.10 Illustration of classification performances for the pre-processing step (i.e., straight line detection) and SOM models.

3.3.3 SOM Pattern Analysis

The SOM represents the high-dimensional data in a low-dimensional map, preserves the topological relationship that cannot be visualized otherwise, and organizes the data automatically according to the inherent structures. The following two subsections will detail the comparison of SOM patterns for Set A and Set B.

3.3.3.1 Training Set A. As shown in Figure 3.11 (a), U-matrix is used to visually show the distances between neurons in the SOM (5-by-5 neuron map) for Set A. The blue hexagons represent neurons, and red lines are connections between two adjacent neurons. The distances between neurons are shown as colored hexagons that embrace red lines. The darker color indicates a larger distance between neurons, and the lighter color is for a smaller distance between neurons.

As shown in Figure 3.11 (a), a cluster of hexagons with light yellow colors appears in the lower-right region, but is bounded by dark red colors. This indicates that neurons are closer to each other in this region, and some input vectors are clustered in this region as being the same class. On the other hand, it may be noted that the upper-left region has darker colors than other regions. This indicates that input vectors in the upper-left region are father apart and belong to a different class. The results shown in Figure 3.11 (a) are further confirmed in the following Figure 3.11 (b) and (c).

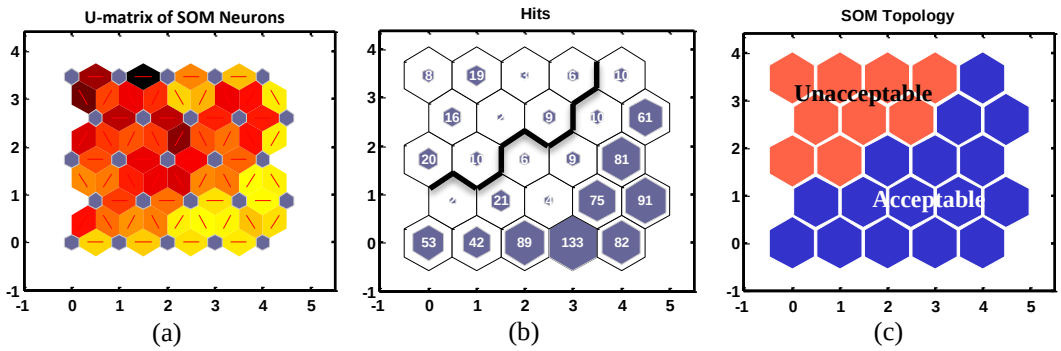


Figure 3.11 Optimized SOM structures for training Set A. (a) U-matrix of SOM neurons; (b) SOM sample hits plot; (c) clustering results on the SOM plane. The decision boundary is marked as a bold line in (b).

As shown in Figure 3.11 (b), each input vector is associated with its best-matching neuron. In other words, the input vector hits on the BMN. Each hexagon in Figure 3.11 (b) represents a neuron, and the number on hexagons is the number of input vectors hitting on this neuron. Because the recordings in the Set A are highly imbalanced, i.e., 769 acceptable and 93 unacceptable recordings, our experimental results show that most of input vectors hit in the lower-right region. Based on the label information of input vectors in Set A, it was found that the lower-right region belongs to the acceptable group. It may be noted that both Figure 3.11 (a) and (b) show clear separation between the acceptable and unacceptable groups.

As shown in Figure 3.11 (c), the SOM map is classified into two groups, i.e., acceptable (blue) and unacceptable (red). The label of each neuron (i.e., hexagon) is determined by the

majority voting of its associated input vectors' labels. For example, if k input vectors hit on this neuron and more than $k/2$ is labeled as acceptable, then this neuron will be marked as acceptable. The SOM patterns are consistent for Figure 3.11 (a), (b) and (c). The upper-left region is classified as unacceptable and a small number of input feature vectors hits in this region. However, a large number of input feature vectors hits in the lower-right region, i.e., acceptable. The decision boundary is marked as a bold black line in Figure 3.11 (b).

SOM weight planes characterize each feature's weight on neurons and how the features are correlated with each other. As shown in Figure 3.12, each weight plane is the graphical characterization of one feature's weight on the neurons. Six weight planes are corresponding to 6 recurrence features (i.e., RR, DET, LMAX, ENT, LAM and TT) extracted from wavelet subseries. Darker colors represent larger weights. If two features show very similar patterns in their weight planes, they are highly correlated. It may be noted from Figure 3.12 that there are significant differences between any pair of six recurrence features. In other words, recurrence features are not correlated with each other and all of them play an important role in the classification of acceptable and unacceptable recordings.

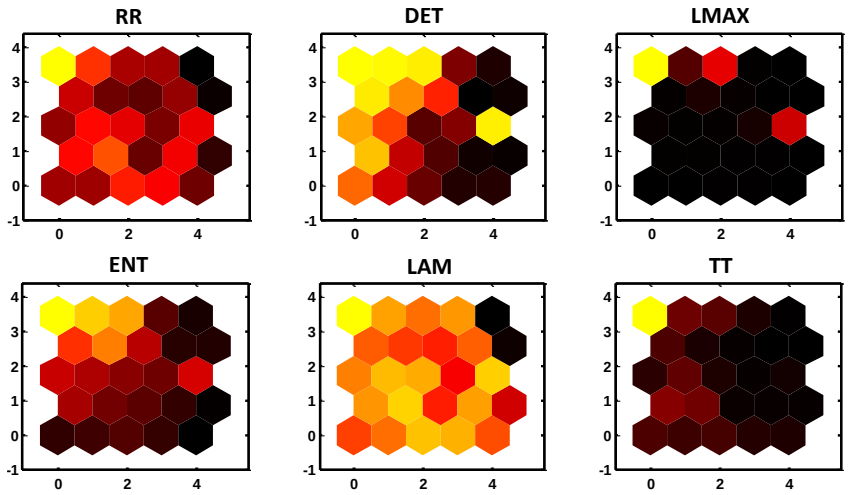


Figure 3.12 Weight planes of SOM for six recurrence features. Each plane corresponds to one of six recurrence features extracted from wavelet subseries.

3.3.3.2 Test Set B. The SOM map, obtained from the supervised training with set A (see Figure 3.11), is used to classify the testing set B. There are a total of 429 recordings left in set B after pre-processing procedure (see section 3.2.2). As shown in Figure 3.13 (a), the sample hits of 429 recordings on the best-matching neuron are clustering in upper-left and bottom regions. This is similar to the distribution of recordings in training set A. The decision boundary (marked as a bold line in Figure 3.13 (a)), obtained from training dataset A, is used to label the recordings of set B as acceptable or unacceptable. The classification results on the SOM map are as shown in Figure 3.13 (b). The accuracy for test set B is 90.0% based on the online evaluation tool of PhysioNet.

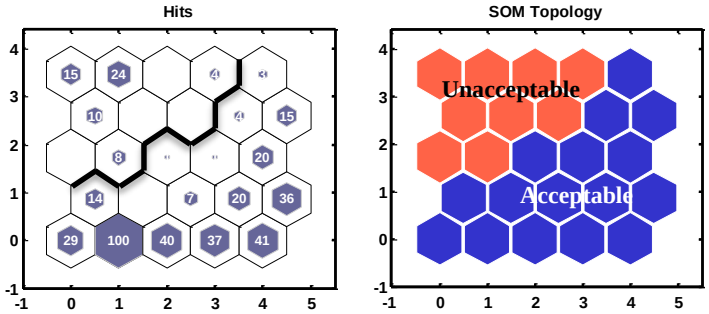


Figure 3.13 Classification results for test Set B using the SOM map obtained from the supervised training with Set A. (a) SOM sample hits plot; (b) clustering results on the SOM plane from training set A. The decision boundary is marked as a bold line in (a).

It may be noticed that that the SOM can also be used as a tool for unsupervised learning. In other words, the SOM is capable of classifying the testing set B even if the training set A is not available. The high-dimensional input feature vectors can be projected onto a low-dimensional SOM map, thereby exploring the inherent data patterns between acceptable and unacceptable groups. The results of unsupervised SOM learning, including U-matrix and sample hits, are shown in Figure 3.14 (a) and (b), respectively. The patterns of U-matrix in Figure 3.14 (a) are similar to the supervised one for training set A (see Figure 3.11 (a)). Light colors cluster in the lower-right region and dark colors appear in the upper-left region. Figure 3.14 (b) shows that sample hits are

more evenly distributed in the upper-left and lower-right neurons than supervised learning in Figure 3.11 (b). However, Figure 3.14 (a) and (b) also show that there are significant differences between acceptable and unacceptable groups. The unsupervised classification of set B is done through the clustering techniques, i.e., fuzzy c-means clustering. The clustering techniques separate two groups based on the inherent feature structures (see Figure 3.14 (a)) organized in the 2-dimensional map, and the decision boundary is obtained as a green dash line in Figure 3.14 (a) and (b) without using the prior label information. The unsupervised SOM learning yields a final accuracy of 89.2% for the test set B without using any prior information from the training set A.

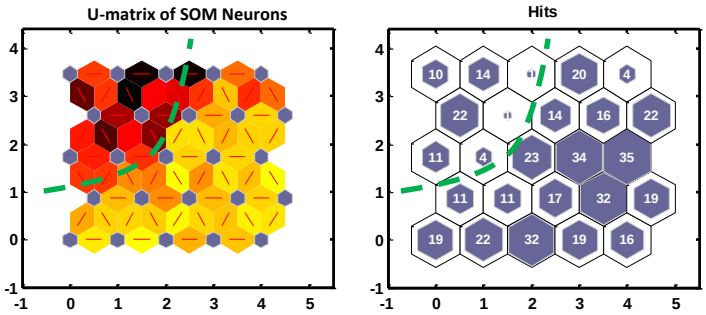


Figure 3.14 Classification results for test Set B using the unsupervised SOM learning. (a) U-matrix of SOM neurons; (b) SOM sample hits plot. The decision boundary is marked as a green dash line in both (a) and (b).

3.4 Discussion and Conclusions

In the telemedicine ECG signal acquisition, recordings are often affected by various uncertainty factors, e.g., artifacts, noises and human errors. These uncertainty factors contaminate the quality of ECG recordings, thereby influencing the quality of cardiovascular diagnostics. It is imperative to develop an effective approach to control the quality of telemedicine ECG signal acquisition.

In this present investigation, as opposed to directly evaluate the 12-lead ECG, we firstly use the inverse Dower transform to derive 3-lead VCG from 12-lead ECG. It may be noted there

is redundant information within the 12-lead ECG, e.g., some leads (Lead III, avR, avL, avF) are calculated from others. Also, the property of linear transformation makes sure that 12-lead ECG is closely associated with 3-lead VCG and the quality-related issues with 12-lead ECG are preserved in the 3-lead VCG. In other words, if there are quality-related issues with 12-lead ECG, the transformation will also induce similar issues in 3-lead VCG. However, 3-lead VCG has a lower dimension than 12-lead ECG, thereby more efficient in the computer processing and analysis of quality-related issues.

Secondly, multiscale recurrence analysis is developed to characterize and quantify nonlinear and nonstationary behaviors of 3-lead VCG within multiple wavelet scales. It is noticed that unacceptable ECG signals (i.e., poor quality) often show significant nonlinear and nonstationary behaviors, e.g., intermittent abruptions and transient noises due to the uncertainty factors. Signal representations in time or frequency domain alone are not adequate for identifying the time-varying spectral components across scales. Wavelet analysis provides better characterization of nonlinear and nonstationary features of ECG signals in the time-frequency domain. Therefore, recurrence methods are utilized to extract the quality-related nonlinear features hidden in the telemedicine ECG signals in multiple wavelet scales.

Finally, self-organizing map is utilized to investigate how the patterns of multiscale recurrence features are correlated with the quality of ECG signals (i.e., acceptable and unacceptable) with both supervised and unsupervised leaning. The SOM represents the high-dimensional data in a low-dimensional map, preserves the topological relationship that cannot be visualized otherwise, and organizes the data automatically according to the inherent structures. Since the dataset A is highly imbalanced between two groups, bootstrapping mechanism is used to reconstruct a balanced training set for SOM classification models. Three performance metrics, i.e., sensitivity, specificity and correct rate are measured for training set A.

The experimental results show that multiscale recurrence features capture nonlinear and nonstationary characteristics in 3-lead VCG signals. In addition, these features are shown to have significant differences between acceptable and unacceptable groups. The average performance was demonstrated to be 95.18% after pre-processing for the training dataset A, and the supervised SOM learning yields an accuracy of 90.0% for test set B based on the online evaluation tool of PhysioNet. This score is 3.2% less than the winning entry of event 1 (closed source) and 1.4% less than the winning entry of event 2 (open source). It may also be noted that the unsupervised SOM learning yields a final accuracy of 89.2% for the test set B without using any prior information from the training set A. This paper presents novel nonlinear dynamic algorithms that have great potentials for the self-organized control of ECG signal quality in telemedicine settings.

CHAPTER 4: SPARSE DESIGN FOR MODELING AND ESTIMATING SPACE-TIME DYNAMICS IN STOCHASTIC SENSOR NETWORKS[3]

Wireless sensor network has emerged as a key technology for monitoring space-time dynamics of complex systems, e.g., environmental sensor network, battlefield surveillance network, and body area sensor network. However, sensor failures are not uncommon in traditional sensing systems. As such, we propose the design of stochastic sensor networks to allow a subset of sensors at varying locations within the network to transmit dynamic information intermittently. Realizing the full potential of stochastic sensor network hinges on the development of novel information-processing algorithms to support the design and exploit the uncertain information for decision making. This paper presents a new approach of sparse particle filtering to model spatiotemporal dynamics of big data in the stochastic sensor network. Notably, we developed a sparse kernel-weighted regression model to achieve a parsimonious representation of spatial patterns. Further, the parameters of spatial model are transformed into a reduced-dimension space, and thereby sequentially updated with the recursive Bayesian estimation when new sensor observations are available over time. Therefore, spatial and temporal processes closely interact with each other. Experimental results on real-world data and different scenarios of stochastic sensor networks (i.e., spatially, temporally, and spatiotemporally dynamic networks) demonstrated the effectiveness of sparse particle filtering to support the stochastic design and harness the uncertain information for modeling space-time dynamics of complex systems.

[3] This chapter is previously published in [124]. Permission is included in Appendix A.

4.1 Introduction

Wireless sensor network has emerged as a key technology to monitor nonlinear stochastic dynamics of complex systems. Recent advancements in wireless communication and electronics have improved the design and development of wireless sensors that are miniature, low-cost, low-power and multi-functional. These inexpensive sensors can be easily networked through wireless links, deployed in large numbers and distributed throughout complex physical systems [75]. Distributed sensing provides an unprecedented opportunity to monitor space-time dynamics of complex systems and to improve the quality and integrity of operation and services.

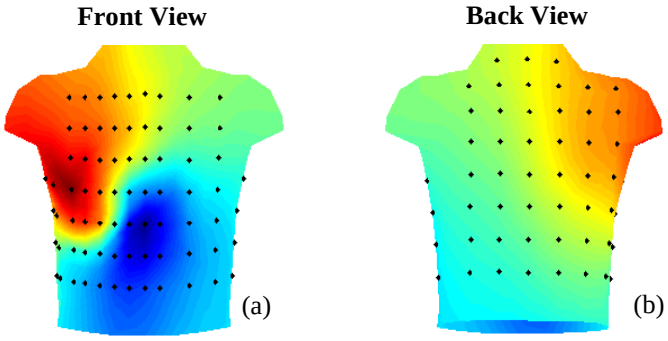


Figure 4.1 Body area ECG sensor network. (a) Front view. (b) Back view. (**Note:** black dots represent sensor locations.)

Further, distributed sensing is widely used to improve the quality of life in healthcare systems. Wireless sensor network shows a significant capability to improve the healthcare services and realize the smart health of chronically ill and elderly. Body area sensor network [76] is the ad hoc network of sensors (e.g., RFID tags, electrocardiogram (ECG) sensors, and accelerometers) that people can carry on their body. Figure 4.1 shows the wearable ECG sensor network that provides high-resolution sensing of cardiac electrical dynamics [77]. Prior research showed that high-resolution ECG mapping substantially enhances the early detection of life-threatening events for cardiovascular patients [78, 79]. When continuously monitored and analyzed, wearable ECG sensor network offers an unprecedented opportunity to realize smart care of patients with high

risks (e.g., acute cardiac events) beyond the confines of, often high-end healthcare settings, and allow rural care centers to acquire sophisticated imaging and diagnostic capability at a fraction of capital investments.

However, it is not uncommon to encounter sensor failures in traditional sensor networks. For example, a subset of sensors often lose contact with the skin surface in ECG sensor networks because of body movements (see Figure 4.1). Maintaining strict skin contacts for hundreds of sensors is not only challenging but also greatly deteriorates the wearability of ECG sensor networks. Therefore, we propose a novel strategy named as "stochastic sensor network" in this present investigation. The stochastic sensor network means that the sensor network is functioning stochastically, in other words, allow a subset of sensors at varying locations within the network to transmit dynamic information intermittently. Notably, the new strategy of stochastic sensor networks is generally applicable in many other domains. For examples, wireless sensor network is often constrained by finite energy resources. Hence, optimal scheduling of activation and inactivation of sensors is imperative to realize long-term survivability and reliability of sensor networks. In addition, a subset of sensor nodes in a battlefield surveillance network may need to enter dormant states to avoid the reconnaissance from the enemy, but others continue working to detect real-time battlefield information. The locations of dormant sensors may also be stochastically varying so as to save energy and build a robust network for information visibility.

Realizing the full potential of stochastic sensor network hinges on the development of novel information-processing algorithms to support the design and exploit the uncertain information for decision making. Real-time distributed sensing generates spatially-temporally big data, which contains rich information of evolving dynamics of complex systems. Further, stochastic sensor network brings greater levels of uncertainty and complexity which pose

significant challenges for information extraction and decision making. This paper presents a new approach of sparse particle filtering to model spatiotemporal dynamics in the big data from the stochastic sensor network. We first leverage the cross-section data (e.g., time t_i in Figure 4.2) to develop a sparse kernel-weighted regression model to achieve a parsimonious representation of spatial patterns. Further, the parameters of spatial model are transformed into a reduced-dimension space, and thereby sequentially updated with the recursive Bayesian estimation when new sensor observations are available at time t_{i+1}. Thus, spatial and temporal processes closely interact with each other. Experimental results on different scenarios of stochastic sensor networks (i.e., spatially, temporally, and spatiotemporally dynamic networks) show the effectiveness of sparse particle filtering to support the stochastic design and harness the uncertain information for decision making.

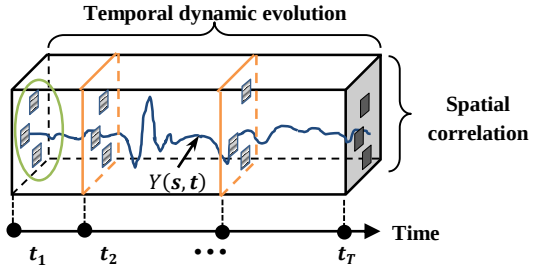


Figure 4.2 Space-time data generated from distributed sensor network.

This paper is organized as follows: Section 4.2 presents the research background. Section 4.3 introduces the research methodology. Section 4.4 presents materials and experimental design. Section 4.5 presents experimental results. Section 4.6 discusses and concludes this investigation.

4.2 Research Background

Wireless sensor network has broad applications in healthcare, environment, logistics, defense and many other areas. However, very little work has been done to design the stochastic sensor network and further develop new analytical methodologies to exploit spatiotemporal data for information extraction and knowledge discovery. For example, ECG sensor network has the

promise to provide high-resolution sensing of space-time cardiac electrical dynamics (see Figure 4.1). However, there are remarkable impetuses towards the use of wearable ECG systems to monitor cardiac activity in the past few decades. The ECG sensor network uses hundreds of electrodes to obtain electrical measurements on the body surface. Each electrode is required to maintain rigid contact with the skin surface, thereby adversely affecting wearability – which has been a stumbling block to widespread applications of ECG sensor networks from research laboratory to healthcare settings.

As the movement of human body is highly dynamic, few, if any, previous studies have considered the new design of stochastic ECG sensor network that allows stochastic sensor-skin contacts. As such, a subset of ECG electrodes will capture cardiac electrical activity intermittently at dynamically varying locations within the network. However, if we lower the hardware requirement on sensor-skin contacts, new analytical algorithms are urgently needed to support the stochastic design and handle data uncertainty. Notably, this new idea of stochastic sensor network received few attention in the past, partly due to traditional designs that sacrifice the wearability for data quality, and partly to the lack of analytical algorithms for spatiotemporal data modeling under uncertainty.

These gaps pose significant technological barriers for realizing the design of stochastic sensor network. Figure 4.2 is an illustration of spatiotemporal data generated from the distributed sensing. Each cross-section represents a snapshot of the underlying complex process at a specific time. As the dynamics of complex systems vary across both space and time, sensor networks give rise to spatiotemporal data: $\{\mathrm{Y}(\boldsymbol{s},t)\colon \boldsymbol{s}\in R\subset\mathbb{R}^{d}, t\in T\}$, where the dependence of spatial domain R on time T symbolizes the changes of spatial domain over time. Traditionally, spatiotemporal data is characterized and modeled in two ways: (i) spatially-varying time series model $Y(\boldsymbol{s},t)\ =$

$Y_{\boldsymbol{s}}(t)$, which separates the temporal analysis for each location; (ii) temporally-varying spatial model $Y(\boldsymbol{s}, t) = Y_t(\boldsymbol{s})$, which separates spatial analysis for each time point. The first model $Y_{\boldsymbol{s}}(t)$ shows specific interests in time-dependent patterns for each sensor, and allows for sensor-to-sensor analysis between time series. For example, Yang et al. studied ECG time series and exploited the useful information for medical decision making [2, 28, 80, 81]. The second model $Y_t(\boldsymbol{s})$ focuses more on space-dependent patterns for each time point. For example, Zarychta et al. studied spatial patterns in each ECG image for the detection of myocardial infarctions [82]. However, both approaches are conditional methods studying either the space given time or time given space, and are limited in their capabilities to characterize and model space-time correlations.

Space-time interactions bring substantial complexity in the scope of modeling, because of spatial correlation, temporal correlation, as well as how space and time interact. Notably, many previous works employed random fields in $\mathbb{R}^{\mathrm{d}+1}$ to model space and time dependencies [83, 84]. However, space and time are not directly comparable, because space does not have the past, present, and future and the spatial dimension is not comparable to temporal dimension. In the past few years, spatiotemporal modeling has received increased attentions due to the proliferation of data sets that are varying both spatially and temporally. Examples of application areas include brain imaging [85, 86], manufacturing [87], environment [88, 89], public health [90, 91], service equity [92] and socio-economics [93]. The specific questions include the analysis of time-varying brain image and fMRI data, nanowire growth modeling at multiple spatial scales, temporal movement of hurricane, geographical diffusion of pandemic infectious diseases, and spatial equity of public services.

However, very little work has been done to realize a highly resilient sensor network by developing new spatiotemporal algorithms. Traditional spatiotemporal methods are not concerned

with the uncertainty in stochastic sensor network but rather assume reliable sensor readings at fixed locations. For example, brain imaging data are homogeneous and synchronized in time for every pixel, but stochastic sensor networks involves heterogeneous data that are asynchronized and incomplete at dynamically-varying locations of the network. Therefore, this present investigation aims to develop new algorithms to realize a highly resilient sensing system, namely stochastic sensor network. Such a resilient sensing system will reduce the requirement in sensor reliability, handle data uncertainty and heterogeneity, and enable fast and recursive prediction of space-time information.

4.3 Research Methodology

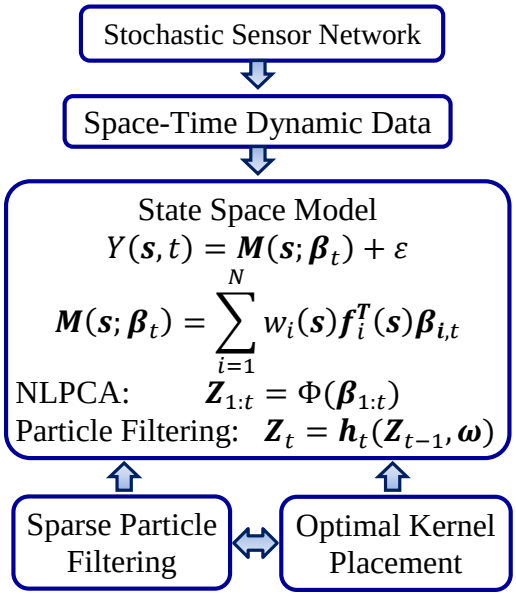


Figure 4.3 Flow chart of the proposed research methodology.

This paper presents the first-of-its-kind technology of stochastic sensor network. We developed a new approach of sparse particle filtering to characterize and model distributed sensing data that are non-homogeneous, asynchronized and incomplete. As shown in Figure 4.3, the proposed stochastic sensor network is supported by the algorithms of sparse particle filtering and optimal kernel placement. First, a state space formulation is utilized to model spatiotemporal

dynamic data generated by the stochastic sensor network. Second, an optimal kernel placement algorithm is developed to improve the compactness of kernel-weighted regression model of spatial patterns. In other words, we aims to minimize the number of kernels used for spatial representation. Third, we developed the new approach of sparse particle filtering model to reduce the dimensionality of parameters in spatial model and sequentially update the parameters when new observations are available at the next time point, thereby effectively modeling space-time dynamic data generated from stochastic sensor networks.

4.3.1 Spatial Modeling

It is well known that two sensors in the distributed sensor network tend to have a stronger correlation when they are closer to each other. Therefore, the spatial correlation between neighboring sensors is critical to estimate the observations of distributed sensors at a given time point.

As shown in Figure 4.4, the observations of body area sensor network in the front and back of the body are varying with respect to time. At a given time point, the approach here is to capture the spatial correlation in distributed sensor networks as a model of the form

$$Y(\boldsymbol{s}) = M(\boldsymbol{s};\boldsymbol{\beta}) + \varepsilon(\boldsymbol{s}) = \sum_{i=1}^{N} w_i(\boldsymbol{s})\boldsymbol{f}_i^T(\boldsymbol{s})\boldsymbol{\beta}_i + \varepsilon(\boldsymbol{s}) \quad (1)$$

where Y($\mathbf{s}$) is an observation taken at the location $\mathbf{s}$ in the body surface, N is the total number of kernel components, and $\varepsilon(\mathbf{s})$ is the Gaussian random noises. The $w_i(\mathbf{s})$ is a non-negative weighting kernel, i.e.,

$$w_i(\boldsymbol{s}) \propto |\boldsymbol{\Sigma}_i|^{-\frac{1}{2}} exp\left\{-\frac{(\boldsymbol{s}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{s}-\boldsymbol{\mu}_i)}{2}\right\} \quad (2)$$

where $\boldsymbol{\mu}_i$ is the center and $\boldsymbol{\Sigma}_i$ is the covariance function of the ith kernel component. The $\boldsymbol{f_i}(\boldsymbol{s}) = \left(f_{i1}(\boldsymbol{s}), \cdots, f_{ip}(\boldsymbol{s})\right)^T$ is a set of known basis functions, e.g., $\boldsymbol{f}_i(\boldsymbol{s}) = (1, x, y)^T$ in this present

investigation. Here, x, y are the spatial coordinates of the location $\mathbf{s}$. Notably, $\boldsymbol{\beta}_{\mathrm{i}} = (\beta_{\mathrm{i}1}, \ldots, \beta_{\mathrm{ip}})^T$ is a vector of model parameters. The dimensionality of $\boldsymbol{\beta}$ is $N \times p$, i.e., the multiplication of the number of kernels w_i (i.e., N) and the number of basis functions $\boldsymbol{f}_i$ (i.e., p).

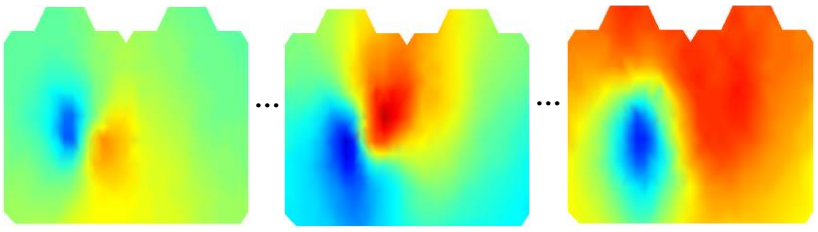


Figure 4.4 Time-varying ECG imaging on the body surface.

Note that the spatial model is designed as a locally weighted mixture of kernel regressions. If we observe $\mathbf{Y} = \big(\mathrm{Y}(\mathbf{s}_1), \ldots, \mathrm{Y}(\mathbf{s}_{\mathrm{K}})\big)^T$ for locations $\mathbf{s}_1, \ldots, \mathbf{s}_{\mathrm{K}}$ at a specific time point, the model can be written in a matrix form as:

$$\boldsymbol{Y} = \begin{bmatrix} Y(\boldsymbol{s}_1) \\ \vdots \\ Y(\boldsymbol{s}_K) \end{bmatrix} = \Psi\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$= \begin{bmatrix} w_1 f_{11}|\boldsymbol{s}_1 & \cdots & w_1 f_{1\mathrm{p}}|\boldsymbol{s}_1 & \cdots & w_N f_{N1}|\boldsymbol{s}_1 & \cdots & w_N f_{N\mathrm{p}}|\boldsymbol{s}_1 \\ w_1 f_{11}|\boldsymbol{s}_2 & & w_1 f_{1\mathrm{p}}|\boldsymbol{s}_2 & & w_N f_{N1}|\boldsymbol{s}_2 & & w_N f_{N\mathrm{p}}|\boldsymbol{s}_2 \\ \vdots & & \vdots & & \vdots & & \vdots \\ & \vdots & & \ddots & & \vdots & \\ w_1 f_{11}|\boldsymbol{s}_K & \cdots & w_1 f_{1\mathrm{p}}|\boldsymbol{s}_K & \cdots & w_N f_{N1}|\boldsymbol{s}_K & \cdots & w_N f_{N\mathrm{p}}|\boldsymbol{s}_K \end{bmatrix} \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_{13} \\ \vdots \\ \beta_{Np} \end{bmatrix} + \boldsymbol{\varepsilon} \quad (3)$$

Once the observed locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K$ and kernel components $w_i, i = 1, \ldots, \mathrm{N}$ are determined, the matrix Ψ is determined. Therefore, the model parameters $\boldsymbol{\beta}$ can be estimated by solving the function $\mathbf{Y} = \boldsymbol{\Psi\beta} + \boldsymbol{\varepsilon}$. For an unobserved location $\tilde{\mathbf{s}}$, we can predict $\mathrm{Y}(\tilde{\mathbf{s}})$ as $\tilde{\boldsymbol{\Psi}}\boldsymbol{\beta}$ giving the known observations $\mathrm{Y}(\mathbf{s}_1), \ldots, \mathrm{Y}(\mathbf{s}_{\mathrm{K}})$ at distributed locations.

Notably, this is the kernel-weighted regression model of spatial data at a specific time point. To consider temporal dynamic effects, we augment the spatial model to include temporal components, i.e., defining the time-varying model parameters as $\boldsymbol{\beta}_{\mathbf{t}} =$

$\left(\beta_{11}, \ldots, \beta_{1p}, \ldots, \beta_{N1}, \ldots, \beta_{Np}\right)_t^T$ at time t. As shown in Figure 4.4, as distributed sensor observations are varying in both space and time, model parameters vary with respect to time, i.e., $\mathrm{M}(\mathrm{s}; \boldsymbol{\beta_t}), \mathrm{M}(\mathrm{s}; \boldsymbol{\beta_{t+1}}), \cdots$. Then, we develop a particle filter to establish temporal correlation to recursively estimate the state of parameters, i.e., link the parameters $\boldsymbol{\beta}_\mathrm{t}$ over time by the evolution equation:

$$\boldsymbol{\beta}_t = g_t(\boldsymbol{\beta}_{t-1}, \gamma) \tag{4}$$

where $\mathrm{g_t}$ is the nonlinear evolution model and γ is process noise. Thus, the spatial model at a given time is defined as:

$$\boldsymbol{Y}_t = \boldsymbol{\Psi}_t \boldsymbol{\beta}_t + \varepsilon \tag{5}$$

The temporal filter is responsible for projecting forward (in time) current estimates of state and error covariance to obtain the estimates for the next time step. The spatial update is responsible for the feedback, i.e., incorporating new observations to derive posterior estimations. Although the temporal part is conceptually similar to general filtering approaches in time series analysis, the difference is that spatial and temporal processes closely interact with each other.

However, there are two practical issues pertinent to the construction of spatiotemporal models. (1) How to optimally place kernels in the design space? Traditionally, kernels are uniformly distributed in the space. However, this unavoidably leads to a very complex model. The objective here is to identify a parsimonious set of kernels that are sufficient to achieve optimal model performances. Section 4.3.2 will detail our developed algorithms for optimal kernel placement. (2) How to identify latent variables and establish the reduced-dimension particle filtering? It may be noted that the dimensionality of parameters $\boldsymbol{\beta}_t$ is large, and the $\boldsymbol{\beta}'s$ have certain nonlinear-correlation structures. In particular, sensor observations are highly nonlinear and non-stationary in both spatial and temporal domains. The objective here is to identify a compact

set of independent latent variables that suffice to represent the nonlinear dynamic process of temporal parameter variations. Section 4.3.3 will detail the sparse particle filtering model.

4.3.2 Optimal Kernel Placement

The objective of optimal kernel placement is to identify a compact set of kernels that minimize the representation error:

$$\text{argmin}_N \left[\left\| \boldsymbol{Y}(s,t) - \sum_{i=1}^{N} w_i(s,\boldsymbol{\theta_i}) \boldsymbol{f}_i^T(s) \boldsymbol{\beta}_i(t) \right\|, \{\boldsymbol{\theta_i}\} \right] \tag{6}$$

In equation (2), it may be noted that each kernel $w_i(\boldsymbol{\theta_i})$ contains free parameters $\boldsymbol{\theta_i} = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\mu}_i$ is the center and $\boldsymbol{\Sigma}_i$ is the covariance function, i.e., $\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_x^2(i) & \rho(i)\sigma_x(i)\sigma_y(i) \\ \rho(i)\sigma_x(i)\sigma_y(i) & \sigma_y^2(i) \end{bmatrix}$. In total, there are five parameters for each kernel $w_i(\boldsymbol{\theta_i})$, i.e., $\boldsymbol{\theta_i} = \{\mu_x(i), \mu_y(i), \rho(i), \sigma_x(i), \sigma_y(i)\}$. In order to identify a sparse set of kernels, we developed the greedy search algorithm to optimally select the kernel center $(\mu_x(i), \mu_y(i))$ in the search space, and then fine-tune the covariance matrix $\boldsymbol{\Sigma}_i$ at the chosen location by Levenberg-Marquardt method.

Figure 4.5 shows the algorithm of optimal kernel placement. First, we define the search space of kernels as $\Theta = \{(\boldsymbol{\mu}, \boldsymbol{\Sigma})\}$, and $\mathcal{A} = \{\boldsymbol{\theta_i}\}_{i=1}^N \subset \Theta$ is the selected set of kernels. The objective function is $\mathcal{R}(\mathcal{A}) = \left\| \boldsymbol{Y} - \sum_{i=1}^N w_i(\boldsymbol{\theta_i}) \boldsymbol{f}_i^T \boldsymbol{\beta}_i \right\|$ with two important and intuitive properties:

(1) *Monotonicity*: $\mathcal{R}(\mathcal{A}) \leq \mathcal{R}(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B}$. Hence, adding kernels decreases the objective function.

(2) *Submodularity*: Adding a new kernel to a small kernel set $\mathcal{A}$ provides more advantage than adding it to a larger kernel set $\mathcal{B} \supseteq \mathcal{A}$. In other words, for a new kernel with the parameter $\boldsymbol{\theta} \notin \mathcal{B} \supseteq \mathcal{A}$, it holds that $\mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}\}) \geq \mathcal{R}(\mathcal{B}) - \mathcal{R}(\mathcal{B} \cup \{\boldsymbol{\theta}\})$.

As a result, the problem of optimal kernel placement is reduced to the minimization of submodular functions, subject to the constraint on a sparse set of kernels. Minimizing the submodular function is well-known to be NP-hard [94]. The greedy algorithm is a commonly used heuristic, which starts with the empty set $\mathcal{A} = \emptyset$, and iteratively adds the new kernel $w(\boldsymbol{\theta}_i)$, $\mathcal{A} = \mathcal{A} \cup \{\boldsymbol{\theta}_i\}$ that maximizes the marginal benefit:

$$\boldsymbol{\theta}_i = \underset{\boldsymbol{\theta} \in \Theta \setminus \mathcal{A}}{\operatorname{argmax}} \mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}\}) \tag{7}$$

The algorithm stops when N kernels are selected. For each kernel location $\boldsymbol{\mu}$, the covariance matrix $\boldsymbol{\Sigma}$ is further fine-tuned to maximize the marginal benefit. Here, we adopted the Levenberg-Marquardt update [95] to optimize the covariance matrix $\boldsymbol{\Sigma}$ that minimizes the sum of squared errors between the data $\{Y(\boldsymbol{s}_k, t)\}$ and the model output $\{\hat{Y}(\boldsymbol{s}_k, t; \boldsymbol{\Sigma})\}$as:

$$\chi^2(\boldsymbol{\Sigma}) = \frac{1}{2} \sum_{k=1}^{K} \sum_{t=1}^{T} \left(Y(\boldsymbol{s}_k, t) - \hat{Y}(\boldsymbol{s}_k, t; \boldsymbol{\Sigma}) \right)^2 = \frac{1}{2} \boldsymbol{Y}^T \boldsymbol{Y} - \boldsymbol{Y}^T \hat{\boldsymbol{Y}} + \frac{1}{2} \hat{\boldsymbol{Y}}^T \hat{\boldsymbol{Y}} \tag{8}$$

The gradient of the sum of squared errors with respect to the parameters is:

$$\begin{gathered} \frac{\partial}{\partial \boldsymbol{\Sigma}} \chi^2 = \left(Y - \hat{Y}(\boldsymbol{\Sigma}) \right)^T \frac{\partial}{\partial \boldsymbol{\Sigma}} \left(Y - \hat{Y}(\boldsymbol{\Sigma}) \right) = - \left(Y - \hat{Y}(\boldsymbol{\Sigma}) \right)^T J \\ J = \left[\partial \hat{Y} / \partial \boldsymbol{\Sigma} \right] \end{gathered} \tag{9}$$

where the Jacobian matrix represents the local sensitivity of the model to variations of covariance parameters. The Levenberg-Marquardt method adaptively updates the covariance parameters with the perturbation $\Delta\boldsymbol{\Sigma}$ between the gradient descent and Gauss-Newton update.

$$\Delta \boldsymbol{\Sigma} = [J^T J + \lambda \operatorname{diag}(J^T J)]^{-1} J^T \left(Y - \hat{Y}(\boldsymbol{\Sigma}) \right) \tag{10}$$

where λ is the learning rate. The learning process stops when convergence is achieved with the following three criteria: (1) Convergence in parameters, $\max\left(\left| \Delta \boldsymbol{\Sigma}^{\mathrm{i}} / \boldsymbol{\Sigma}^{\mathrm{i}} \right| \right) < \epsilon$; (2) Convergence in the gradient, $\max\left(\left| J^T \left(\mathrm{Y} - \hat{\mathrm{Y}}(\boldsymbol{\Sigma}) \right) \right| \right) < \epsilon$; (3) Convergence in $\chi^2(\boldsymbol{\Sigma})$, $\chi^2 < \epsilon$.

Initialization:

$\Theta = \{\boldsymbol{\theta}|(\boldsymbol{\mu}, \boldsymbol{\Sigma})\}$ *// search space*

$\mathcal{A} = \emptyset$ *// selected set of kernels*

$i = 1$

While $i \leq N$

$\mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}\}) = \left\| \boldsymbol{Y} - \left(\sum_{j=1}^{i-1} w_j(\boldsymbol{\theta}_j) \boldsymbol{f}_j^T \boldsymbol{\beta}_j + w(\boldsymbol{\theta}) \boldsymbol{f}^T \boldsymbol{\beta} \right) \right\|$

$\boldsymbol{\theta}_i = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}} \mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}\})$

//the ith selected kernel location

$\boldsymbol{J} = \left[\partial \widehat{\boldsymbol{Y}} / \partial \boldsymbol{\Sigma} \right]$ *//Jacobian matrix*

// Levenberg-Marquardt update of covariance function

$\Delta \Sigma^{\mathrm{i}} = [\boldsymbol{J}^T \boldsymbol{J} + \lambda \mathrm{diag}(\boldsymbol{J}^T \boldsymbol{J})]^{-1} \boldsymbol{J}^T \left(\boldsymbol{Y} - \widehat{\boldsymbol{Y}}(\Sigma^{\mathrm{i}}) \right)$

$\Sigma^{\mathrm{i}+1} = \Sigma^{\mathrm{i}} + \Delta \Sigma^{\mathrm{i}}$ until convergence

// Update the selected set of kernels

$\mathcal{A} = \mathcal{A} \cup \{\boldsymbol{\theta}_i\}$

$\Theta = \Theta \backslash \{\boldsymbol{\theta}_i\}$

$i = i + 1$

End

Figure 4.5 Optimal kernel placement algorithm.

However, the greedy algorithm is very expensive for a large search space Θ. Therefore, we exploit the submodularity of the objective function $\mathcal{R}(\mathcal{A})$ to achieve a scaling-up implementation, namely "lazy" greedy algorithm [96]. Initially, we compute the marginal benefit as $\delta(\boldsymbol{\theta}) = \mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}\})$, and construct a priority heap structure. At each iteration, the kernel with the largest benefit is popped from the heap and included in $\mathcal{A}$. The inclusion of this kernel affects the benefits of the remaining kernels in the heap. The traditional way is to update the heap at each iteration. However, due to the submodular property, a more efficient way is to only update the marginal benefit of the top kernel in the heap. Therefore, we update the benefit of the top kernel $w(\boldsymbol{\theta}^*)$ as $\delta(\boldsymbol{\theta}^*) = \mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}^*\})$ in each iteration, and compared with the benefits of all kernels in the heap. If $\delta(\boldsymbol{\theta}^*)$ remains to be the largest benefit among $\delta(\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta \backslash \mathcal{A}$, then it will be selected. Otherwise, $\delta(\boldsymbol{\theta}^*)$ will be moved downward in the priority heap and the benefit of the next top kernel will be recalculated. As such, a compact set of kernels are sequentially selected to minimize the representation error.

Notably, the "lazy" greedy algorithm does not require the update of marginal benefits for all kernels in the heap at every iteration. In addition, it enables the selection of an optimal kernel with the largest marginal benefit through a "lazy" evaluation of objective function $\mathcal{R}(\mathcal{A})$. The integration of "lazy" greedy algorithm with Levenberg-Marquardt method effectively maximizes the marginal benefit $\delta(\theta) = \mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A} \cup \{\boldsymbol{\theta}\})$ in each iteration of the learning process.

4.3.3 Sparse Particle Filtering

As shown in Section 4.3.1, the spatiotemporal model is formulated as:

$$\begin{aligned} \mathbf{Y}_t(\mathbf{s}) &= \boldsymbol{\Psi}_t(\mathbf{s})\boldsymbol{\beta}_t + \varepsilon \\ \boldsymbol{\beta}_t &= g_t(\boldsymbol{\beta}_{t-1}, \gamma) \end{aligned} \tag{11}$$

The alternative probabilistic representation of this model is $p(\boldsymbol{Y}_t|\boldsymbol{\beta}_t)$ for the observation equation and $p(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1})$ for the equation of parameter evolution. When new observations Y_t are available from the distributed sensor network, $p(\boldsymbol{\beta}_t|\boldsymbol{Y}_t)$ can be estimated based on the Bayesian theory. Further, the parameters $\boldsymbol{\beta}_{t+1}$ at the next time point can be predicted based on the equation of parameter evolution. The particle filtering includes two sequential steps. The estimation step provides the posterior probability $p(\boldsymbol{\beta}_t|Y_{1:t})$, while the prediction step provides the prior probability $p(\boldsymbol{\beta}_{t+1}|Y_{1:t})$ for the estimation step. This process is recursively updated over time:

(1) *Estimation*: The posterior density estimation of $p(\boldsymbol{\beta}_t|\boldsymbol{Y}_t)$ via Bayesian theory is represented as

$$p(\boldsymbol{\beta}_t|Y_{1:t}) = \frac{p(Y_t|\boldsymbol{\beta}_t)}{p(Y_t|Y_{1:t-1})} p(\boldsymbol{\beta}_t|Y_{1:t-1}) \tag{12}$$

where $p(Y_t|\boldsymbol{\beta}_t)$ provides the likelihood for observation Y_t given parameters $\boldsymbol{\beta}_t$. The posterior distribution of parameters $p(\boldsymbol{\beta}_t|Y_{1:t})$ is estimated by maximizing the likelihood function $p(Y_t|\boldsymbol{\beta}_t)$, based on the available observations $\boldsymbol{Y}_{1:t}$ up to time t and the prior distribution of $p(\boldsymbol{\beta}_t|Y_{1:t-1})$,

(2) *Prediction*: When the posterior distribution of $p(\boldsymbol{\beta}_t|Y_{1:t})$ is obtained via the estimation step, the parameters at the next time point $\boldsymbol{\beta}_{t+1}$ can be predicted conditionally on present observations $\boldsymbol{Y}_{1:t}$ as

$$p(\boldsymbol{\beta}_{t+1}|\boldsymbol{Y}_{1:t}) = \int p(\boldsymbol{\beta}_{t+1}|\boldsymbol{\beta}_t)p(\boldsymbol{\beta}_t|\boldsymbol{Y}_{1:t})\, d\boldsymbol{\beta}_t \tag{13}$$

As such, the parameters $\boldsymbol{\beta}_{t+1}$ at time point $t+1$ can be predicted based on observations $Y_{1:t}$, which provides the prior probability $p(\boldsymbol{\beta}_{t+1}|Y_{1:t})$ for the estimation step.

However, the computation in spatiotemporal modeling equations (11-13) poses significant challenges as follows:

(1) *Nonlinear and non-Gaussian properties*:

It is not uncommon that spatiotemporal data from distributed sensing of complex systems are highly nonlinear and nonstationary. Hence, the function $\boldsymbol{\beta}_t = g_t(\boldsymbol{\beta}_{t-1}, \gamma)$ is in a nonlinear form and $p(\boldsymbol{\beta}_t|\boldsymbol{Y}_{1:t})$ is a non-Gaussian distribution. Traditional methods, e.g., Kalman filter (KF), are only effective when the state space model is linear and the posterior density follows a Gaussian distribution. If these assumptions are not satisfied, the KF fails to establish an effective state space model. Therefore, we propose the particle filter (PF) to recursively compute the estimation and prediction steps, and approximate the posterior distribution of states with a large number of particles generated by sequential Monte Carlo sampling.

As shown in Figure 4.6, particle filtering represents the multimodal distribution by drawing a large number of samples from it, so that the density of samples in one area of the state space represents the probability of that region. As the number of particles increases, the Monte Carlo characterization is analogous to the usual functional description of posterior distribution. Particle filtering has the advantage to represent any arbitrary complex distribution, and is effective for non-Gaussian, multi-modal distributions.

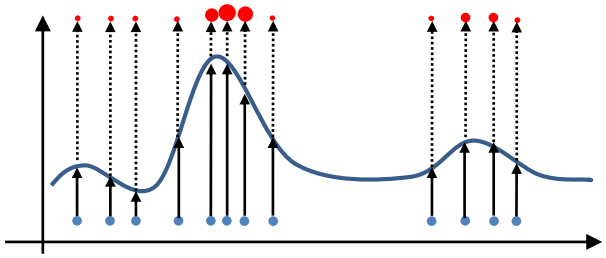


Figure 4.6 Representation of multi-modal distribution using importance sampling.

(2) *Curse of dimensionality*:

It is also important to note that the dimensionality of parameter vector, $\boldsymbol{\beta} = (\beta_{11}, \dots, \beta_{1p}, \dots, \beta_{N1}, \dots, \beta_{Np})$ is very large. If N kernels $w_i(\boldsymbol{s})$ and p basis functions $\boldsymbol{f}_i(\boldsymbol{s})$ are used in the spatiotemporal model, the dimensionality of $\boldsymbol{\beta}$ is $N \times p$, i.e., the multiplication of number of kernels and number of basis functions. This creates a serious "curse of dimensionality" problem for estimation and prediction steps in particle filtering. The high-dimensional states will easily cause both overfitting and ill-posed estimation problems. In particular, the $\boldsymbol{\beta}'$s have certain nonlinear-correlation structures. Now, an immediate question becomes "Can we identify a compact set of latent state variables from the high-dimensional parameter space, and thereby constructing a sparse particle filtering model of the nonlinear dynamic process?"

Notably, very little work has been done to develop a sparse particle filtering model that identifies a compact set of independent states that suffice to represent the nonlinear dynamic process. Prior research on particle filtering is limited in its capability in identifying latent factors. Principal component analysis (PCA) decomposes high-dimensional variables Y into linearly uncorrelated sources Z through a linear mapping A, i.e., $Y = AZ + v$. However, PCA is only capable of delineating linear components, and often fails to handle nonlinear correlation structures in high-dimensional variables. In addition, traditional PCA methods do not account for the dynamics of hidden factors even though complex systems evolve dynamically.

In this present investigation, we integrated the nonlinear PCA (NLPCA) with particle filtering for modeling space-time dynamics in distributed sensor networks. The NLPCA generalizes the principal components from straight lines to curves and thereby captures inherent structures of the nonlinear data by curved subspaces. As shown in Figure 4.7, instead of directly modeling the parameters $\boldsymbol{\beta}_t$ over time, we establish the sparse particle filter to model the time-varying principal components $\boldsymbol{Z}_t$. After the estimation step, the maximum posteriori probability (*MAP*) *estim*ate of $\boldsymbol{\beta}_t$ is obtained. It may be noted that the *MAP estim*ate of $\boldsymbol{\beta}_t$ is used in Eq. 6 to identify a compact set of kernels. Then, nonlinear PCA transforms the high-dimensional $\boldsymbol{\beta}_t$ to the low-dimensional latent variables $\boldsymbol{Z}$ by $\boldsymbol{Z} = \Phi(\boldsymbol{\beta})$. The temporal predictor equation will project forward (in time) the current state $\boldsymbol{Z}_t$ and error covariance estimates to obtain the estimates of $\boldsymbol{Z}_{t+1}$ for the next time point. To this end, inverse nonlinear PCA will recover the estimate of high-dimensional $\boldsymbol{\beta}_{t+1}$ from the low-dimensional latent variables $\boldsymbol{Z}_{t+1}$ by $\boldsymbol{\beta}_{t+1} = \Phi^{-1}(\boldsymbol{Z}_{t+1})$. The process is recursively updated by the interactions between estimation and prediction steps. For more information on nonlinear PCA, see for example [97].

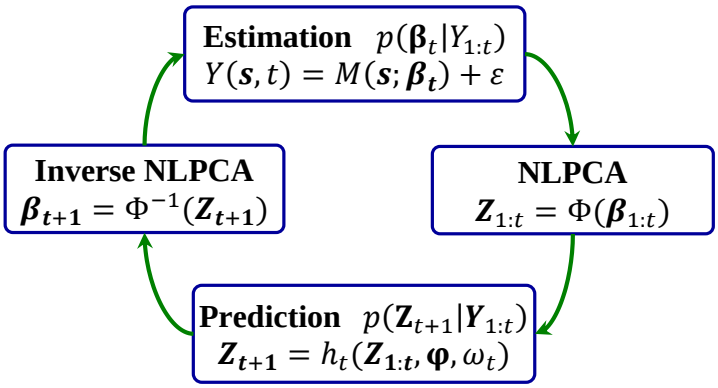


Figure 4.7 Sparse particle filtering in the reduced-dimension space.

Further, model structure $h_t(\cdot)$ and model parameters $\boldsymbol{\varphi}$ need to be determined and estimated in the nonlinear state space equation $\mathbf{Z}_{t+1} = h_t(\mathbf{Z}_{1:t}, \boldsymbol{\varphi}, \omega_t)$, where ω_t is white noise. As aforementioned, spatiotemporal data from distributed sensor network are highly nonlinear and

nonstationary. In this present investigation, we used the logistic model to model the nonlinear relationships of the hidden states, i.e.,

$$\begin{gathered}\mathbf{Z}_{t+1} = h_t(\mathbf{Z}_{1:t}, \boldsymbol{\varphi}, \omega_t) \\ \mathbf{Z}_{t+1} = \frac{a_1}{1+e^{-b_1 \mathbf{Z}_t}} + \frac{a_2}{1+e^{-b_2 \mathbf{Z}_{t-1}}} + \cdots + \frac{a_\tau}{1+e^{-b_\tau \mathbf{Z}_{t-\tau+1}}} + c\end{gathered} \tag{14}$$

where τ is the time lag, and $\boldsymbol{\varphi} = (a_1, \ldots, a_\tau, b_1, \ldots, b_\tau, c)$ are model parameters. To this end, we integrate the sequential Monte Carlo method with Kernel smoothing approach [98-100] to simultaneously estimate model states $\mathbf{Z}_{t+1}$ and model parameters $\boldsymbol{\varphi}$. Note that Bayesian theory provides the joint distribution as

$$\begin{gathered}p(\mathbf{Z_{t+1}}, \boldsymbol{\varphi_{t+1}} | Y_{1:t+1}) \\ = p(Y_{t+1} | \mathbf{Z_{t+1}}, \boldsymbol{\varphi_{t+1}}) p(\mathbf{Z}_{t+1} | \boldsymbol{\varphi_{t+1}}, Y_{1:t}) p(\boldsymbol{\varphi}_{t+1} | Y_{1:t})\end{gathered} \tag{15}$$

Parameter estimation (kernel smoothing):

$$p(\boldsymbol{\varphi}_{t+1} | Y_{1:t}) = \sum_{i=1}^{Q} w_t^{(i)} N(\boldsymbol{\varphi} | \boldsymbol{m}_t^{(i)}, v^2 \boldsymbol{V}_t)$$

Where $\boldsymbol{m}_t^{(i)} = a\boldsymbol{\varphi}_t^{(i)} + (1-a)\bar{\boldsymbol{\varphi}}_t$

$a = \sqrt{1-v^2}$ and $0 < v < 1$ is a constant

$\mathbf{V}_t$ is the variance matrix of $\left\{\boldsymbol{\varphi}_t^{(i)}\right\}_{i=1}^{Q}$

State estimation (sampled from the model):

$$\boldsymbol{Z}_{t+1}^{(i)} = h_t(\boldsymbol{Z}_{1:t}^{(i)}, \boldsymbol{\varphi}_{t+1}^{(i)}, \omega_t)$$

Weight update:

$$w_{t+1}^{(i)} \propto \frac{p\left(\mathbf{Y}_{t+1} \middle| \mathbf{Z}_{t+1}^{(i)}, \boldsymbol{\varphi}_{t+1}^{(i)}\right)}{p\left(\mathbf{Y}_{t+1} \middle| \boldsymbol{\pi}_{t+1}^{(i)}, \mathbf{m}_t^{(i)}\right)}$$

where $\boldsymbol{\pi}_{t+1}^{(i)} = E\left(\boldsymbol{Z_{t+1}} | \boldsymbol{Z}_t^{(i)}, \boldsymbol{\varphi}_t^{(i)}\right)$

Normalization: $\sum_i w_{t+1}^{(i)} = 1$

Figure 4.8 Procedures for simultaneous estimate of states and parameters.

The combined sample $\left\{\boldsymbol{Z}_t^{(i)}, \boldsymbol{\varphi}_t^{(i)}\right\}_{i=1}^{Q}$, and associated weights $\left\{w_t^{(i)}\right\}_{i=1}^{Q}$ are introduced to characterize the posterior $p(\boldsymbol{Z}_t, \boldsymbol{\varphi} | Y_{1:t})$ at time t. Note that the subscript of t on the $\boldsymbol{\varphi}$ samples

indicate the time t posterior, not that $\boldsymbol{\varphi}$ is time-varying. Also, the weights are normalized so that $\sum_i w_t^{(i)} = 1$. The kernel smoothing approach for the simultaneous estimate of states $\mathbf{Z}_{t+1}$ and model parameters $\boldsymbol{\varphi}$ is as shown in Figure 4.8. The posterior $p(\mathbf{Z_{t+1}}, \boldsymbol{\varphi_{t+1}}|Y_{1:t+1})$ is obtained through the iterative estimation of the sample $\left\{\mathbf{Z}_{t+1}^{(i)}, \boldsymbol{\varphi}_{t+1}^{(i)}\right\}_{i=1}^{Q}$ and associated weights $\left\{w_{t+1}^{(i)}\right\}_{i=1}^{Q}$ at the time point $t+1$. The proposed method of sparse particle filtering not only resolves the problem of "curse of dimensionality" but also effectively captures the space-time dynamics in big data from stochastic sensor networks.

4.4 Materials and Experimental Design

The proposed methodology is evaluated and validated using real-world data from ECG sensor networks, available in the PhysioNet database (http://www.physionet.org/) [40]. As shown in Figure 4.9, the body area sensor network consists of 120 torso-surface electrodes for recording the high-resolution ECG data. The sampling frequency of ECG sensors is 2 kHz. The time length of ECG mapping data is about 820 milliseconds, i.e., a full ECG cycle. Figure 4.9 shows detailed anatomic locations of 120 ECG electrodes in the body area sensor network.

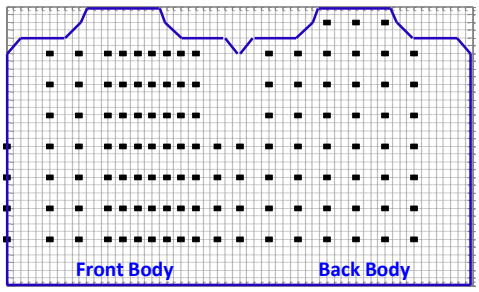


Figure 4.9 Anatomic locations of sensors in the wearable ECG sensor network.

First, we evaluated the performance of sparse particle filtering for characterizing and modeling the ECG potential mapping data. To improve the model compactness, we optimally placed the kernels in the spatial region with the "lazy" greedy algorithm and Levenberg-Marquardt

method. Further, a compact set of latent state variables is identified from the high-dimensional parameter space. Then, we developed a particle filtering model to recursively estimate and update the latent state variables for predicting nonlinear space-time dynamics over time. In particular, we compared the sparse particle filtering model with other alternative methods, including Kalman filtering (KF) and autoregressive moving average model (ARMA), to benchmark the performance for real-world applications in ECG sensor network.

Second, because stochastic ECG sensor networks employ a variable degree of contacts between the skin surface and sensors, we simulate three experimental scenarios of stochastic sensor network, namely, spatially, temporally, and spatiotemporally dynamic networks (also see Figure 4.10). Each simulation scenario is detailed as follows:

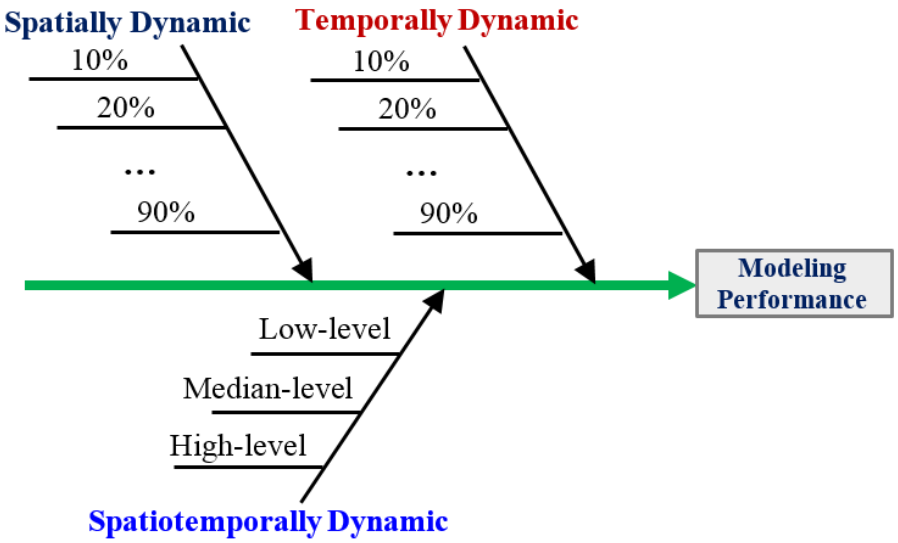


Figure 4.10 Experimental design of online relax-fit scenarios.

4.4.1 Spatially Dynamic Network

As shown in Figure 4.10, the scenario of spatially dynamic network includes 9 levels from 10% to 90%. For each level, the simulation deactivates or disconnects a certain percentage of electrodes out of 120 for the whole time interval of 820 ms. For example, there will be 12 electrodes (i.e., randomly selected out of 120) inactive for the whole recording period in the level

of 10%. We will replicate the experiments for 100 times by randomly choosing the locations of 12 inactive electrodes.

4.4.2 Temporally Dynamic Network

Similarly, the scenario of temporally dynamic network has 9 levels from 10% to 90%. For each level, every sensor is inactivated or disconnected from the ECG network for a certain percentage of time that is randomly chosen within the whole recording period. For example, the level of 10% temporally inactive network indicates there will be 10% of time points for each sensor to be inactive in the network. We will replicate the experiments for 100 times by randomly choosing 10% temporally missing values for each sensor in the network.

4.4.3 Spatiotemporally Dynamic Network

The third scenario of spatiotemporally dynamic network will simulate a random subset of sensors to transmit information intermittently at dynamically varying locations within the network of sensors. Here, we extended the idea of stochastic Kronecker graphs [101] to generate the time-varying dynamic network with a non-standard matrix operation, i.e., Kronecker product. Notably, prior work shows that Kronecker graphs can effectively model statistical properties of real networks, including static properties (heavy-tailed degree distribution, small diameter) and temporal properties (densification, shrinking diameter). The Kronecker product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of sizes $N \times M$ and $K \times L$ is defined as

$$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1m}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\boldsymbol{B} & \cdots & a_{nm}\boldsymbol{B} \end{bmatrix} \tag{16}$$

where $\boldsymbol{A} = [a_{ij}], i = 1, \dots, N, j = 1, \dots, M$ and $\boldsymbol{B} = [b_{ij}], i = 1, \dots, K, j = 1, \dots, L$. To simulate a stochastic Kronecker graph, the process starts with a probability matrix $\boldsymbol{K}$ and then iterates the Kronecker product with itself for k times.

As shown in Figure 4.11, we used the initiator matrix $\boldsymbol{K'}$ of size 2×2 with all elements 0.5 and a random perturbation matrix with elements $\varepsilon \sim N(0, 0.02^2)$. The probability matrix $\boldsymbol{K}$ is calculated by adding the random matrix $\boldsymbol{\varepsilon}$ to the initiator matrix $\boldsymbol{K'}$. Then we iterate the Kronecker product of the probability matrix $\boldsymbol{K}$ for n times to compute the n^{th} Kronecker power $\boldsymbol{K}^{[n]}$. Each entry p_{ij} of $\boldsymbol{K}^{[n]}$ indicates the existence of an edge with the probability p_{ij}. Figure 4.11 shows the second Kronecker power $\boldsymbol{K}^{[2]}$, and the network adjacency matrix is obtained by thresholding the matrix $\boldsymbol{K}^{[2]}$. The active sensor nodes are those connected by an edge in the network, while the sensors without edge connections are inactive.

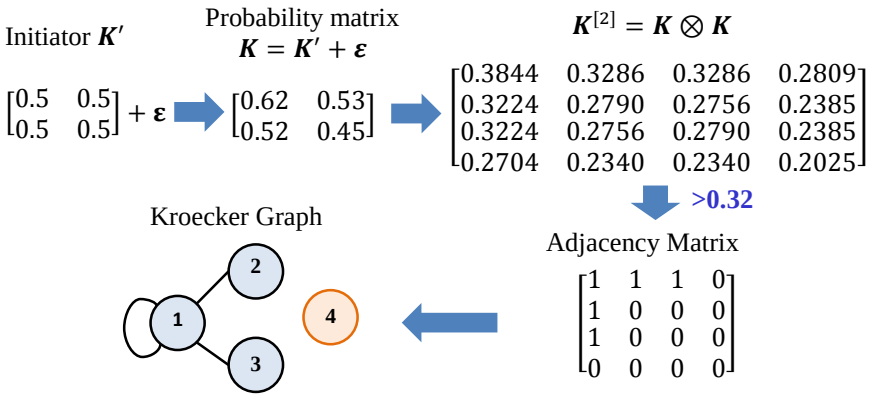


Figure 4.11 Stochastic Kronecker graph.

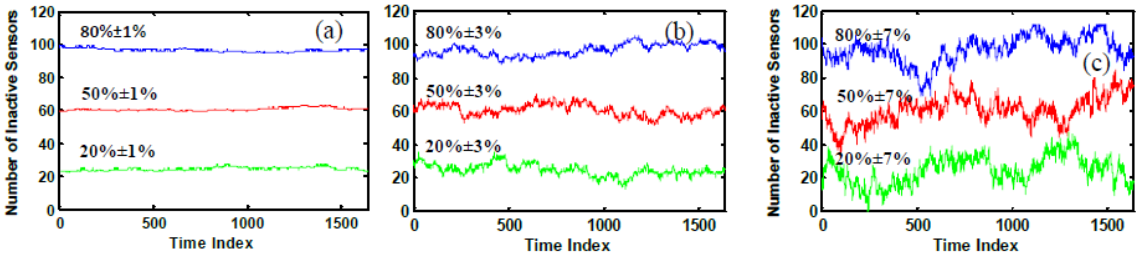


Figure 4.12 Stochastic Kronecker graph based simulations of spatiotemporally dynamic network.

In order to simulate the spatiotemporally dynamic network, we randomly generate the perturbation matrix $\boldsymbol{\varepsilon}$ at each time point and thereby create time-varying Kronecker power matrices $\boldsymbol{K}^{[7]}$. In addition, we designed three levels of thresholds (20%, 50% and 80%) to simulate

low-, median- and high-level dynamics of sensor-skin contacts, each of which also has three degrees of variations (1%, 3% and 7%). Figure 4.12 shows the simulated variations of inactive sensors in the dynamic network. Note that not only the number of inactive sensors but also their locations are varying with respect to time.

To evaluate and validate the proposed methodology, we utilized the relative prediction error (RPE) as the performance metric, which is computed as the ratio of mean absolute error (MAE) and mean magnitude (MM). The MAE is a quantity that measures the discrepancy between real-world data and model predictions under the variety of experimental scenarios as aforementioned, and is defined as:

$$MAE = \frac{1}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T}\left|Y(s_k,t) - \hat{Y}(s_k,t)\right| \tag{17}$$

The mean magnitude of observations from the body area sensor network is defined as:

$$MM = \frac{1}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T}|Y(s_k,t)| \tag{18}$$

4.5 Experimental Results

4.5.1 Optimal Kernel Placement

In Section 4.3.2, we propose to identify a parsimonious set of kernels that are sufficient to achieve optimal model performance. Figure 4.13(a) shows the model performance with respect to sequentially placed kernels. Here, the overall RPE is calculated using the ECG mapping data throughout the whole recording period. As the number of kernels is increasing, a monotonically decreasing trend of prediction errors is achieved. The increment of model performance is large when the number of kernels is increased from 1 to 12. After 12 kernels, the resulting improvement on model performance becomes smaller and smaller. This phenomenon is also called "diminishing returns". After 20 kernels, the overall RPE reaches a stable period and is less than 2%. Notably,

the overall RPE achieves 0.97% for 60 optimally placed kernels, which indicates an effective model for the representation of ECG mapping data.

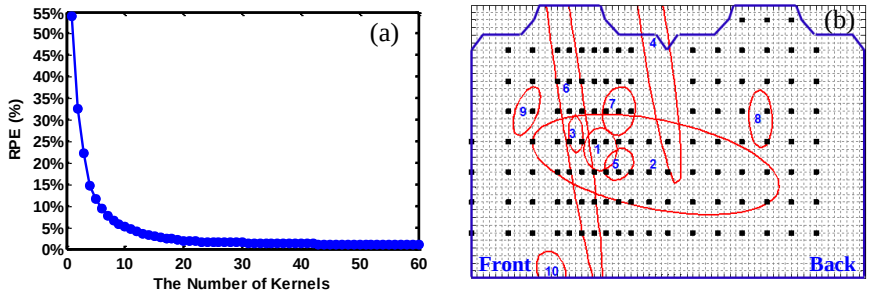


Figure 4.13 (a) Model performances for sequentially placed kernels, and (b) Location and dispersion of the first 10 placed kernels.

Figure 4.13(b) shows the locations and dispersions of the first 10 placed kernels. Most of kernels are centered in the middle area of the front body, which is close to the location of heart. Notably, the second, the fourth and the sixth kernels have bigger dispersions than other kernels. The second and sixth kernels mainly accounts for the large variations of electrical potentials in the body surface due to ventricular depolarization and repolarization. In addition, the fourth one is dealing with significant variations between the front and back of the body. The spatial model of ECG image is a locally weighted mixture of kernel regressions (also see equations (1) and (3)).

Figure 4.14 shows the representation performances of the proposed model with 60 kernels (180 parameters) at two time points (i.e., resting state before P wave and R peak). Blue dots are real-world measurements from 120 electrodes in the ECG sensor network. The contour of predicted surface is shown under the 3D colored surface with labels indicating the magnitudes. When the heart is in the resting state before P wave (see Figure 4.14(a)), the ECG magnitude is within a small range from -0.3 to 0.18. Notably, the RPE reaches 0.40% at this specific time point, in spite of high variability in the space. However, the ECG magnitude is within a bigger range from -20 to 18 at the time point of R peak (see Figure 4.14(b)). The model RPE is about 2.99%.

The small prediction errors (<3%) demonstrated the effectiveness of spatiotemporal model in representing the time-varying ECG mapping data.

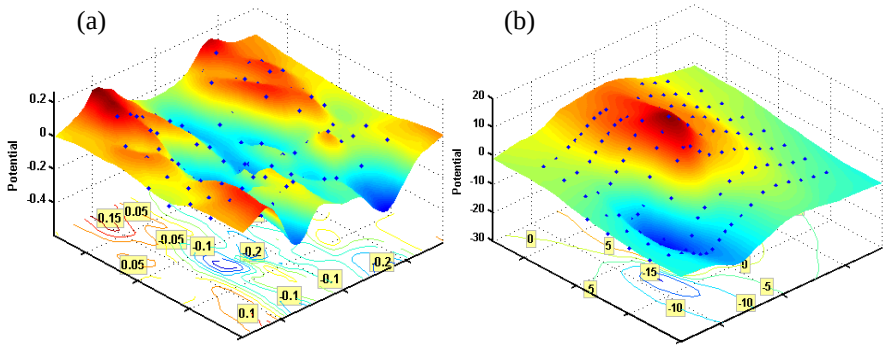


Figure 4.14 Model performance of 60 kernels at two time points. (a) resting state before P wave (RPE=0.40%) and (b) R peak (RPE=2.99%, blue dots are data from 120 electrodes).

4.5.2 Sparse Particle Filtering Model

Note that 60 optimally placed kernels yields the overall RPE of 0.97%, but a total of 180 parameters are involved in the spatial model. This leads to the problem of "curse of dimensionality" in process modeling. In section 4.3.3, we proposed the new idea to reduce the dimensionality of model parameters with NLPCA, and reconstruct the parameter values by inverse NLPCA. Hence, we reduced the high-dimensional set of parameters (i.e., 180) to a low-dimensional set of latent variables that represent the hidden states.

However, the first question is to identify an optimal dimensionality of latent variables. Figure 4.15 shows the Pareto chart of the variations of R^2 with respect to the number of principal components. Notably, nonlinear PCA transforms the high-dimensional $\boldsymbol{\beta}$ (180 parameters) to the low-dimensional latent variables $\boldsymbol{Z}$ by $\boldsymbol{Z} = \Phi(\boldsymbol{\beta})$. The inverse nonlinear PCA recovers the estimate $\widetilde{\boldsymbol{\beta}}$ from latent variables $\boldsymbol{Z}$ by $\widetilde{\boldsymbol{\beta}} = \Phi^{-1}(\boldsymbol{Z})$. Hence, we calculated the variations of R^2 values with respect to the dimensionality of latent variables $\boldsymbol{Z}$.

$$R^2 = 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}} = 1 - \frac{\|\boldsymbol{\beta} - \Phi^{-1}(\boldsymbol{Z})\|}{\|\boldsymbol{\beta} - \overline{\boldsymbol{\beta}}\|}$$

where SS_{residual} is the residual sum of squares, and SS_{total} is the total sum of squares. Figure 4.15 shows that the first principal component (PC) achieves the R^2 value of 53.8% and the first 10 PCs reaches the total R^2 value of 95.6%. As such, we selected the low-dimensional set of 10 PCs to construct the sparse particle filtering model in this present case study.

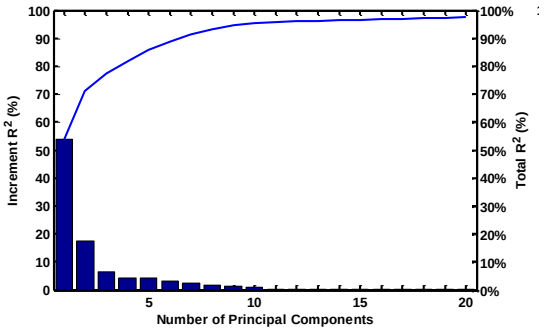


Figure 4.15 The Pareto chart of the R^2 explained by principal components.

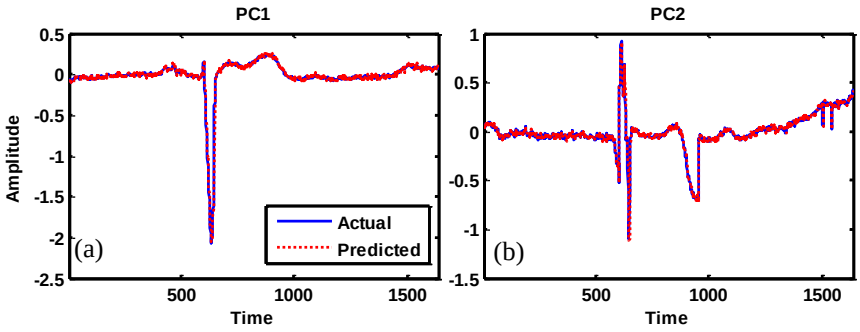


Figure 4.16 Predicted result for principal components (PCs) using particle filtering. (a) The first PC. (b) The second PC.

Figure 4.16 shows the effectiveness of prediction results with the use of sparse particle filtering (PF) model. The actual PCs are the solid blue curves and the predicted PCs are the dashed red curves. As shown in Figure 4.16, sparse PF model effectively captured nonlinear and nonstationary properties in the PCs that are extracted from space-time data of distributed sensor network. Due to the limitation in space, the remaining eight plots for PCs 3 to 10 are not shown. However, there is a close consistency between the remaining 8 PCs and the predicted results. Figure 4.17 shows the comparison of MAE for the prediction of 10 PCs using particle filtering

(PF), Kalman filtering (KF) and autoregressive-moving-average model [ARMA(3,3)]. Experimental results of MAE between actual and predicted PCs show the superiority of particle filtering method over other prediction methods. Note that the first 5 PCs have smaller MAE values then the last 5 PCs, because the first 5 PCs show predominant patterns (also see Figure 4.16) and the last 5 PCs have more random variations. In nonlinear PCA, the majority of variations in the data are captured by a low-dimensional set of PCs. Although the performances of all three models degrade for the last 5 PCs, the PF results are shown to be better than KF and ARMA models.

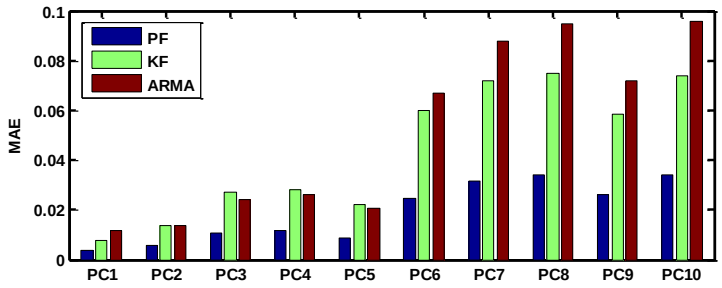


Figure 4.17 Performance comparison for the prediction of 10 PCs using PF, KF and ARMA.

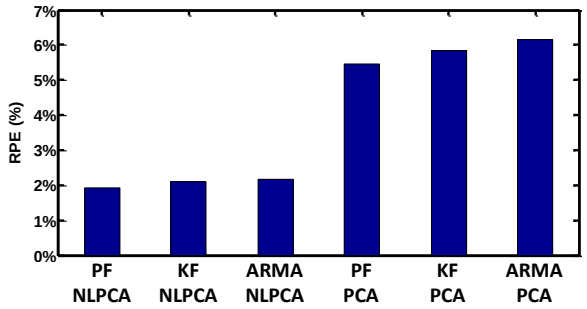


Figure 4.18 Comparison of model performances using various methods.

Further, we compared the model performances between linear PCA and nonlinear PCA, each of which transforms the high-dimensional set of 180 parameters to 10 PCs and then construct spatiotemporal models. Figure 4.18 shows experimental results for predicting the real-world data from the wearable ECG sensor network. Evidently, nonlinear PCA yields much better results than the linear PCA. The relative prediction errors of linear PCA are around 5.5% for the whole space-time datasets, while nonlinear PCA is about 1.9%. This significant improvement is because

spatiotemporal data from ECG sensor network are highly nonlinear and nonstationary. In addition, particle filtering also achieves smaller prediction errors than the other two methods, i.e., KF and ARMA(3,3). Notably, the sparse particle filtering model achieves the RPE of 1.9% with the low-dimensional set of 10 PCs as the hidden state variables.

4.5.3 Stochastic Sensor Network

Note that stochastic sensor networks employ a variable degree of contacts between skin and sensors and thereby generate missing data to some extent. As mentioned in Section 4.4, we simulated three scenarios of stochastic sensor network (i.e., spatially, temporally, and spatiotemporally dynamic networks). In this section, we compared prediction performances of sparse particle filtering in the presence of missing data in three simulation scenarios.

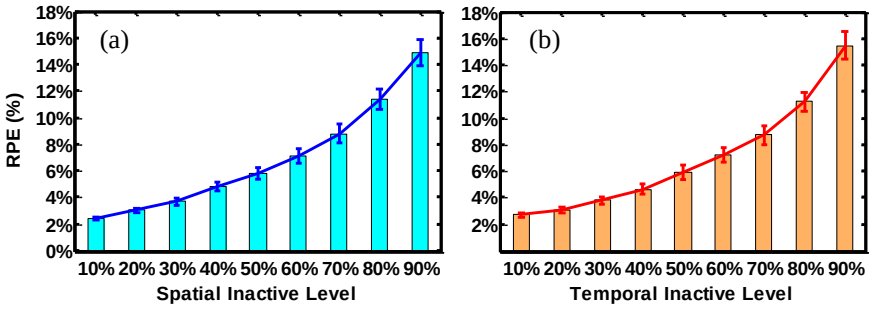


Figure 4.19 Model performances in (a) spatially and (b) temporally dynamic networks.

Figure 4.19 (a) shows the variations of RPE with respect to the inactivation of a certain level of electrodes out of 120 for the whole recording period. In this present investigation, we replicated the experiments for 100 times for each level. When the level of inactivation increases from 10% to 90%, the RPE monotonically increases from 2.4% to 14.9%. Notably, Figure 4.18 shows that the model RPE is 1.9% for 120-sensor ECG network without missing data. The RPE is only increased to 2.4% for the level of 10% disconnected electrodes, and 5.8% for 50% disconnected electrodes. These results show the effectiveness of sparse particle filtering to support the stochastic design and harness the uncertain information.

Similarly, Figure 4.19(b) shows the variations of RPEs with respect to 9 different levels of temporally dynamic network. For each level, every sensor is inactivated or disconnected from the ECG network for a corresponding percentage of time that is randomly chosen within the whole recording period. Here, we also replicated the experiments for 100 times for each level. When the level of temporal inactivation increases from 10% to 90%, the model RPE monotonically increases from 2.7% to 15.4%. Notably, the RPEs of sparse particle filtering model for temporal dynamic network are slightly higher (0.3%~0.5%) than spatially dynamic network, albeit they share similar magnitude and trend.

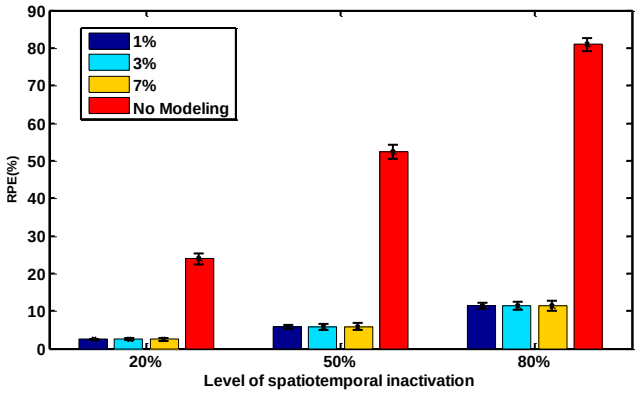


Figure 4.20 Model performances in the spatiotemporally dynamic network.

Furthermore, Figure 4.20 shows the distributions of RPEs for 9 different levels of spatiotemporally dynamic network (20%±1%, 20%±3%, 20%±7%, 50%±1%, 50%±3%, 50%±7%, 80%±1%, 80%±3%, and 80%±7%, also see Figure 4.12), which are simulated via stochastic Kronecker graph. For each level, we also replicated the experiments for 100 times. For the level of 20% spatiotemporal inactivation, the median of RPE distribution is about 3% for all three deviations (i.e., ±1%, ±3% and ±7%), but the RPE is around 23.9% if we do not use the algorithm of sparse particle filtering. It may also be noted that the median of RPE distribution is about 5.8% for the level of 50% spatiotemporal inactivation, but 52.3% without modeling efforts. The median is close to 11.4% for 80% spatiotemporal inactivation, but 81% without modeling efforts.

Figure 4.19 and Figure 4.20 show that the effectiveness of sparse particle filtering model in handling uncertainty data from three scenarios of stochastic sensor network (i.e., spatially, temporally, and spatiotemporally dynamic networks). When the level of inactivation is less than 50%, the median of model RPE is smaller than 6% for all three scenarios. Notably, sparse particle filtering can capture the characteristics of time-varying ECG data when half sensors are inactivated in the network, albeit at the expense of 6% prediction errors. Hence, this present investigation shows the feasibility to realize a highly resilient sensing system with the proposed novel technology of stochastic sensor network.

Furthermore, we evaluated and validated the algorithms of sparse particle filtering on a range of heart rates including healthy controls with 73 beats per minutes (bpm), tachycardia (110 bpm), and bradycardia (49 bpm). It should be noted that sparse particle filtering is a sequential Bayesian method with the unique feature of recursive estimation and prediction. Experimental results showed that different heart rates have trivial impacts on the performance of proposed algorithms.

4.6 Discussion and Conclusions

This paper presents a novel approach of sparse particle filtering for the characterization and modeling of space-time dynamics in big data emerged from stochastic sensor networks. First, we developed the sparse kernel-weighted spatial model based on cross-correlations among spatial locations. As opposed to traditional isotropic placement of kernels in the space, we developed a lazy greedy algorithm to optimally place a parsimonious set of kernels, thereby achieving a compact representation of spatial patterns. Further, the parameters of spatial model are time-varying, which are sequentially updated when new observations are available at the next time point. As such, spatial and temporal processes closely interact with each other. To address data

nonlinearity and nonstationarity, we propose the particle filter (PF) to recursively compute the estimation and prediction steps, and approximate the posterior distribution of states with a large number of particles generated by sequential Monte Carlo sampling.

However, the dimensionality of parameter vector is large. This poses a significant challenge of "curse of dimensionality" for estimation and prediction steps in particle filtering. The high-dimensional states will easily cause both overfitting and ill-posed estimation problems. Very little work has been done to develop a sparse particle filtering model that identifies a compact set of independent states that suffice to represent the nonlinear dynamic process. In this study, we integrated the nonlinear PCA with particle filtering for effectively modeling space-time dynamics in stochastic sensor networks.

Experimental results with real-world data from the body area sensor network show that the proposed methodology not only resolves the problem of "curse of dimensionality" but also effectively handles data uncertainty in the stochastic sensor network. The proposed sparse particle filtering model with a low-dimensional set of 10 PCs as the hidden state variables yields the RPE of 1.9% for 120-sensor ECG network. In addition, we simulated three scenarios of stochastic sensor network (i.e., spatially, temporally, and spatiotemporally dynamic networks). Experimental results demonstrated the effectiveness of sparse particle filtering to support the design of stochastic sensor networks and exploit spatiotemporal data for information extraction and knowledge discovery. When the level of inactivation is less than 50%, the median of model RPE is smaller than 6% for all three scenarios. This present investigation demonstrates the feasibility to realize a highly resilient sensing system with the new idea of stochastic sensor network. The proposed algorithms are very general and can be potentially applicable for stochastic sensor networks in a variety of disciplines, e.g., environmental sensor network and battlefield surveillance network.

Our future research will continue investigating the application of stochastic sensor networks for wearable cardiac monitoring, informatics and diagnostics. For example, optimal kernels are adaptively estimated based on the patterns of ECG imaging, which have great potentials in identifying the location and extent of infarcts. In addition, we will conduct further experiments to evaluate the performance of stochastic ECG sensor networks when the subjects are stationary or moving.

CHAPTER 5: HETEROGENEOUS RECURRENCE QUANTIFICATION OF DYNAMIC TRANSITIONS IN CONTINUOUS NONLINEAR PROCESSES[4]

Many real-world systems are evolving over time and exhibit dynamical behaviors. In order to cope with system complexity, sensing technology is commonly deployed to monitor system dynamics. Online sensing brings the proliferation of big data that are nonlinear and nonstationary. Although there is rich information on nonlinear dynamics, significant challenges remain in realizing the full potential of sensing data for system control. This paper presents a new approach of heterogeneous recurrence analysis for online monitoring and anomaly detection in nonlinear dynamic processes. A partition scheme, named as Q-tree indexing, is firstly introduced to delineate local recurrence regions in the multi-dimensional continuous state space. Further, we design a new fractal representation of state transitions among recurrence regions, and then develop new measures to quantify heterogeneous recurrence patterns. Finally, we develop a multivariate detection method for on-line monitoring and predictive control of process recurrences. Case studies show that the proposed approach not only captures heterogeneous recurrence patterns in the transformed space, but also provides effective online control charts to monitor and detect dynamical transitions in the underlying nonlinear processes.

5.1 Introduction

Many real-world systems are evolving over time and exhibit dynamical behaviors. As complex systems evolve in time, dynamics deal with change. Whether the system settles down to

[4] Part of this chapter is previously published in [152]. Permission is included in Appendix A.

a steady state, undergoes incipient changes, or deviates into more complicated variations, we can leverage dynamics to help analyze system behaviors. Therefore, modern industry is investing in a variety of sensor networks and data acquisition devices to monitor system dynamics. It is expected that such sensing technologies will increase information visibility and cope with system complexity. However, online sensing brings the proliferation of big data that are nonlinear and nonstationary. Realizing the full potential of big data depends on the information-processing capabilities to harness and exploit nonlinear dynamics in the underlying process.

Indeed, nonlinear dynamics pose significant challenges for effective process monitoring and control. It is well known that nonlinear dynamics defy understanding based on the traditional reductionist's approach, in which one attempts to understand a system's behavior by combining all constituent parts that have been analyzed separately. Traditional statistical process control (SPC) is not concerned with waveform signals but key process features or characteristics, and is limited in its ability to readily address nonlinear dynamics in complex systems. In addition, linear methods and tools are widely adopted in data analysis to interpret regular structures (e.g., dominant frequencies), but have encountered certain difficulties to capture nonlinear variations [3]. Dealing with nonlinear dynamics is a general problem facing both the traditional and next-generation innovation practices in process monitoring and control. There is an urgent need to develop new monitoring and control schemes that can extract useful information about nonlinear dynamics and exploit the acquired information for process control.

This paper presents a new approach of heterogeneous recurrence analysis for online monitoring and anomaly detection in nonlinear dynamic processes. First, we reconstruct the state space using sensing data collected from dynamical systems, and then recursively partition this state space into a hierarchical structure of local recurrence regions. Second, we develop a new method

of fractal representation to characterize heterogeneous recurrence behaviors in the state space and further extract statistical quantifiers of recurrence patterns. Third, we integrate multivariate SPC methods with heterogeneous recurrence analysis to simultaneously monitor dynamic transitions in continuous nonlinear processes.

Further, we evaluate and validate the developed methodology of heterogeneous recurrence analysis with simulation datasets, as well as two real-world case studies in nanoscale machining and cardiac operations. Although they are disparate in disciplines, both systems exhibit recurrence dynamics and involve greater levels of complexity. Real-time sensing (e.g., cutting force, vibration, and acoustic emission) is required mostly to monitor nano-machining dynamics and process-machine interactions for higher yields and better repeatability [102]. In addition, electrocardiogram (ECG) signals are recorded on the body surface to track the continuous dynamic details of cardiac functioning [43, 81]. Effective monitoring and control of nonlinear dynamics will increase system quality and integrity, thereby leading to significant economic and societal impacts. Experimental results demonstrate that the proposed approach not only captures heterogeneous recurrence patterns in the transformed space, but also provides effective online control charts to monitor and detect dynamical transitions in the underlying nonlinear process.

The rest of this paper is organized as follows: Section 5.2 introduces the state of the art in process monitoring of nonlinear dynamics. Section 5.3 presents the research methodology. Section 5.4 and 5.5 show the experimental design and results, and Section 5.6 includes the discussion and conclusions arising from this investigation.

5.2 Research Background

Real-time sensing brings large amounts of sensor signals from complex systems. Traditional linear methods and tools focus on the analysis of time-domain signals, and attempt to

understand a system's behavior by breaking it down into parts and then combining all constituent parts that have been examined separately. This idea underlies such methods as principal component analysis (PCA), Fourier analysis, and factor analysis. These methods encounter difficulties in capturing nonlinear and nonstationary dynamics. The breakthrough in nonlinear theory came with Poincaré's geometric thinking of dynamical systems, which focuses on the analysis of geometric properties in the state space.

Poincaré recurrence theorem [4] shows that if a dynamical system has a measure invariant under a transformation, its trajectories eventually reappear in the ε-neighborhood of former states, which are commonly referred to be "recurrence dynamics". Process monitoring of dynamic transitions in complex systems (e.g., disease conditions or machine faults) is more concerned with aperiodic recurrences and recurrence variations. However, very little has been done to investigate different types of recurrences and heterogeneous recurrence variations for process monitoring and control.

Traditional SPC is not concerned with nonlinear dynamics in sensor signals but key process characteristics [103-105]. Most previous work mainly focused on the analysis of time-domain signals to extract process features or characteristics. Sensing capabilities are not fully utilized to investigate nonlinear dynamics such as the variation of recurrence patterns in the state-space domain. However, English et al. [106] and Ben-Gal et al. [107] showed that engineering control implementations often bring nonlinear dynamics in sensor signals. It is imperative to develop new process monitoring and control schemes that consider state-dependent and nonlinear dynamics of complex systems. As a result, there is an increasing attention on the analysis of nonlinear dynamics in sensor signals for process monitoring and control [108]. Ruschin-Rimini et al. [109] developed a fractal-SPC method that uses fractal dimensions to measure the probability of the occurrence of

correlated data sequences for process monitoring and change detection. Qiang et al. [110-112] considered nonlinear phase synchronization and physical interactions of process variables for conditional monitoring and diagnosis in a chemical-mechanical planarization process. In spite of these important advances, few, if any, previous approaches investigated heterogeneous recurrence variations for on-line process monitoring and control.

Recurrence is one of the most common phenomena in natural and engineering systems. The recurrence plot is a graphical tool to characterize the proximity of two states $\vec{\boldsymbol{x}}(i)$ and $\vec{\boldsymbol{x}}(j)$ in the state space, i.e., $R(i,j) \coloneqq \Theta(\varepsilon - \|\vec{\boldsymbol{x}}(i) - \vec{\boldsymbol{x}}(j)\|)$, where Θ is the Heaviside function and $||\cdot||$ is a distance measure [15]. For example, Figure 5.1 shows the recurrence plot of an ECG state space. If two states are located close to each other in the d-dimensional state space (e.g., 3D space in Figure 5.1), the color code is black. If they are located farther apart, the color is white. The structure of a recurrence plot has distinct topology and texture patterns. These patterns are closely pertinent to the properties of system dynamics. For example, ridges locate nonstationarity and/or switching between local behaviors. Parallel diagonal lines indicate the near-periodicity of system behaviors.

Recurrence quantification analysis measures intriguing structures and patterns in the recurrence plot, including small-structures (e.g., small dots, vertical and diagonal lines), chaos-order transitions, as well as chaos-chaos transitions (for a comprehensive review, see [15]). Examples of recurrence quantifiers include: (1) determinism (DET) - the percentage of recurrence points which form diagonal lines, $\mathrm{DET} = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=1}^{N} lP(l)}$, where $P(l)$ is the histogram of diagonal line length; (2) laminarity (LAM) - the percentage of recurrence points which form vertical lines, $LAM = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=1}^{N} vP(v)}$, where $P(v)$ is the histogram of vertical line lengths. Recurrence

quantification analysis goes beyond the graphical representation of recurrence plots and provides complexity measures of nonlinear dynamics. In addition, our prior work developed new recurrence methods (i.e., local recurrence model and multiscale recurrence analysis) for the characterization, representation and prediction of nonlinear dynamics in manufacturing assembly lines [22, 113] and cardiovascular systems [3, 43, 80].

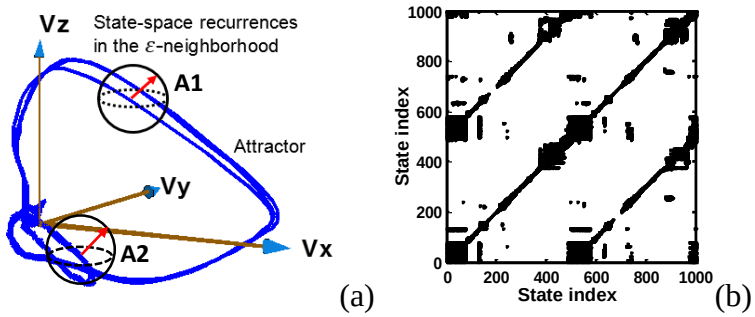


Figure 5.1 An example of the ECG state space (a) and its recurrence plot (b).

However, traditional recurrence plots treat recurrence states homogeneously using the Heaviside function. In other words, if there is a spatial partition of the state space, recurrences in one region are treated the same as in other regions. Recurrences can be different in kind because of state properties (e.g., state values and relative locations in the state space) and the evolving system dynamics (e.g., sequential state transitions before and after). For example, there are two recurrence regions A1 and A2 in Figure 5.1. Traditional recurrence methods treat both regions homogeneously as black dots in the recurrent plot (see Figure 5.1). However, recurrence states in region A1 are pertinent to ventricular activity of the heart, which has different physical meanings from recurrence states in region A1 (i.e., atrial activity of the heart). Very little work has been done to delineate heterogeneous recurrences (i.e., different kinds of recurrence behaviors in local regions of state space). Nonetheless, process monitoring and control are more concerned with heterogeneous recurrences and their variations hidden in nonlinear and nonstationary sensor signals. We have developed a heterogeneous recurrence model for stochastic Markov processes

with a discrete and finite set of states [114]. However, this preliminary work did not specifically consider the continuous state space of nonlinear dynamic processes.

5.3 Heterogeneous Recurrence Analysis

The present paper proposes an effective partition scheme to delineate local recurrence regions in the multi-dimensional continuous state space. Further, we design a new fractal representation of state transitions among recurrence regions, and then develop new measures to quantify heterogeneous recurrence patterns. Finally, we develop a multivariate detection method for on-line monitoring and predictive control of process recurrences.

5.3.1 State Space Segmentation

In the theory of nonlinear dynamics, the state evolution of underlying processes is modeled by a set of nonlinear differential equations, i.e., $\dot{\boldsymbol{X}} = \frac{d\boldsymbol{X}}{dt} = F(\boldsymbol{X}, \boldsymbol{\theta}), F \in \mathbb{R}^n \to \mathbb{R}^n$, where $\boldsymbol{X}$ is a multi-dimensional state variable, F is the nonlinear function, and $\boldsymbol{\theta}$ contains the model parameters. Thus, the solution, i.e., $\boldsymbol{X} = f(\boldsymbol{X}(0), t)$, generates a trajectory representing the state evolution for a given initial condition $\boldsymbol{X}(0)$. When there is a small perturbation in $\boldsymbol{\theta}$ or $\boldsymbol{X}(0)$, the dynamics of a nonlinear process can undergo abrupt changes and reveal complex characteristics such as chaos and recurrences. Although sensor observations from complex systems are often chaotic in nature, dynamics manifest in the vicinity of an attractor (e.g., ECG attractor shown in Figure 5.1), an invariant set defined in a d-dimensional state space. For the time series $\boldsymbol{X} = \{x_1, x_2, \cdots, x_N\}^T$, a state vector $\vec{\boldsymbol{x}}$ is reconstructed using a delay sequence of $\{x_i\}$ as $\vec{\boldsymbol{x}}(i) = \left[x_i, x_{i+\tau}, \cdots, x_{i+\tau(d-1)}\right]$, where d is the embedding dimension and τ is the time delay. Here, the embedding dimension d is determined by the false nearest neighbor algorithm [12], and the time delay parameter τ is estimated with the mutual information method [13].

In order to delineate local recurrence regions, it is necessary to segment or decompose the state space. Such a segmentation facilitates the clustering of states and the identification of heterogeneous recurrence patterns. K-Means clustering is perhaps one of the most popular algorithms for data clustering. However, features (or variables) in traditional clustering problems are different from the evolving states in the state space. In addition, K-Means clustering has several drawbacks for online process monitoring and control. First, dissimilarity computation in each iteration is based on all the data. As such, computation is highly expensive for the time-evolving state space. Second, the K-Means algorithm needs all previous states to compute dissimilarity measures, as well as recalculate the objective function for assigning a cluster label to a new state. Third, the results of K-Means clustering are also time-varying (i.e., not unique) due to the recalculation of the objective function.

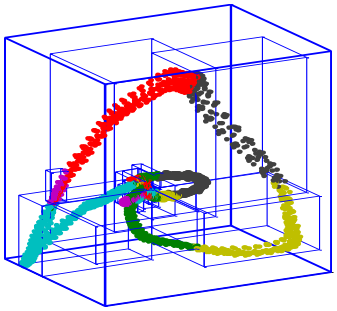


Figure 5.2 An illustration of Q-tree segmentation of state space.

Therefore, we propose a multidimensional indexing method (i.e., Q-tree indexing) to derive the hierarchical structure of state space. It is worth mentioning that Q-tree indexing is an effective and efficient methodology for multidimensional data indexing in large-scale database management [115]. In the present investigation, we bring this new idea to develop Q-tree indexing of the state space for delineating heterogeneous recurrence regions. The Q-tree recursively divides the d-dimensional state space into 2^d sub-regions. Each sub-region is the d-dimensional hyperrectangle that has the same size as other sub-regions. If a sub-region is further decomposed, the Q-tree

generates 2^d new child sub-regions and then distributes states into these child sub-regions. Figure 5.2 shows an example of the hierarchical partition of a three-dimensional state space. This hierarchical partition will continue until the leaf sub-region reaches the capacity, where the capacity is the maximal number of states in a sub-region. In other words, the partition stops when the number of states in a sub-region is smaller than the capacity. The Q-tree indexing of state space provides a hierarchical structure of states and delineates salient patterns pertinent to heterogeneous recurrences in the state space.

5.3.2 Heterogeneous Recurrence Representation

The Q-tree partitions the state space into sub-regions, each of which contains a number of states less than the capacity. If we assign a categorical variable to each nonempty sub-region, we obtain a time series of this categorical variable when the system evolves among sub-regions. Here, we introduced the iterative function system (IFS) to characterize state transitions and represent heterogeneous recurrences in the time series of this categorical variable. For a state space segmented with a finite set of sub-regions, a unique value of categorical variable k is assigned to each sub-region $\vec{\boldsymbol{s}}(n)$, where k belongs to a finite set $\mathcal{K}$ of positive integers. The IFS sequentially maps each sub-region $\vec{\boldsymbol{s}}(n)$ to an address $\left[c_x(n), c_y(n)\right]$ in the 2D coordinate system as:

$$\vec{\boldsymbol{s}}(n) \to k \in \mathcal{K} = \{1,2,\cdots,K\}$$

$$\begin{bmatrix} c_x(n) \\ c_y(n) \end{bmatrix} = \varphi\left(k, \begin{bmatrix} c_x(n-1) \\ c_y(n-1) \end{bmatrix}\right) = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} c_x(n-1) \\ c_y(n-1) \end{bmatrix} + \begin{bmatrix} cos\left(k \times \frac{2\pi}{K}\right) \\ sin\left(k \times \frac{2\pi}{K}\right) \end{bmatrix} \quad (1)$$

where $\begin{bmatrix} c_x(0) \\ c_y(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, α is a control parameter that prevents overlaps of two sub-regions in the graph. If the inequality $\frac{\alpha}{1-\alpha} < sin\left(\frac{\pi}{K}\right)$ is satisfied, K sub-regions will not overlap in the IFS mapping graph. It may be noted that the IFS of a circle transformation (Eq. 1) is widely used to

characterize and model fractal properties of a time series [116, 117]. The iterative contractive mapping $\varphi(k, \boldsymbol{c})$ represents the dynamics of state transitions among regions as vectors in $\Re^2$. Figure 5.3 shows the IFS representation of the time series of categorical variable k with 8 sub-regions via the IFS. Each circle in Figure 5.3b shows the recurrences of one out of 8 possible categories in the time series (see Figure 5.3a). These categories are centered around 8 addresses that are uniformly distributed on the unit circle. As such, heterogeneous recurrences are effectively separated in the 2D graph at the level of 8 individual sub-regions. Zooming into address 1 (i.e., marked by the blue rectangle in Figure 5.3b) leads to Figure 5.3c. Every circle in Figure 5.3c represents one of eight transition sequences from all sub-regions to sub-region 1, i.e., $11,21, \dots ,81$ in the time series. Notably, there are four two-region sequences (i.e., 11, 21, 51 and 71) missing in Figure 5.3c, i.e., there is a zero transition probability between these two regions. The density of points in each circle is pertinent to the frequency of a two-region sequence. As such, the density and distribution of points in each circle characterize heterogeneous recurrence variations of the dynamics of state transitions among sub-regions in the state space.

Further, we developed three new quantifiers in our previous work [114] to describe heterogeneous recurrence patterns in the IFS representation (see Figure 5.3b and 3c). As the IFS clusters the states within the same region in the 2D graph, we denote these clusters as heterogeneous recurrence sets, $W_{k_1,k_2,\dots,k_L} = \{\varphi(k_1|k_2, \dots, k_L): \vec{\boldsymbol{s}}(n) \to k_1, \dots, \vec{\boldsymbol{s}}(n-L+1) \to k_L\}$ and $k_1, k_2 \dots, k_L \in \mathcal{K}$. In other words, W_{k_1} denotes the recurrence set of an individual region k_1, W_{k_1,k_2} denotes the recurrence set of 2-region sequence $[\vec{\boldsymbol{s}}(n) \to k_1, \vec{\boldsymbol{s}}(n-1) \to k_2]$, and $W_{k_1,k_2,\dots,k_L}$ denotes the recurrence set of L-region sequence $[\vec{\boldsymbol{s}}(n) \to k_1, \vec{\boldsymbol{s}}(n-1) \to k_2, \dots, \vec{\boldsymbol{s}}(n-L+1) \to k_L]$. As a result, we develop 3 new quantifiers of heterogeneous recurrences, namely heterogeneous recurrence rate (HRR), heterogeneous mean (HMean) and heterogeneous entropy

(HENT), that are pertinent to the dynamics of a complex system. Where HRR measures the percentage of recurrences, HMean provides general information about average distance among elements and HENT shows the uncertainty in the heterogeneous recurrence plot, see more details in our previous work [114].

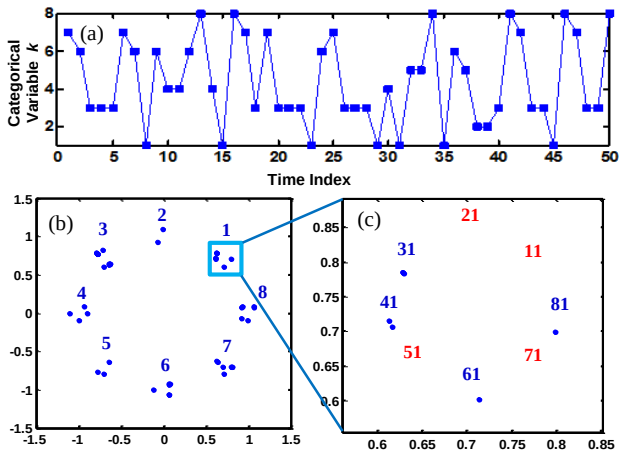


Figure 5.3 (a) Time series of categorical variable k when states evolve among regions, where $k \in \mathcal{K} = \{1,2,\cdots,8\}$. (b) IFS addresses of individual regions. (c) IFS addresses of two-region transitions.

5.3.3 Heterogeneous Recurrence Monitoring

Heterogeneous recurrence quantification leads to multiple features (or quantifiers), which can be utilized for monitoring system dynamics. The aim of a statistical hypothesis test for the presence of a dynamic transition is to determine whether there is a significant change in the feature vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_p]^T$, where $\boldsymbol{y}$ denotes the p-dimensional vector of heterogeneous recurrence features (i.e., HRR, HMean and HENT). Suppose that $\boldsymbol{y}$ is a multivariate random variable with population mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Under the null hypothesis H_0, process dynamics does not change over time. Hence, we assume that $\boldsymbol{y}$ follows a multivariate normal distribution. If a dynamic transition occurs, the joint distribution of heterogeneous recurrence features will change markedly. The hypothesis test is to identify data-driven evidences that reject the null hypothesis H_0 at a specific level of significance [114]. However, population mean $\boldsymbol{\mu}$ and

covariance matrix $\boldsymbol{\Sigma}$ need to be estimated from the data. If we replace $\boldsymbol{\mu}$ by the sample mean $\bar{\boldsymbol{y}}$ and $\boldsymbol{\Sigma}$ by the sample covariance matrix $\boldsymbol{S}$, the test statistic becomes $T^2 = (\boldsymbol{y} - \bar{\boldsymbol{y}})'\boldsymbol{S}^{-1}(\boldsymbol{y} - \bar{\boldsymbol{y}})$, named Hotelling T^2 statistic. The confidence interval of the Hotelling T^2 statistic is: $UCL = \frac{p(M+1)(M-1)}{M^2-Mp}F_{\alpha,p,M-p}$, where M is the number of samples, and $F_{\alpha,p,M-p}$ is the upper $100\alpha\%$ percentile of F distribution with p and $M-p$ degrees of freedom [114].

However, a significant challenge resides in the inversion of the sample covariance matrix $\boldsymbol{S}$. Because the transition probability between two regions may be zero, this leads to a singular covariance matrix, thereby making the computation of the Hotelling T^2 statistic impractical. Hence, we transform the feature matrix $\boldsymbol{Y}_{M\times p}$ into a set of principal components (PCs) that are linearly uncorrelated. First, the feature matrix $\boldsymbol{Y}_{M\times p} = [\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_M}]^T$ is centered by subtracting the column means, i.e., $\boldsymbol{Y}^* = [\boldsymbol{y_1} - \bar{\boldsymbol{y}}, \boldsymbol{y_2} - \bar{\boldsymbol{y}}, \dots, \boldsymbol{y_M} - \bar{\boldsymbol{y}}]^T$. Then, the singular value decomposition (SVD) of $\boldsymbol{Y}^*$ will be

$$\boldsymbol{Y}^* = \boldsymbol{U\Psi V^T} \tag{2}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are $M \times M$ and $p \times p$ orthogonal matrices, $\boldsymbol{\Psi}$ is a $M \times p$ diagonal matrix, with diagonal entries $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$(i.e., the singular values of $\boldsymbol{Y}^*$). Hence, the covariance matrix $\boldsymbol{Y}^{*\boldsymbol{T}}\boldsymbol{Y}^*$ is

$$\boldsymbol{Y}^{*\boldsymbol{T}}\boldsymbol{Y}^* = \boldsymbol{V\Psi U^T U\Psi V^T} = \boldsymbol{V\Psi^2 V^T} \tag{3}$$

which is the eigen decomposition of $\boldsymbol{Y}^{*\boldsymbol{T}}\boldsymbol{Y}^*$. The eigenvectors $\boldsymbol{v}_i$ are the principal component directions of $\boldsymbol{Y}^*$. As a result, principal components are given as

$$\boldsymbol{Z} = \boldsymbol{Y}^*\boldsymbol{V} = \boldsymbol{U\Psi V^T V} = \boldsymbol{U\Psi} \tag{4}$$

The first principal component $\boldsymbol{Z}(:,1) = \mathbf{Y}^*\boldsymbol{v}_1$ has the largest sample variance among all the principal components. The sample covariance matrix $\boldsymbol{S}$ is easily seen to be:

$$\boldsymbol{S} = \frac{1}{M-1}\sum_{i=1}^{M}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y_i} - \bar{\boldsymbol{y}})^T = \boldsymbol{V}\boldsymbol{S_z}\boldsymbol{V^T} \tag{5}$$

Notably, principal components $\boldsymbol{Z}(:,i)$ and $\boldsymbol{Z}(:,j)$ are mutually orthogonal. Hence, the covariance matrix of principal components $\boldsymbol{S_z}$ is a diagonal matrix with diagonal entries ${\lambda_1}^2 \geq {\lambda_2}^2 \geq \cdots \geq {\lambda_p}^2$. The Hotelling T^2 statistic becomes

$$\begin{aligned} T^2(i) &= (\boldsymbol{y_i} - \bar{\boldsymbol{y}})^T\boldsymbol{S}^{-1}(\boldsymbol{y_i} - \bar{\boldsymbol{y}}) = \boldsymbol{Z}(i,:)\boldsymbol{V^T}(\boldsymbol{V}\boldsymbol{S_z}\boldsymbol{V^T})^{-1}\boldsymbol{V}\boldsymbol{Z}(i,:)^{\boldsymbol{T}} \\ &= \boldsymbol{Z}(i,:)\boldsymbol{S_z^{-1}}\boldsymbol{Z}(i,:)^{\boldsymbol{T}} = \sum_{k=1}^{p}\frac{Z(i,k)^2}{\lambda_k^2} \end{aligned} \tag{6}$$

where $\boldsymbol{Z}(i,:)$ is the projection of the ith sample on the principal component directions, that is, $(\boldsymbol{y_i} - \bar{\boldsymbol{y}})'\boldsymbol{V}$. In addition, a truncated M-by-q score matrix $\boldsymbol{Z}_q$ can be obtained by considering only the first q eigen values and eigenvectors ($q < p$), which explain the majority of variations in the feature matrix $\boldsymbol{Y}_{M\times p}$. Hence, the Hotelling T^2 statistic in the reduced dimension q is: $\tilde{T}^2(i) = \sum_{k=1}^{q}\frac{Z(i,k)^2}{\lambda_k^2}$. The Hotelling T^2 statistic can be plotted for each sample on a control chart together with its confidence interval, which characterizes the differences between multivariate recurrence features of different samples.

5.4 Experimental Design

Next, we evaluate and validate the proposed methodology with both simulation and real-world case studies. Materials and experimental details are described in the following two subsections.

5.4.1 Design of Simulation Experiments

Two models are used in our simulation studies to generate time series: (1) an auto-regressive model of order 2 - AR(2), $x_i = \mathrm{a}x_{i-1} + bx_{i-2} + c\varepsilon_i$, where i is the time index, a, b, c are model parameters, and $\varepsilon_i \sim N(0,1)$; (2) the Lorenz system, i.e., $\dot{x} = \sigma(y - x)$, $\dot{y} =$

$x(\rho - z) - y,\ \dot{z} = xy - \beta z$, where (x, y, z) is the state vector the Lorenz system, and σ, ρ, β are parameters. Here, we assume that dynamic transitions are caused by the drift of model parameters. As shown in Figure 5.4, the considered parameters of AR(2) process are $a = 1, b = -1, c = 0.5$ and the parameters of the Lorenz system are $\sigma = 10$, $\rho = 20$, $\beta = 1$. We will then vary each parameter to generate the dynamic transitions for testing the performance of heterogeneous recurrence monitoring.

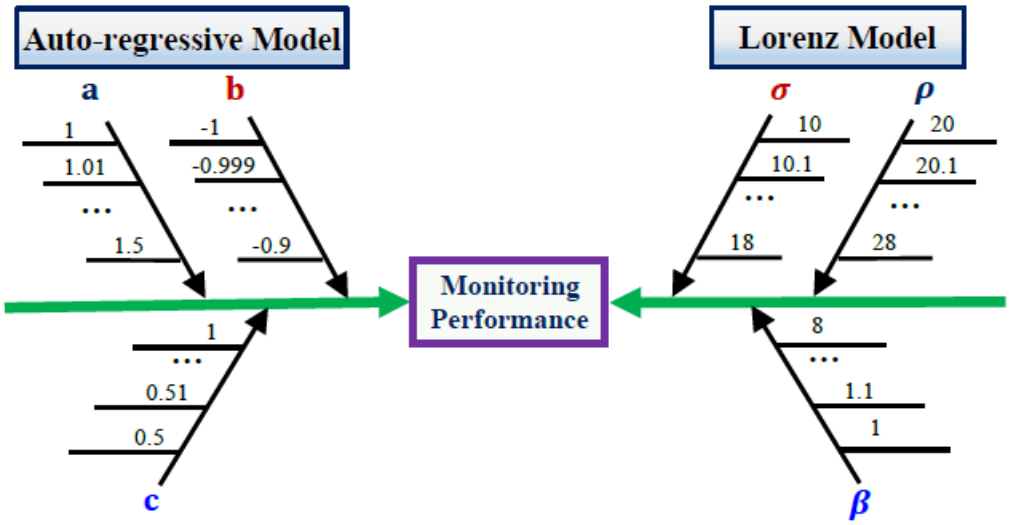


Figure 5.4 Experimental design of the simulation studies.

Each process is represented by a time series that is a specific realization of the process. For each parametric scenario, we generate samples of 1000 independent process realizations. Further, the Hotelling T^2 statistic is derived using multivariate recurrence features that are extracted from the time series of each realization. To test the monitoring performance, we compute the detection power that is the proportion of dynamic transitions correctly detected. In other words, the detection power describes how many dynamic transitions are correctly detected from 1000 simulated cases. In order to benchmark the performance, we compare the proposed heterogeneous recurrence T^2 statistic with two traditional recurrence quantifiers: DET and LAM.

5.4.2 Real-world Case Studies

In addition to the artificial data examples, we evaluate the proposed methodology of heterogeneous recurrence statistics for real-time monitoring of the characteristics of surface finishes in the ultra-precision machining (UPM) process. In our UPM experiments, a polycrystal diamond cutting tool ($60\mu m$ nose radius) was used for the machining of 36 workpieces ($\Phi 8cm \times 3cm$ cylindrical aluminum) [118]. Equipped with air-bearing spindles (model Block-Head® 4R) and diamond tools, UPM is a direct method to generate optical surfaces for applications such as telecommunication, defense, and biomedical product components [102]. In the industry practice, real-world uncertainty factors such as microstructure heterogeneity, environmental changes and process drifts often cause the variations of surface finishes in UPM machining. We mounted an acoustic emission (AE) sensor (R80) from Physical Acoustics on the top of tool holder in the UPM machine for in-process monitoring and quality control. The sampling rate of the acoustic emission signals is $50kHz$.

Our research objective is to extract recurrence features from sensor signals, and then develop real-time monitoring schemes for characterizing the quality of surface finishes. We collected acoustic emission (AE) signals from a total of 127 samples in UPM experiments. The surface characteristics were measured offline with a confocal optical microscope MicroXAM®. There are 116 samples with surface roughness (R_a) < 100 nm, which are used as the normal group. In turn, there is another group of 11 samples with R_a>100 nm. If the surface finish (Ra) is in the range of 1-100 nm, then the machined surface has good quality. This is also the reason to choose Ra=100 nm as a boundary to detect the poor quality of surface finishes [119]. Figure 5.5 shows two examples of 3D surface profiles that are acquired using the confocal optical microscope MicroXAM®. We again compare the heterogeneous recurrence method with conventional RQA

measures (DET and LAM) to demonstrate the performance of sensor-based approaches in characterizing and modeling the UPM dynamics and its pertinent surface characteristics.

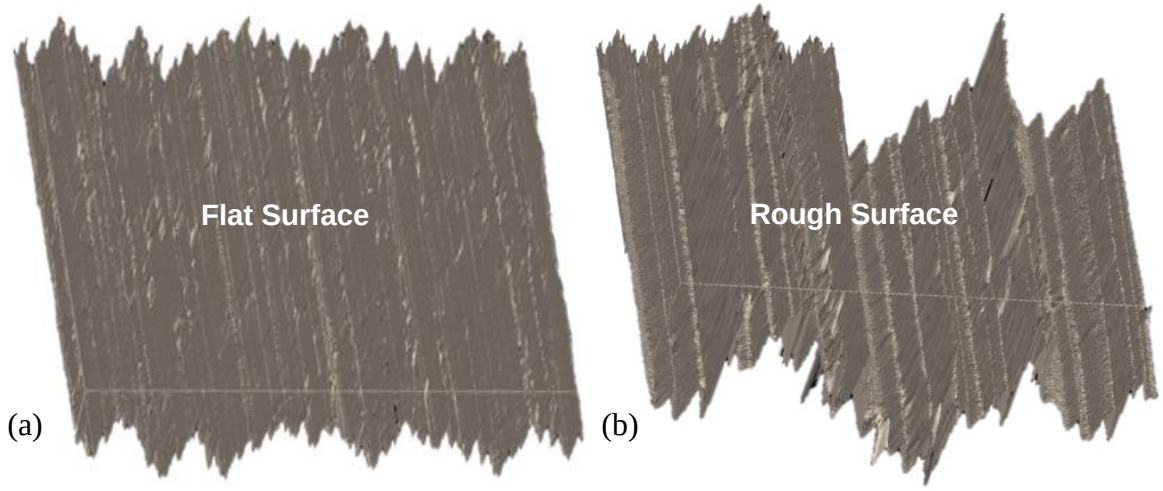


Figure 5.5 3D Profiles of workpieces. (a) Flat surface (Ra<100nm). (b) Rough surface (Ra>100nm).

Finally, we conduct a pilot study on heterogeneous recurrence monitoring of ECG signals for the prediction of abnormal heart beats. As there are many different types of heart diseases, this pilot study only focuses on a specific cardiac arrhythmia, premature ventricular contractions (PVCs). We study two ECG samples from the MIT-BIH Arrhythmia Database (https://www.physionet.org/physiobank/database/mitdb/) [40], record 101 with 5500 data points and record 116 with 5000 data points. The ECG recordings are digitized at 360 data points per second. Heterogeneous recurrence features are extracted from the reconstructed state space of ECG signals. We apply sliding windows of size $w = 500$ time steps and a step size of 50 time steps along the state space trajectory. Notably, this pilot study is designed to show the potentials of heterogeneous recurrence methods for cardiac monitoring, while a systematic investigation with more ECG samples will be further conducted to show the statistical validity and significance in our future work.

5.5 Experimental Results

5.5.1 Simulation Studies on AR(2) Model

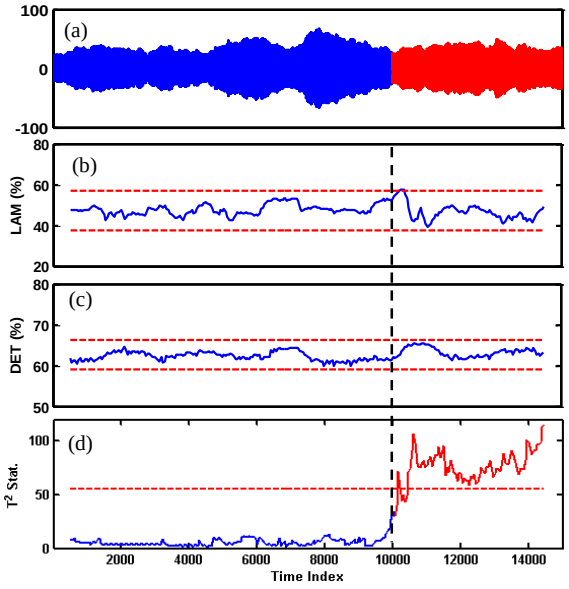


Figure 5.6 (a) AR(2) time series with varying parameters. The blue segment corresponds to the parameters $a = 1, b = -1, c = 0.5$, and the red one to $a = 1.05, b = -1, c = 0.5$. In addition, the evolution of (b) DET, (c) LAM and (d) heterogeneous recurrence T^2 statistics is shown using sliding windows as described in the text.

Figure 5.6a shows time series of the AR(2) model. The first 10000 data points (blue color) are simulated with parameters $a = 1, b = -1, c = 0.5$, and the next 5000 data points (red color) are simulated by slightly changing the parameters to $a = 1.05, b = -1, c = 0.5$. We use sliding windows of size $w = 1000$ and a step size of 50 to compute the recurrence T^2 statistics and the two RQA measures DET and LAM from the reconstructed state space of AR(2) time series. Here, we use the embedding dimension 3 and time delay 1. The state space is partitioned into 8 sub-regions for heterogeneous recurrence analysis. When computing RQA measures, the threshold ε is chosen to preserve a constant recurrence rate of 5%. As shown in Figure 5.6, the recurrence T^2 statistics yields a clearly better performance than the traditional RQA measures DET and LAM. The red dashed lines in Figure 5.6b-d represent confidence intervals of monitoring statistics, where

the confidence intervals of RQA measures are sample mean $\pm 3 \times$ sample standard deviation, and the upper confidence bound of T^2 statistics is computed as $UCL = \frac{p(M+1)(M-1)}{M^2-Mp} F_{\alpha,p,M-p}$ (see Section 5.3.3). See the definition of confidence intervals in [114]. The black dashed line represents the time index when model parameters are slighted change.

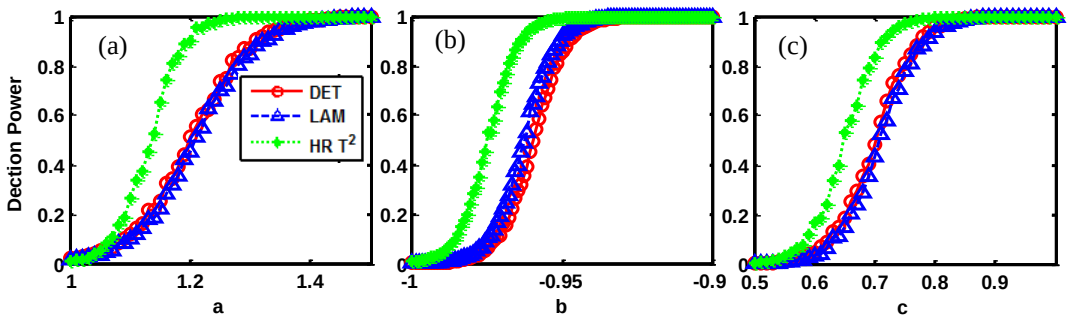


Figure 5.7 Performance comparison of detection power of DET, LAM and heterogeneous recurrence T^2 statistics for AR(2) models with varying parameter (a) a, (b) b and (c) c.

Figure 5.7 shows the performance comparison of detection power of DET, LAM and heterogeneous recurrence T^2 statistics for AR(2) models with varying parameters. The detection power is the proportion of dynamic transitions correctly detected out of 1000 simulated cases. Parameter variations are based on the experimental design in Figure 5.4. We used 1000 samples for both normal and dynamic transition cases, each of which has 500 data points in the time series. As shown in Figure 5.7, the detection power of the proposed recurrence T^2 statistics is much better than that of DET and LAM. Figure 5.7a shows the variations of parameter a from 1 to 1.5 in the AR(2) model with fixed parameters $b = -1, c = 0.5$. Notably, the recurrence T^2 statistics generates an alarm for the presence of a dynamic transition much earlier than the two RQA measures, which indicates an effective monitoring scheme for dynamical transitions in AR(2) model. As the parameter a gets bigger than 1.4, all three characteristics perform equally well in generating an out-of-control alarm.

5.5.2 Simulation Studies on the Lorenz Model

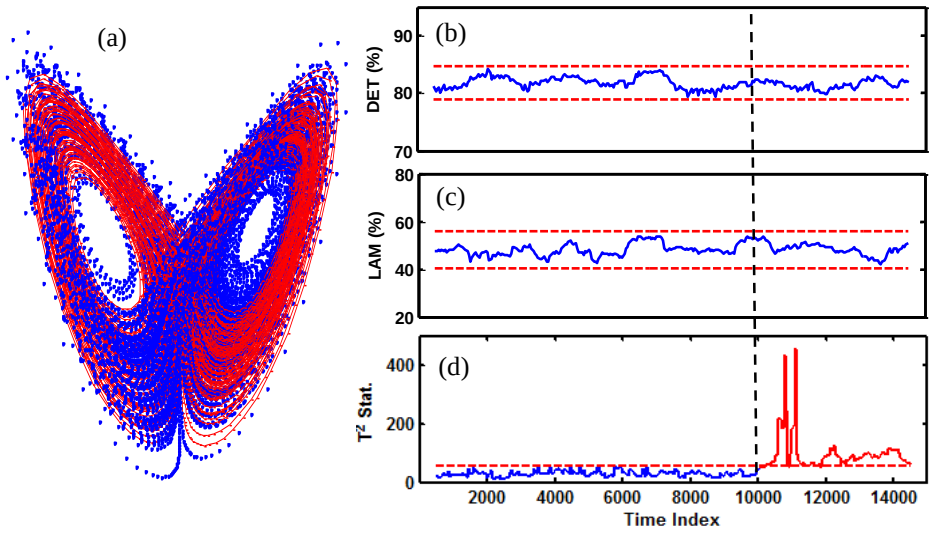


Figure 5.8 (a) State space of the Lorenz model with parameter changing from $\sigma = 10, \rho = 28, \beta = 8/3$ (blue dots) to $\sigma = 10, \rho = 27, \beta = 8/3$ (red lines), and evolution of (b) DET, (c) LAM and (d) Heterogeneous recurrence T^2 statistics with time.

Figure 5.8a shows the state space of the Lorenz system with the set of parameters changing from $\sigma = 10, \rho = 28, \beta = 8/3$ (blue dots – the first 10000 states) to $\sigma = 10, \rho = 27, \beta = 8/3$ (red lines – the next 5000 states). Due to the slight change of parameter ρ, it is difficult to visually discern the variations in the state space. Hence, we used the sliding window of size $w = 1000$ and a step size of 50 to compute the recurrence T^2 statistics and the RQA measures DET and LAM. The state space is partitioned into 15 sub-regions for heterogeneous recurrence analysis. When computing RQA measures, the threshold ε is chosen to preserve a constant recurrence rate of 5%. As shown in Figure 5.8b-d, the recurrence T^2 statistics yields superior performance to detect dynamic transition of Lorenz model than the traditional RQA measures DET and LAM. The red dashed lines in Figure 5.8b-d represent confidence intervals, and the black dashed line represents the time index when the model parameters change slightly. Notably, the RQA measures do not detect the slight change of parameters in the Lorenz system, while the recurrence T^2 statistics shows early evidences of an out-of-control alarm.

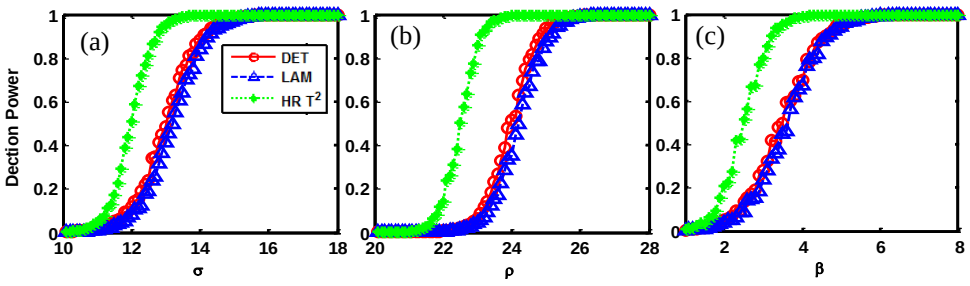


Figure 5.9 Performance comparison of detection power of DET, LAM and heterogeneous recurrence T^2 statistics for Lorenz models with varying parameter (a) σ, (b) ρ and (c) β.

Figure 5.9 shows the performance comparison of detection power of DET, LAM and heterogeneous recurrence T^2 statistics for Lorenz models with varying parameters. We used 1000 samples for both normal and dynamic transition cases, each of which has 500 data points in the time series. The detection power is the proportion of dynamic transitions correctly detected out of 1000 simulated cases. Figure 5.9 shows that the recurrence T^2 statistics yields a better performance than DET and LAM regarding the detection power of parametric variations in Lorenz models. Figure 5.9(a-c) show the detection power when changing the parameter σ, ρ, β respectively, while the other two are fixed. Parameter variations are based on the experimental design on Figure 5.4.

5.5.3 UPM Processes

Real-time control of UPM processes is critical to ensure the quality of surface finishes, improve the manufacturing performance and increase competitive advantage in the global market. Currently, offline methods are the common practice in the quality inspection of UPM surface finishes, e.g., microscope and profilometer. Notably, the microscope can only measure the surface in a small area. Also, the profilometer measures only a line across the surface, which is not sufficient to characterize the surface quality. Existing methods are offline inspection rather than real-time monitoring of UPM processes. As the UPM process is highly nonlinear and nonstationary,

there is an urgent need to develop new nonlinear methods and tools for detecting dynamic transitions in UPM processes.

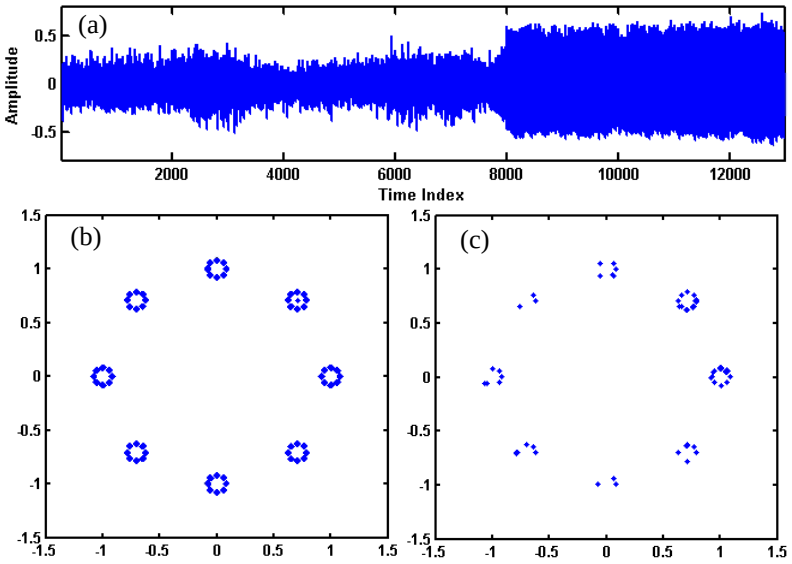


Figure 5.10 (a) Sensor signals with the transition from stable to unstable cutting at time index of 8000. (b) IFS addresses of stable cutting. (c) IFS addresses of unstable cutting.

In this case study, we firstly evaluate the performance of the heterogeneous recurrence method in detecting the transition from the regime of stable cutting to unstable cutting. Although the range of stable operating conditions can be broadened by an optimal machine design (e.g., increased stiffness and damping of the machine tool structure), the change of dynamic regimes can limit the range of stable cutting. Unstable cutting is one of the primary detriments to quality assurance of UPM surfaces. As shown in Figure 5.10a, the system evolves from stable cutting to the regime of unstable cutting at the time point of 8000. Notably, the amplitude of sensor signals increases abruptly. Unstable cutting leads to such an abrupt change in the roughness of surface finishes. We then reconstructed the state space from sensor signals with the embedding dimension $d = 3$ and time delay $\tau = 5$. The state space is partitioned into 8 regions using the Q-tree method for heterogeneous recurrence analysis. Figure 5.10 b and c show that fractal representations of stable and unstable cuttings are significantly different from each other. Figure 5.11 shows that the

recurrence T^2 statistics yields superior performance to detect dynamic transition from stable to unstable cutting than the traditional RQA measures DET and LAM. Although DET can detect the change, it is at a slower rate and smaller magnitude compared to the heterogeneous method. Here, we used sliding windows of size $w = 500$ and a step of 50 to compute recurrence T^2 statistics and RQA measures (i.e., DET and LAM). When computing RQA measures, the threshold ε is chosen to preserve a constant recurrence rate of 5%.

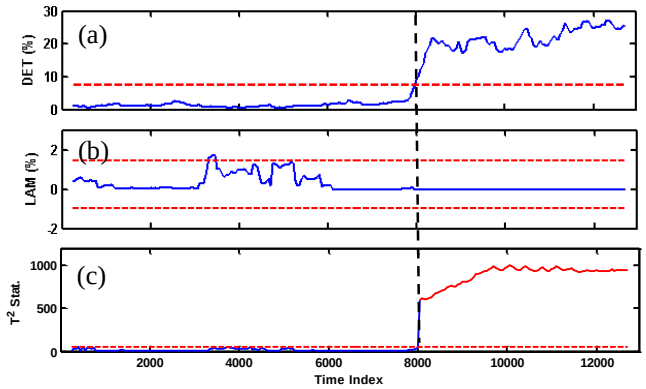


Figure 5.11 Time evolution of (a) DET, (b) LAM and (c) Heterogeneous recurrence T^2 statistics during the dynamic transition from stable to unstable cutting in UPM processes.

Furthermore, we designed a statistical experiment to investigate the relationship between heterogenous recurrence T^2 statistics and surface characteristics of 127 aluminum samples. Here, acoustic emission (AE) signals are collected with each aluminum sample. Notably, AE signals are the high-frequency microelastic pulses that are generated when a metallic material undergo plastic deformation. Low levels of plastic deformation cause sharp intermittent pulses, while large deformation causes continuous aperiodic emissions with spikes [104]. We reconstruct the AE state space with the embedding dimension $d = 3$ and time delay $\tau = 1$ and then compute recurrence T^2 statistics and RQA measures DET and LAM. Among the 127 samples, the first 116 samples have surface finish $R_a < 100$ nm and the last 11 samples have $R_a > 100$ nm.

As shown in Figure 5.12a-c, the recurrence T^2 statistics yields better performance in monitoring UPM dynamics than traditional RQA measures DET and LAM. The red dashed lines in Figure 5.12a-c represent confidence intervals of monitoring statistics. Notably, the RQA measures cannot detect the change of surface finish between $R_a < 100$ nm and $R_a > 100$ nm (red shaded area). Figure 5.12c shows that the proposed recurrence T^2 statistics is closely pertinent to the R_a values of UPM surface finishes. Heterogeneous T^2 statistics of rough surface finishes (i.e., $R_a > 100$ nm) are above the upper confidence bound. Experimental results show that the proposed recurrence method provides a new and effective means for monitoring UPM machining processes.

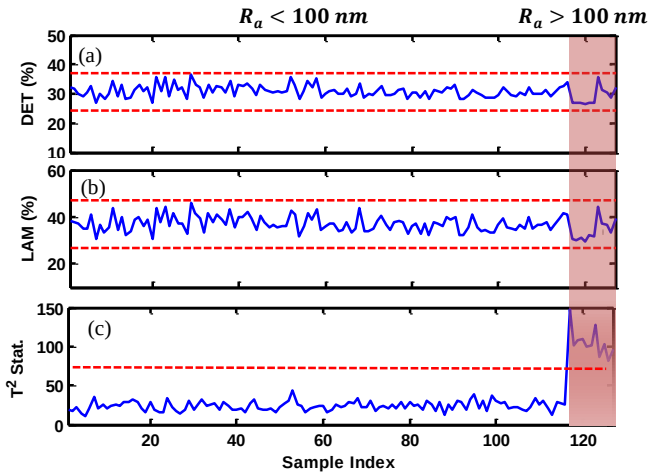


Figure 5.12 (a) DET, (b) LAM and (c) Heterogeneous recurrence T^2 statistics for monitoring the quality of surface finishes in UPM processes.

5.5.4 A Pilot Study on Cardiac Monitoring

Human heart is near-periodically beating to maintain vital living organs. Real-time sensing brings the proliferation of ECG data that contain rich signatures and "memories" of recurrence dynamics in cardiac operations. As shown in Figure 5.13, ECG signals possess some common characteristics: 1) Within one cycle, the waveform at different segments changes significantly. The reason is that different ECG segments (e.g., P, QRS and T waves) correspond to different stages of cardiac operations. 2) Between cycles, the ECG waveform is similar to each other but with

slight variations. Most existing approaches adopt linear methodologies to study recurrences in ECG signals. Traditional linear methods interpret the regular structure, e.g., dominant frequencies in the signals. They have encountered certain difficulties to capture the nonlinearity, nonstationarity and high-order variations. Instead, as shown in Figure 5.13, arrhythmia conditions (red/ dashed line) change significantly between cycles compared to normal conditions (blue/ solid line), which are more concerned with heterogeneous recurrence variations in cardiac processes.

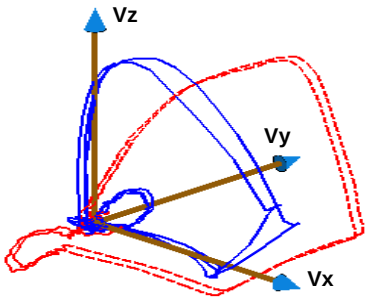


Figure 5.13 Near-periodic patterns of space-time ECG signals (blue/ solid line) and arrhythmia disturbances with recurrence variations (red/ dashed line).

However, very little work has been done to investigate heterogeneous recurrence variations in ECG signals and further conduct real-time monitoring of cardiac process and predict the incipience of arrhythmia conditions. As ECG is noninvasive and always obtainable with extremely portable equipment, real-time ECG monitoring, in days, months and even years, generates large amounts of data that provide an unprecedented opportunity for the early anticipation of rhythm disturbances. However, traditional ECG analytics are rooted in the clinic-centered care and have been mainly conducted by healthcare professionals empirically. Though it is effective to a certain level, cardiologists are not always available for real-time interpretation of ECG signals. In addition, big data poses significant challenges for human experts to accurately and precisely inspect all generated ECG signals for warning signs. There is an urgent need to develop novel real-time monitoring schemes of ECG signals to advance patient-centered cardiac care anytime anywhere.

As discussed in Section 5.4.2, this pilot study is aimed at developing the monitoring scheme of a specific arrhythmia condition, namely premature ventricular contraction. The incidence of PVCs are common for older people above the age of 50 (2.2%) and individuals with a history of heart diseases. The PVC is also called "skip a beat" that disrupts the heart rhythm. During the PVC, the heartbeat is initiated by Purkinje fibers in the ventricles rather than by the sinoatrial node as in the normal cardiac activity. As a result, ventricles contract before the atria fills blood to them, resulting in an inefficient circulation. As shown in Figure 5.13, ectopic PVC beats lead to space-time shifts of cardiac electrical activity. We evaluate the proposed methodology for the detection of recurrence variations pertinent to PVC ectopic beats using two ECG samples (record 101 and record 116) in the MIT-BIH Arrhythmia Database. The experimental results are shown in Figure 5.14 and Figure 5.15.

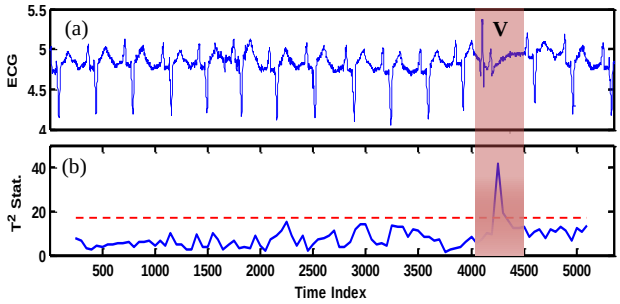


Figure 5.14 (a) ECG signal of record 101 and (b) associated time-variation of the heterogeneous recurrence T^2 statistics.

Figure 5.14a shows the ECG signals of record 101 with a time duration of 15.3 seconds. Notably, the shadowed area is annotated as "V" (i.e., premature ventricular contraction) by the cardiologist. These annotations are publically available in the MIT-BIH Arrhythmia Database. We compute the recurrence T^2 statistics from the reconstructed ECG state space with sliding windows of size $w = 500$ and a step size of 50 (embedding dimension $d = 3$ and delay $\tau = 1$). As shown in Figure 5.14b, the proposed recurrence T^2 statistics delineates the rhythm disturbances in the

heart and shows strong potentials for the detection of abnormal heartbeats in cardiac monitoring. Notably, the PVC segment yields larger T^2 statistics than normal heartbeats, which are above the upper confidence bound in the shadowed area.

Figure 5.15a shows another example of heterogeneous recurrence T^2 statistics on ECG signals with two annotated "V" (record 116, time duration 13.9 seconds). Similarly, the recurrence T^2 statistics are obtained from the reconstructed ECG state space with sliding windows of size $w = 500$ and a step size of 50 (embedding dimension $d = 3$ and delay $\tau = 1$). Notably, sliding window and step size depend on the sampling frequency of the ECG signals. If a different frequency is used for the acquisition of ECG signals, sliding window and step size should be adjusted accordingly. As shown in Figure 5.15b, the proposed recurrence T^2 statistics effectively detects two PVC heartbeats, which yield T^2 statistics higher than the upper confidence bound. This pilot study demonstrates the promising potentials of heterogeneous recurrence methods for cardiac monitoring, while a systematic investigation with more ECG samples will be conducted in our future work.

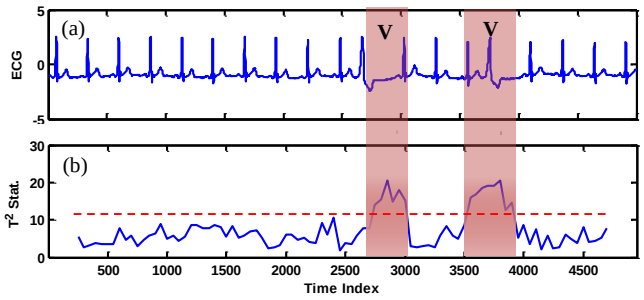


Figure 5.15 As in Figure 5.14 for ECG record 116.

5.6 Discussion and Conclusions

Few, if any, previous approaches investigated heterogeneous recurrence variations for online process monitoring and quality control in the literature. The presented work is the first-of-this-kind to not only extract heterogeneous recurrence patterns inherent in the nonlinear dynamic

processes but also develop a recurrence-based T^2 statistics for process monitoring and anomaly detection of the underlying process. First, the state space is originally reconstructed by sensing data collected from dynamical systems, and then recursively decomposed into hierarchical subspaces to delineate local recurrence regions. Second, we extract statistical quantifiers of heterogeneous recurrences from a new representation scheme. Third, we propose a novel heterogeneous recurrence T^2 statistics to simultaneously monitor two or more related quantifiers.

Experimental results on simulation studies of auto-regressive and Lorenz model and real-world case studies (UPM sensor and ECG signals) show that the proposed methodology not only captures heterogeneous recurrence patterns, but also effectively detects dynamical transitions in the nonlinear processes. In addition, the detection power of recurrence T^2 statistics is shown to be better than the traditional recurrence measures DET and LAM. Detecting surface characteristic variations in the UPM process and ectopic beats in real-world ECG signals further demonstrated that the proposed heterogeneous recurrence T^2 statistics has strong potential in the monitoring of manufacturing quality and cardiac conditions. In addition, there are many newly developed recurrence characteristics (e.g., recurrence network-based measures [120, 121], recurrence time statistics [122]) that may provide useful information on the detection of dynamical transitions. In the future work, it will be worthwhile to design a set of experimental studies to make a systematic comparison of available recurrence characteristics.

CHAPTER 6: SIMULATION MODELING AND PATTERN CHARACTERIZATION OF NONLINEAR SPATIOTEMPORAL DYNAMICS ON FRACTAL SURFACES

Engineered and natural systems often involve irregular and self-similar geometric forms, which is called fractal geometry. For examples, precision machining produces a visually flat surface, while which looks like a rough mountain in the nanometer scale under the microscope. Human heart consists of a fractal network of muscle cells, Purkinje fibers, arteries and veins. Cardiac electrical activity exhibits highly nonlinear and fractal behaviors. Although space-time dynamics occur on the fractal geometry, e.g., chemical etching on the surface of machined parts and electrical conduction in the heart, most of existing works modeled space-time dynamics (e.g., reaction, diffusion and propagation) on the Euclidean geometry (e.g., flat planes and rectangular volumes). This brings inaccurate approximation of real-world dynamics, due to sensitive dependence of nonlinear dynamical systems on initial conditions. In this paper, we developed novel methods and tools for the simulation modeling and pattern recognition of spatiotemporal dynamics on fractal surfaces of complex systems, which include (i) characterization and modeling of fractal geometry, (ii) fractal-based simulation and modeling of spatiotemporal dynamics, (iii) recognizing and quantifying spatiotemporal patterns. Experimental results show that the proposed methods outperform traditional modeling approaches based on the Euclidean geometry, and provide effective tools to model and characterize space-time dynamics on fractal surfaces of complex systems.

6.1 Introduction

Simulation modeling plays an important role in describing complex phenomena, predicting system behaviors and optimizing control actions to improve the performance of system operations. However, many natural phenomena occur on complex engineered and natural objects that show irregular and self-similar geometric forms, namely fractal geometry. For examples, chemical etching is a space-time process on the surface of silicon wafer. Although wafer surface is visually flat, it looks like a rough mountain in the nanometer scale under the microscope. Also, cardiac electrical activity varies across a fractal network of muscle cells, Purkinje fibers, arteries and veins. An anatomically realistic model of human heart involves irregular node spacing, complex boundaries, and regionally dependent conductivity. As such, electrical excitation and conduction in the human heart exhibit complex spatiotemporal dynamics. Spatiotemporal simulation provides a better understanding of natural phenomena and further leads to significant economic and societal impacts. For examples, simulation modeling of chemical etching dynamics will enable semiconductor manufacturing processes to make products with better quality and higher throughput. Most importantly, heart disease is responsible for 1 in every 4 deaths in the United States, amounting to an annual loss of $448.5 billion [1]. Simulation-based optimization of cardiac treatments will help improve the quality of healthcare services, reduce healthcare costs and promote the health of our society.

However, most of previous simulation models were developed for space-time dynamics (e.g., reaction, diffusion and propagation) on the Euclidean geometry, e.g., flat planes and rectangular volumes. Traditional simulation models consider finite-difference schemes for space-time computation on finely spaced grids in 3-D spatial dimensions. This brings inaccurate approximation of real-world dynamics on complex surfaces, due to the sensitive dependence of

nonlinear dynamical systems on initial conditions. It is well known that many engineered and natural objects show fractal behaviors. For examples, mountain terrain exhibits self-similar geometry across spatial scales. Heartbeat time series shows self-similar patterns across temporal scales [41]. Also, human heart consists of a fractal network of muscle cells, Purkinje fibers, arteries and veins. In the literature, fractal dimension is commonly used to characterize and model the complexity of a fractal object or time series. Fractal dimension describes the degree of self-similarity and quantifies the fractal patterns. As the complexity of objects increases, a single fractal dimension may not be sufficient for system characterization, but rather multi-fractal spectrums are used. Note that very little has been done to investigate the simulation model of spatiotemporal dynamics (e.g., cardiac electrical excitation and conduction) on the fractal geometry. There is an urgent need to investigate the modeling differences between fractal and Euclidean geometry and further develop effective simulation models of space-time dynamics on complex and irregular surfaces.

On the other hand, dimensionality reduction is an effective method to project high-dimensional data into the low-dimensional space, which is widely used in data analysis. However, little has been done to leverage dimensionality-reduction techniques to develop computationally-efficient simulation models of space-time dynamics on fractal surfaces. Notably, high-dimensional fractal surfaces can be isometrically mapped onto a low-dimensional one. As such, space-time simulation of system dynamics can be efficiently conducted in the low-dimensional space, which can then be one-to-one projected onto the original fractal surface. Nonetheless, traditional dimensionality-reduction techniques, e.g., principal component analysis (PCA) and multidimensional scaling (MDS), only preserve the data structure lying on a linear subspace of high-dimensional objects. For examples, PCA identifies a low-dimensional surface that best

preserves the data variance as measured in the high-dimensional surface. Classical MDS finds an embedding that preserves inter-point distances, which is equivalent to PCA when those distances are Euclidean. However, fractal surfaces in the high-dimensional space have essential nonlinear structures that pose significant difficulties to PCA and MDS methods. In other words, both PCA and MDS fail to reconstruct true degrees of freedom of fractal surfaces. Notably, isometric feature mapping (ISOMAP) is an extended version of PCA and MDS that handles nonlinear surfaces by preserving geodesic manifold distances [123]. Therefore, we propose to investigate isometric-graphing-based simulation models of space-time dynamics on fractal surfaces.

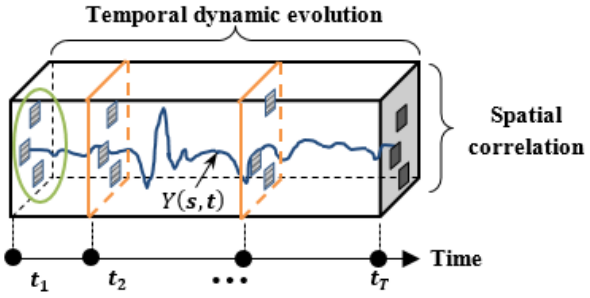


Figure 6.1 Spatiotemporal data of reaction-diffusion dynamics.

Furthermore, running simulation model of reaction-diffusion dynamics on fractal surfaces gives rise to spatiotemporal data $\{Y(\boldsymbol{s},t): \boldsymbol{s} \in R \subset \mathbb{R}^d, t \in T\}$ (see Figure 6.1), where the dependence of spatial domain R on time T symbolizes the condition where the spatial domain changes over time [75, 124, 125]. The next step is to extract useful information from spatiotemporal data. Traditionally, space-time indexed data is analyzed in two ways: (i) spatially-varying time series model $Y(\boldsymbol{s},t) = Y_{\boldsymbol{s}}(t)$, which separates the temporal analysis for each spatial location; (ii) temporally-varying spatial model $Y(\boldsymbol{s},t) = Y_t(\boldsymbol{s})$, which separates spatial analysis for each time point. The first model $Y_{\boldsymbol{s}}(t)$ shows specific interests in time-dependent patterns for each spatial location. The second model $Y_t(\boldsymbol{s})$ focuses more on space-dependent patterns for each

time point. However, both approaches are conditional methods studying either the space given time or time given space, and are limited in capturing space-time correlations [92].

This paper presents a new approach to simulate spatiotemporal dynamics of spiral wave formation and turbulent patterns on fractal surfaces. This present approach involves four key steps, namely, (i) fractal surface simulation; (ii) isometric graphing for surface characterization; (iii) dimensionality reduction for reaction-diffusion modeling and (iv) Spatiotemporal pattern recognition. First, fractal surfaces generated by random midpoint displacement algorithm in step (i) are characterized by isometric graph approach to extract geodesic distances (i.e., rough estimation of each data point's neighbor on the surface). Second, we simulate and model reaction-diffusion dynamics on the two-dimensional isometric graph, including spiral formation and turbulent patterns. Then, spatiotemporal dynamics are one-to-one projected onto the original fractal surfaces. Third, reaction-diffusion dynamics are investigated on the 3-D heart model to simulate spatiotemporal propagation and conduction of cardiac electrical activity. Finally, we developed new methods to quantify spatiotemporal patterns on irregular surfaces and investigate how fractal characteristics change these patterns.

This paper is organized as follows: Section 6.2 presents the research background. Section 6.3 introduces the research methodology. Section 6.4 shows the materials and experimental design used in this investigation. Section 6.5 presents experimental results, and Section 6.6 discusses and concludes this investigations.

6.2 Research Background

6.2.1 Fractal Characterization and Modeling

Euclidean geometric objects are comprised of lines, planes, cubes, cylinders, spheres with integer dimensions, 1, 2, or 3. However, there are many irregular objects that do not conform to

Euclidean geometry. Mandelbrot firstly introduced fractal as "A rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approximately) a reduced/size copy of the whole" [126]. Note that fractals are not limited to spatial patterns, but can also describe the processes in time. Most of existing works focused on the following two aspects:

(1) *Characterization of fractal dimension:*

The fractal dimension is a statistical measure describing how the patterns change with the scale at which it is measured. Monofractal refers to the homogeneous self-similarity across scales, characterized by a single fractal dimension. Examples of monofractal dimension used for system characterization include physiology [127], gait dynamics [128], and geology [129]. However, multifractal signals or objects require an infinite number of indices (i.e., singularity spectrum) to characterize their scaling properties. Example applications of multifractal spectrum include heart rate variability [41], ECG signals [130], financial markets [131] and material sciences [132]. Note that box counting method is widely used to characterize the fractal dimension [126, 133]. In the multifractal case, the probability, P, of a number of measures appearing in a box, i, varies with the box size, ϵ, according to scaling exponents, α_i, which changes over the set, as $\alpha_i \propto \frac{\log P_{i,\epsilon}}{\log \epsilon^{-1}}$. In contrast, the scaling exponent does not change meaningfully over the set for monofractals.

(2) *Modeling the fractal object or process:*

Fractals are usually modeled with an iterative or recursive construction or algorithm. Examples of fractal models to generate rough surfaces include shear displacement algorithm [134], diamond-square algorithm [135], and Fourier filtering algorithms [136]. These algorithms are widely used in computer games or movies (e.g., star trek II) to simulate realistic mountains or landscapes. In addition, structure function method, i.e., the random cascade model was utilized to simulate multifractal self-similar behaviors in the heart rate dynamics [137, 138]. Heart rate time

series $r(t)$ are modeled as a product of J cascade components: $r_J(t) = \prod_{j=1}^{J} \omega_j(t)$ and $\omega_j(t) = 1 + \xi_j$, where $\xi_j, j = 1, \cdots, J$ are independent Gaussian variables with $\langle \xi_j \rangle = 0$ and $\langle \xi_i \xi_j \rangle = \delta_{ij} \sigma_j^2$, where δ_{ij} is the Kronecker delta.

However, fractal literature focuses on either the extraction of fractal dimensions for system characterization, or the modeling of fractal geometry. Few, if any, previous studies investigated the simulation model of spatiotemporal dynamics (e.g., cardiac electrical propagation and conduction) on fractal geometry. Further, modeling differences between fractal and Euclidean geometry have not been fully investigated before. It is critical to investigate how fractal characteristics impact the patterns of spatiotemporal dynamics on irregular surfaces.

6.2.2 Dimensionality Reduction

Both simulated and real-world spatiotemporal dynamics bring the proliferation of big data, which is high-dimensional and difficult to visualize and interpret. In order to explore meaningful patterns underlying big data, many previous works developed the methods and tools for dimensionality reduction. Examples of dimensionality reduction approaches include principal component analysis (PCA) [139], multidimensional scaling (MDS) [140], self-organizing map (SOM) [72] and isometric feature mapping (ISOMAP) [123]. PCA uses an orthogonal transformation to find the projected data that captures the principal variations in the original high-dimensional space. Classical MDS transforms the high-dimensional vectors into a low-dimensional embedding that preserves Euclidean distances. Note that the PCA and MDS methods only characterize linear subspaces in the high-dimensional data. However, SOM neural network automatically organizes a low-dimensional map according to the inherent structures in the high-dimensional data. Furthermore, isometric graphing extends the MDS by preserving geodesic manifold distances, which better characterize the nonlinear degrees of freedom that underlie complex natural observations.

However, most of previous studies focused on the reduction of high-dimensional data and then the extraction of useful information from the low-dimensional data. Few previous approaches considered the construction of simulation models in the low-dimensional space, and further investigated the projection onto the original high-dimensional space. This will be an original contribution of the present paper that is expected to bring computationally efficient simulation models, and potentially be transplanted to other simulation domains.

6.3 Research Methodology

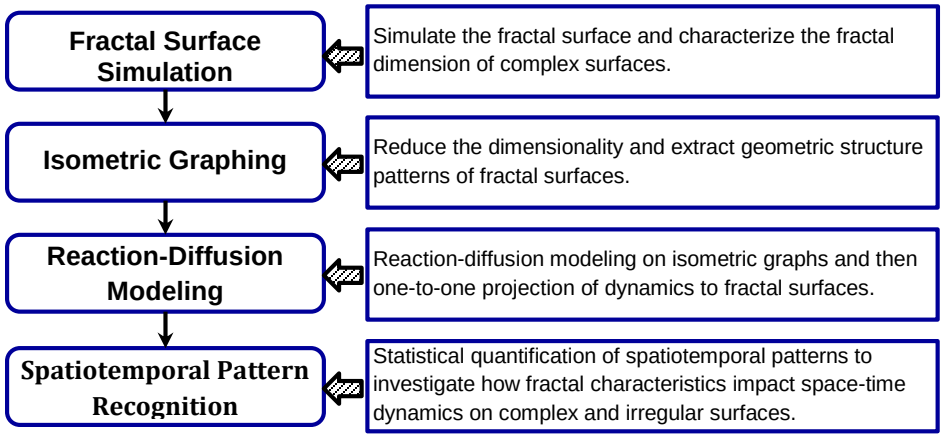


Figure 6.2 Flow chart of research methodology.

This present paper studies the simulation modeling of spatiotemporal reaction-diffusion dynamics on irregular surfaces that are generated from the fractal geometry, as opposed to traditional surfaces in the Euclidean geometry. Few, if any, previous investigations have focused on simulation modeling of fractal surfaces in the reduced-dimension space and further characterized the spatiotemporal dynamic patterns on the fractal surfaces. As shown in Figure 6.2, this present paper is embodied by four core components focusing on simulation modeling on fractal surfaces, including fractal surface simulation, isometric graphing, reaction-diffusion modeling, and spatiotemporal pattern recognition. All four components are eventually integrated together to develop better simulation models on fractal surfaces and better understanding of spatiotemporal phenomena in real-world complex systems.

6.3.1 Fractal Surface Simulation

Fractals objects have irregular geometric forms, and cannot be well described using topological dimensions. However, fractal objects often look similar regardless of the magnification, which is so-called self-similar behaviors. Many real-world objects exhibit self-similarity, e.g., scribbles, dust, ocean waves, or clouds. If one zooms in or out the fractal set, there is a similar appearance in the geometric shape. Hence, fractal dimension is introduced to describe such "infinitely complex" fractal objects (or shape). It is worth mentioning that fractal dimension is not topological, and needs not to be an integer. Fractal sets have theoretical dimensions that exceed their topological dimensions.

As self-similarity across scales is a typical characteristic of fractals, fractal dimension measures the changes of coverings with respect to the scaling factor. It also characterizes the space-filling capacity of a fractal object. In the literature, the box-counting method is widely used to estimate the relationship between scaling and covering so as to estimate the fractal dimension of an irregular object [141]. The basic idea is to cover a fractal set with measure elements (e.g., box) in different scales and examine how the number of boxes changes with respect to the scaling factor. If $\mathrm{N}(\mathrm{a})$ is the number of boxes that are needed to cover a fractal object at the scale a, then the fractal dimension D_F specifies how $\mathrm{N}(\mathrm{a})$ changes with respect to the scaling factor a as: $\mathrm{N}(\mathrm{a}) \propto (1/a)^{\mathrm{D}_F}$. In general, the box-counting method defines the fractal dimension as

$$\mathrm{D}_F \coloneqq \lim_{a \to 0} \frac{\ln N(a)}{\ln(1/a)}$$

In order to model space-time dynamics on irregular surfaces, we will first need to generate the fractals. This present investigation utilizes the random midpoint displacement method [135] to generate various types of fractal surfaces. Figure 6.3 shows the detailed steps of the algorithm, which starts with a square with pixel values at four corners (green triangles in Figure 6.4a) drawn

from a Gaussian distribution $N(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 1$. Then, we recursively generate the center points and edge points at each step until the stopping criteria is satisfied. The fractal surface generated is monofractal and its fractal dimension D_F can be uniquely determined by the parameters (i.e., Hurst exponent H) used in the algorithm.

Initialization:

// Start with an square with pixel values at four corners drawn from a Gaussian distribution $N(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = 1$.

While $i \leq$ desired iterations

Center point:

// The center is the average of its four neighbors plus a random value δ_i generated from a Gaussian distribution $N(\mu, \sigma_i^2)$, where $\sigma_i^2 = \frac{1}{2^{2H(i+1)}}\sigma^2$.

Edge point:

// The edge point is the average of its neighbors plus a random value δ_i generated from a Gaussian distribution $N(\mu, \sigma_i^2)$, where $\sigma_i^2 = \frac{1}{2^{2H(i+1)}}\sigma^2$.

$i = i + 1$

End

Calculate the fractal dimension:

// The simulated fractal surface is monofractal and the fractal dimension D_F is determined as $D_F = 3 - H$.

Figure 6.3 Random midpoint displacement algorithm for generating fractal surfaces.

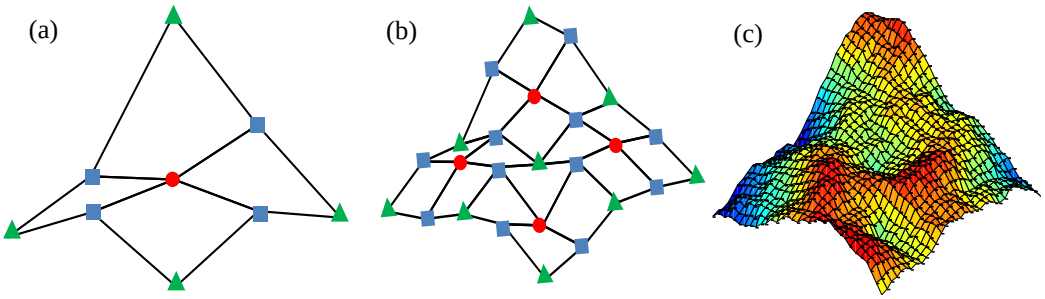


Figure 6.4 Recursive steps to generate fractal surfaces. (a) the first iteration, (b) the second iteration, and (c) the fifth iteration.

Three key steps in this recursive algorithm (i.e., center points, edge points and determining fractal dimension) to generate fractal surfaces are described as follows:

(1) *Recursive placement of the center point:*

For each square, a center point (e.g., red dot Figure 6.4a) is placed in the middle of X-Y plane, and its height is the average of its four corners (i.e., green triangles in Figure 6.4a) plus a random value δ_i generated from a Gaussian distribution $N(\mu, \sigma_i^2)$. Notably, the variance σ_i^2 of random perturbation in the ith iteration is also recursively modified to obtain a fractal Brownian motion (fBm) surface as

$$\sigma_i^2 = \frac{1}{2^{2H(i+1)}}\sigma^2 \tag{1}$$

where $H(0 \leq H \leq 1)$ is the Hurst exponent. Then, this process is iteratively repeated for each subsquare. Figure 6.4b shows the second iteration to place four center points (i.e., 4 red dots). Each subsquare is formed by 4 corner points (i.e., green triangles) in Figure 6.4b. The heights of center points are the average of four corners in each subsquare plus a random perturbation. It may be noted that the first iteration has the biggest influence to the shape of the simulated surface (i.e., large-scale effects), while the following iterations have smaller influence (i.e., small-scale details).

(2) *Recursive placement of edge points:*

The position of an edge point is in the middle of 2 corner points. The heights of edge points (blue rectangles in Figure 6.4a) are calculated as the average of neighbors plus a random perturbation δ_i generated from a Gaussian distribution $N(\mu, \sigma_i^2)$. The variance σ_i^2 is also recursively modified as shown in Eq. (1). Figure 6.4a shows the edge points (i.e., 4 blue rectangles) whose heights are calculated as the average of three nearest neighbors. This algorithm leads to nonstationary steps that iteratively generate the fractal surface. Figure 6.4c shows a fractal surface generated from the algorithm with 33×33 locations, i.e., $\boldsymbol{s}_i = (x_i, y_i, z_i), i = 1,2, \ldots, 1069$, at the fifth iteration. It may be noted that the surface color represents the height. The hotter color indicates a higher surface, and the cooler color represents a lower surface.

(3) *Determine the fractal dimension:*

The fractal surface provides a better approximation of real-world geometric objects and landscape. Next, the Hurst exponent H uniquely determines the fractal dimension of generated surfaces in the random midpoint displacement algorithm. The singularity spectrum $\mathrm{D}(\mathrm{H})$ provides a statistical distribution of Hurst exponents $\mathrm{H}(\boldsymbol{s})$ at the locations $\boldsymbol{s}$ such that $\mathrm{H}(\boldsymbol{s}) = \mathrm{H}$, i.e., $\mathrm{D}(\mathrm{H}) = \mathrm{D}_F\ (\{\boldsymbol{s}\colon \mathrm{H}(\boldsymbol{s}) = \mathrm{H}\})$, where D_F is the fractal dimension. As shown in Figure 6.3, the algorithm iteratively generates random perturbations on the fBm surface and all the locations $\mathbf{s}$ have a single Hurst exponent H (also see Eq. (1)). As such, the fractal dimension D_F of fBm surfaces is determined as $\mathrm{D}_F = 3 - H$. For examples, if the Hurst exponent H is 0.05 in Eq. (1), then the fractal dimension D_F of fBm surfaces will be 2.95. If the Hurst exponent H is 0.95 in Eq. (1), then the fractal dimension D_F of fBm surfaces will be 2.05. In this investigation, we will generate different types of fractal surfaces with fractal dimensions from 2.05 to 2.95, step size 0.05 to study how fractal characteristics impact spatiotemporal dynamic patterns on complex surfaces.

6.3.2 Isometric Graphing for Surface Characterization

The fractal surface is nonlinear and nonstationary, which significantly challenges the characterization of geometric structures and surface patterns. In this present investigation, we introduce the isometric graphing approach [123] to extract inherent geodesic properties of the high-dimensional surface. As shown in Figure 6.5a, the Euclidean distance between two locations is represented by the blue dash line, and their geodesic distance is shown as red solid line. It may be noted that geodesic distance is the true distance for the reaction-diffusion dynamics (or electrical conduction) to travel between two locations instead of the Euclidean distance. As such, it is imperative to characterize and quantify the intrinsic geometry for reaction-diffusion modeling on the fractal surface. We obtain nonlinear and nonstationary characteristics of fractal surfaces

asymptotically by capturing geodesic manifold distances between all the locations. These geodesic distances are further utilized to construct the low-dimensional embedding of fractal surfaces. The proposed approach not only captures nonlinear degrees of freedom of fractal surfaces, but also efficiently yields globally optimal solutions for dimensionality reduction. This greatly facilitates the simulation modeling of reaction-diffusion dynamics in the reduced dimension that will be detailed in Section 6.3.3.

The key steps in the isometric graphing algorithm to characterize the fractal surface and construct the low-dimensional embedding are described as follows:

(1) *Build the graph of k nearest neighbours:*

Define the k-nearest neighbor graph of the fractal surface by connecting each location $\boldsymbol{s}_i$ to its k nearest neighbors $\boldsymbol{s}_j,\ j \in \{k \text{ nearest neighbors of } i\}$, and compute the distance between locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ as $d(\mathrm{i,j})$, i.e., $d(\mathrm{i,j}) = \|\boldsymbol{s}_i - \boldsymbol{s}_j\| = \left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2\right)^{\frac{1}{2}}$. Otherwise, spatial locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ are disconnected.

(2) *Compute the shortest path between two locations $d_G(i,j)$ on the surface:*

First, we initialize $d_G(i,j) = d(i,j)$ if locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ are connected in the k-nearest neighbor graph; $d_G(i,j) = \infty$ otherwise. Then, we utilized the Dijkstra's algorithm [142] to compute the shortest path between any two locations. For each $k = 1, \cdots, N$, all entries $d_G(i,j)$ are replaced by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The final matrix $D_G = \{d_G(i,j)\}$ will contain the distance of the shortest path between any two locations on the fractal surface.

(3) *Construct low-dimensional embedding:*

Here, the objective is to derive the isometric graph in the 2-dimensional plane that preserves the shortest distance matrix $\mathrm{D_G}$ on the 3-D fractal surface. It may be noted that we are reducing the dimension, while preserving the inherent geodesic properties of fractal surfaces. Let

$\tilde{\boldsymbol{s}}_i$ and $\tilde{\boldsymbol{s}}_j$ denote the locations in the isometric graph in the 2-dimensional plane. Then, the objective function is formulated as:

$$min \sum_{i<j} (\left\| \tilde{\boldsymbol{s}}_i - \tilde{\boldsymbol{s}}_j \right\| - d_G(i,j)) \,;\; i,j \;\in [1, \boldsymbol{N}] \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm. To solve this optimization problem, the Gram matrix M is firstly reconstructed from the N×N shortest distance matrix $\mathrm{D_G}$: $M = -\frac{1}{2} Q \mathrm{D}_G^{(2)} Q$, where the centering matrix $\mathrm{Q} = \left\{\delta_{ij} - 1/N\right\}_{i,j=1}^{N}$, δ_{ij} is the Kronecker delta. The $\mathrm{D}_G^{(2)}$ is a squared matrix and each element is $\{d_G^2(i,j)\}_{i,j=1}^{N}$ (i.e., the squares of $d_G(i,j)$ in the matrix $\mathrm{D}_G^{(2)}$). The element M_{ij} in matrix M is:

$$M(i,j) = -\frac{1}{2}\left[d_G^2(i,j) - \frac{1}{N}\sum_{k=1}^{N} d_G^2(i,k) - \frac{1}{N}\sum_{k=1}^{N} d_G^2(k,j) + \frac{1}{N^2}\sum_{g=1}^{N}\sum_{h=1}^{N} d_G^2(h,g) \right] \tag{3}$$

It is known that the Gram matrix M is defined as the scalar product $M = \tilde{S}\tilde{S}^T$, where the matrix $\tilde{S}$ minimizes the aforementioned objective function. The Gram matrix M can be further decomposed as: $M = V\Lambda V^T = V\sqrt{\Lambda}\sqrt{\Lambda}V^T$, where $V = [v_1, v_2, \dots, v_{D'}]$ is a matrix of eigenvectors and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_{D'})$ is a diagonal matrix of eigenvalues. Then, the matrix of feature vectors is obtained as: $\tilde{S} = V\sqrt{\Lambda}$. In this present investigation, the reduced dimensionality D' is 2 because we derive the isometric graph in the 2-dimensional plane from the 3-D fractal surface.

As shown in Figure 6.5, the simulated fractal surface (Figure 6.5a) is converted into a isometric graph in the two-dimensional plane (Figure 6.5b), i.e., $\tilde{\boldsymbol{s}}_{\boldsymbol{i}} = (x_i, y_i), i = 1,2, \dots, N$. The isometric graph preserves the shortest path matrix of geodesic distances $\mathrm{D_G}$, thereby preserving the inherent geodesic properties of fractal surfaces. The gray nodes represent all N spatial

locations, and the gray lines show the connections between each pair of neighboring locations (with $K = 4$ nearest neighbors and $N = 1089$ data points). Moreover, the geodetic distance of two locations in Figure 6.5a is embedded in the isometric graph in the two-dimensional plane (see the red solid line in Figure 6.5b), where the blue solid line in Figure 6.5b represents a Euclidean approximation to the true geodetic distance. Therefore, the proposed approach of isometric graphing algorithm not only characterizes the inherent geodesic structure of fractal surfaces, but also provides an optimal solution for dimensionality reduction that facilitates the simulation modeling of reaction-diffusion dynamics in the reduced dimension (see next Section 6.3.3).

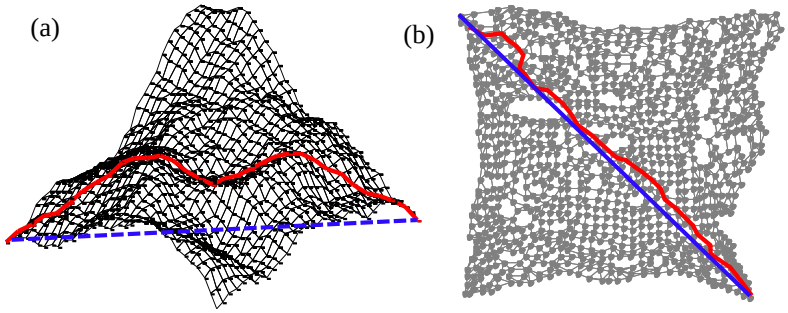


Figure 6.5 (a) Euclidean distance (blue dash line) and geodesic distance (red solid line) between two locations on the fractal surface. (b) Isometric graph in the 2-dimensional plane that preserves the geodesic distance on the 3-D fractal surface.

6.3.3 Reaction-Diffusion Modeling in the Reduced Dimension

Next, we simulate and model reaction-diffusion dynamics on the two-dimensional isometric graph (see Figure 6.5b). This present investigation utilized a two-component reaction-diffusion model on a bounded domain Ω (i.e., isometric graph) with concentration variables u, v and nonlinear reaction expressions f, g as

$$\begin{aligned} \frac{\partial u}{\partial t} &= f(u,v) + \mathcal{D}_1 \nabla^2 u \\ \frac{\partial v}{\partial t} &= g(u,v) + \mathcal{D}_2 \nabla^2 v \end{aligned} \tag{4}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are the diffusion constants of concentration variables u and v, $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator in the two-dimensional space. We discretize the reaction-diffusion model at the time index $t = 1,2,\dots,T$ and locations $i = 1,2,\dots,N$ as

$$\begin{aligned}(u_i^t - u_i^{t-1})/\delta t &= f(u_i^{t-1}, v_i^{t-1}) + \mathcal{D}_1 \nabla^2 u_i^t \\ (v_i^t - v_i^{t-1})/\delta t &= g(u_i^{t-1}, v_i^{t-1}) + \mathcal{D}_2 \nabla^2 v_i^t\end{aligned} \tag{5}$$

Then the reaction-diffusion model can be rewritten as follows:

$$\begin{aligned}(1 - \mathcal{D}_1 \delta t \cdot \nabla^2) u_i^t &= u_i^{t-1} + \delta t \cdot f(u_i^{t-1}, v_i^{t-1}) \\ (1 - \mathcal{D}_2 \delta t \cdot \nabla^2) v_i^t &= v_i^{t-1} + \delta t \cdot g(u_i^{t-1}, v_i^{t-1})\end{aligned} \tag{6}$$

The matrix form of differential equations will become

$$\begin{pmatrix} \boldsymbol{B_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B_2} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}^t \\ \boldsymbol{V}^t \end{pmatrix} = \begin{pmatrix} \boldsymbol{U}^{t-1} + \delta t \cdot \boldsymbol{F} \\ \boldsymbol{V}^{t-1} + \delta t \cdot \boldsymbol{G} \end{pmatrix} \tag{7}$$

where $\boldsymbol{U}^t = [u_1^t, \dots, u_N^t]^T$ and $\boldsymbol{V}^t = [v_1^t, \dots, v_N^t]^T$, $\{\boldsymbol{F}\}_i = f(u_i^{t-1}, v_i^{t-1})$ and $\{\boldsymbol{G}\}_i = g(u_i^{t-1}, v_i^{t-1})$ for $i = 1,2,\dots,N$, δt is the time step, $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ are the matrices of constant coefficients pertinent to the domain Ω and diffusion constants $\mathcal{D}_1, \mathcal{D}_2$:

$$\boldsymbol{B_1} = \boldsymbol{I} - \mathcal{D}_1 \delta t \cdot \begin{bmatrix} \boldsymbol{\nabla}_{u_1}^2 \\ \vdots \\ \boldsymbol{\nabla}_{u_N}^2 \end{bmatrix}, \boldsymbol{B_2} = \boldsymbol{I} - \mathcal{D}_2 \delta t \cdot \begin{bmatrix} \boldsymbol{\nabla}_{v_1}^2 \\ \vdots \\ \boldsymbol{\nabla}_{v_N}^2 \end{bmatrix} \tag{8}$$

where $\boldsymbol{I}$ is the identity matrix of size N, $\boldsymbol{\nabla}_{u_i}^2$ and $\boldsymbol{\nabla}_{v_i}^2$ are scalars of size $1 \times N$, denoting the Laplacian operator of u_i and v_i respectively, i.e., $\boldsymbol{\nabla}_{u_i}^2 \cdot \boldsymbol{U}^t = \nabla^2 u_i^t$ and $\boldsymbol{\nabla}_{v_i}^2 \cdot \boldsymbol{V}^t = \nabla^2 v_i^t$. Hence, concentration variables $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$ at time step t can be solved through the linear equations $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, where $\mathbf{A} = \begin{pmatrix} \boldsymbol{B_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B_2} \end{pmatrix}$ is a constant matrix and $\mathbf{b} = \begin{pmatrix} \boldsymbol{U}^{t-1} + \delta t \cdot \boldsymbol{F} \\ \boldsymbol{V}^{t-1} + \delta t \cdot \boldsymbol{G} \end{pmatrix}$ that are from the concentration variables $\boldsymbol{U}^{t-1}$ and $\boldsymbol{V}^{t-1}$ at the previous time step $\mathrm{t} - 1$. Here, the square matrix $\mathbf{A}$ can be decomposed as $\mathbf{A} = \mathbf{LU}$ through LU factorization, where $\mathbf{L}$ is a lower triangular matrix and

U is an upper triangular matrix. The linear equation $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ can be solved by forward and backward substitution [143].

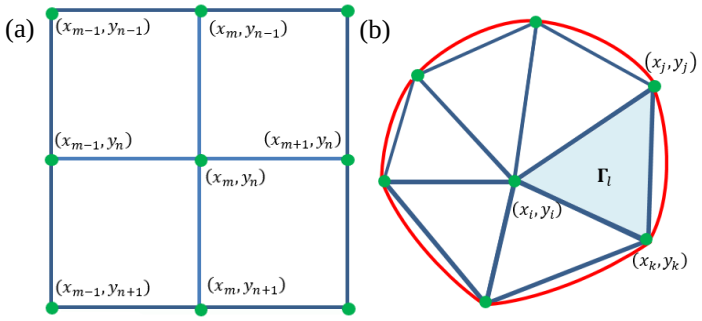


Figure 6.6 Illustrations of the domain Ω with (a) regular grids and (b) irregular grids.

However, it is necessary to formulate the matrices $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ for the specific domain Ω and diffusion constants $\mathcal{D}_1, \mathcal{D}_2$. As shown in Figure 6.6a, the discrete Laplacian operator at different locations, i.e., corner, edge and middle, in the regular grid structure can be written as the five-point central difference approximation

$$\text{Corner: } \nabla^2 u^t_{m-1,n-1} = \frac{u^t_{m,n-1} + u^t_{m-1,n} - 2u^t_{m-1,n-1}}{(\delta x)^2}$$

$$\text{Edge: } \nabla^2 u^t_{m-1,n} = \frac{u^t_{m-1,n-1} + u^t_{m,n} + u^t_{m-1,n+1} - 3u^t_{m-1,n}}{(\delta x)^2} \tag{9}$$

$$\text{Middle: } \nabla^2 u^t_{m,n} = \frac{u^t_{m+1,n} + u^t_{m-1,n} + u^t_{m,n+1} + u^t_{m,n-1} - 4u^t_{m,n}}{(\delta x)^2}$$

where δx is the differences between two adjacent grids for both x and y directions. Similarly, $\nabla^2 v^t_{m,n}$ can be obtained by substituting v for u in Eq. (9). For a $\mathrm{N} = \mathrm{J} \times \mathrm{J}$ regular grid, Eq. (8) can be written as

$$\boldsymbol{B_1} = \boldsymbol{I}_N - \frac{\mathcal{D}_1 \delta t}{(\delta x)^2} \cdot \begin{bmatrix} E & \boldsymbol{I}_J & & & & & \\ \boldsymbol{I}_J & C & \boldsymbol{I}_J & & & & \\ & \boldsymbol{I}_J & C & \boldsymbol{I}_J & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \boldsymbol{I}_J & C & \boldsymbol{I}_J & \\ & & & & \boldsymbol{I}_J & C & \boldsymbol{I}_J \\ & & & & & \boldsymbol{I}_J & E \end{bmatrix} \tag{10}$$

where $\boldsymbol{I}_J$ is the identity matrix of size J, E and C are J × J matrices as

$$E = \begin{bmatrix} -2 & 1 & & & \\ 1 & -3 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -3 & 1 \\ & & & 1 & -2 \end{bmatrix}, C = \begin{bmatrix} -3 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -3 \end{bmatrix} \tag{11}$$

However, isometric graph in the two-dimensional domain Ω has irregular grid structures. As shown in Figure 6.6b, the domain $\Omega = \{\Gamma_l\}_{l=1}^{L}$ involves non-overlapping closed triangles Γ_l, where triangles must intersect along a common edge, a common vertex, or not at all. The five-point central difference approximation in Eq. (9) is not applicable for the irregular grid. Hence, we introduce the finite element method (FEM) to develop approximate solutions for the matrices $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$.

Let $\phi_1, \phi_2, \dots, \phi_N$ be the basis functions satisfying $\phi_i(\tilde{\boldsymbol{s}}_j) = \delta_{ij}$, where $\tilde{\boldsymbol{s}}_j, j = 1,2, \dots, N$ are locations (triangle vertices) in the domain Ω and δ_{ij} is the Kronecker delta function, i.e., $\delta_{ij} = 1$ when $i = j$, otherwise, $\delta_{ij} = 0$. Then concentration variables u and v can be rewritten as a linear combination of the basis functions, i.e., $u(\tilde{\boldsymbol{s}}) = \sum_{i=1}^{N} u_i \phi_i(\tilde{\boldsymbol{s}})$. The Laplacian operator of u_i is derived as

$$\nabla^2 u_i = -\sum_{j=1}^{N} u_j \cdot \left(\int_\Omega \phi_i(\tilde{\boldsymbol{s}}) d\Omega \right)^{-1} \cdot \left(\int_\Omega \nabla\phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla\phi_j(\tilde{\boldsymbol{s}}) d\Omega \right) = \boldsymbol{\nabla}_{u_i}^2 \cdot \boldsymbol{U} \tag{12}$$

where $\boldsymbol{U} = [u_1, u_2, \dots, u_N]^T$. The detailed derivation of Eq. (12) is in Appendix B.

If we define the diagonal lumped mass matrix as

$$\boldsymbol{M} = [M_{ii}]_{i=1}^{N} = \left[\int_\Omega \phi_i d\Omega \right]_{i=1}^{N} \tag{13}$$

and the symmetric stiffness matrix as

$$\boldsymbol{K} = \left[K_{ij}\right]_{i,j=1}^{N} = \left[\int_\Omega \nabla\phi_i \cdot \nabla\phi_j d\Omega \right]_{i,j=1}^{N} \tag{14}$$

Then we have $\begin{bmatrix} \nabla^2_{u_1} \\ \vdots \\ \nabla^2_{u_N} \end{bmatrix} = -\boldsymbol{M^{-1}K}$, and $\begin{bmatrix} \nabla^2_{v_1} \\ \vdots \\ \nabla^2_{v_N} \end{bmatrix} = -\boldsymbol{M^{-1}K}$ according to Eq. (12). The practical calculation of matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ is given in Appendix C. Therefore, the matrices $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ can be approximated as

$$\boldsymbol{B_1} = \boldsymbol{I} + \mathcal{D}_1 \delta t \cdot \boldsymbol{M^{-1}K}, \boldsymbol{B_2} = \boldsymbol{I} + \mathcal{D}_2 \delta t \cdot \boldsymbol{M^{-1}K} \tag{15}$$

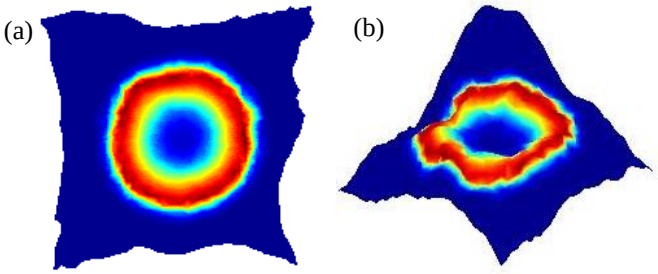


Figure 6.7 Reaction-diffusion modeling on the (a) isometric graph and (b) fractal surface.

To this end, we can utilize the LU factorization to solve the Eq. (7) and simulate reaction-diffusion dynamics on the isometric graph with irregular grid structures. Figure 6.7a shows the $\boldsymbol{U}^t = [u_1^t, \dots, u_N^t]^T$ at each location $\tilde{\boldsymbol{s}}_i = (x_i, y_i), i = 1,2, \dots, N$ at a specific time index t on the isometric graph. Note that geodesic distances between spatial locations are preserved and these locations have the one-to-one correspondence from the isometric graph to the fractal surface. Therefore, reaction-diffusion variables $\boldsymbol{U}^t$ are then one-to-one mapped onto the fractal surface at locations $\boldsymbol{s}_i = (x_i, y_i, z_i)$ (see Figure 6.4b). It may be noted that wave patterns are shown to be irregular and chaotic while they are mapped onto the high-dimensional fractal surface.

6.3.4 Spatiotemporal Pattern Recognition

New reaction-diffusion model in the reduced dimension provides an effective tool to simulate spatiotemporal dynamics on fractal surfaces, e.g., propagation and conduction of cardiac electrical activity. Next, it is imperative to quantify such spatiotemporal patterns on irregular surfaces and investigate how fractal characteristics change these patterns. In this present paper, we

propose manifold learning to extract low-dimensional basis for describing space-time dynamics in the high-dimensional space. This attractive technique leverages similarity or dissimilarity in the spatiotemporal data to make inference about system dynamics that are otherwise not apparent in the high-dimensional space. Manifold learning not only facilitates the visualization of high-dimensional dynamics but also provides statistical quantifiers of spatiotemporal patterns on fractal surfaces. Let $\mathrm{Y}(\boldsymbol{s}_i, t), \mathrm{t} = 1, \cdots, \mathrm{T}$ denote the spatiotemporal data, where $\boldsymbol{s}_i$ is the spatial location $(x_i, y_i, z_i), \mathrm{i} = 1, \cdots, \mathrm{N}$, for N locations and T is the total number of snapshots forming an ensemble with time index t (see Figure 6.8). The similarity or dissimilarity in the spatiotemporal data is quantified by the distance matrix D_T between different time indices.

$$\mathrm{D}_T(l, m) = \left[\sum_{i=1}^{N} \left(\mathrm{Y}(\boldsymbol{s}_i, t_l) - \mathrm{Y}(\boldsymbol{s}_i, t_m)\right)^2\right]^{1/2} \tag{20}$$

where $\mathrm{D}_T(l, m)$ is a hyper-distance denoting the spatial dissimilarity between two snapshots $\mathrm{Y}(\boldsymbol{s}_i, t_l)$, $\mathrm{Y}(\boldsymbol{s}_i, t_m)$ with time indexes t_l, t_m, respectively.

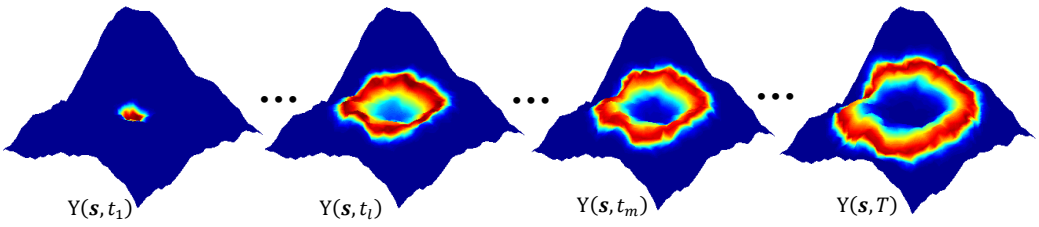


Figure 6.8 Spatiotemporal dynamics on the fractal surface at different time indices.

The hyper-distance matrix D_T quantifies the dissimilarity of spatial data over time. However, it cannot be directly used as statistical quantifiers of spatiotemporal patterns. Therefore, we propose to treat spatial data at a specific time point, e.g., $\mathrm{Y}(\boldsymbol{s}, t_l)$, as nodes in the network and D_T as the edge weight between nodes (i.e., adjacency matrix). Very little work has been done to derive the network structure from the hyper-distance matrix D_T. This is a new means to visualize high-dimensional dynamics in the low-dimensional network. In addition, traditional network

statistics provide new and effective ways to quantify spatiotemporal patterns on the fractal surface. This present investigation is an extension from our previous works in self-organizing topology of recurrence networks and variable clustering [144, 145].

Hence, network structure will be derived from the hyper-distance matrix D_T through the self-organizing process. Let $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$ be the directed and weighted network, where $\boldsymbol{V}$ is the set of nodes and $\boldsymbol{E}$ is the set of edges. We utilized the spring-electrical model to assign two forces (i.e., attractive and repulsive forces) between nodes. The repulsive force is defined as

$$f_r(l,m) = -\frac{1}{\|\boldsymbol{z}_l - \boldsymbol{z}_m\|^2} * e^{\mathrm{D}_T(l,m)} \tag{21}$$

where $\boldsymbol{z}_l$ and $\boldsymbol{z}_m$ are the locations of network nodes l and m. It may be noted that the repulsive force is proportional to the distance between two points. This is because a large repulsive force is expected to separate the two points when they have a large distance. The attractive force is defined as

$$f_a(l,m) = \|\boldsymbol{z}_l - \boldsymbol{z}_m\|^2 * e^{-\mathrm{D}_T(l,m)}, l \leftrightarrow m \tag{22}$$

The attractive force only exists between two connected nodes and is inversely proportional to the distance between them, because a bigger attractive force will pull two nodes closer when they have a smaller distance. The combined force on a node l is the summation of all repulsive forces and attractive forces on the node:

$$f(l,\boldsymbol{z}) = \sum_{l \neq m} -\frac{e^{\mathrm{D}_T(l,m)}}{\|\boldsymbol{z}_l - \boldsymbol{z}_m\|^3}(\boldsymbol{z}_l - \boldsymbol{z}_m) + \sum_{l \leftrightarrow m} \|\boldsymbol{z}_l - \boldsymbol{z}_m\|(\boldsymbol{z}_l - \boldsymbol{z}_m) * e^{-\mathrm{D}_T(l,m)} \tag{23}$$

where $\overrightarrow{\boldsymbol{z}_l} - \overrightarrow{\boldsymbol{z}_l}$ is the force-directional vector, which is separated from $f_r(l,m)$ and $f_a(l,m)$ to define the direction of combined force $f(l,\boldsymbol{z})$.

The objective of self-organizing process is to optimize spatial locations of network nodes by minimizing the total network energy as: $\mathrm{Min}_{\boldsymbol{z}}\{\sum_{l=1,\ldots,T} f^2(l,\boldsymbol{z})\}$. As a result, the structure of

the network is steady with the minimal energy. This, in turn, derives a unique geometry from the hyper-distance matrix D_T. In other words, the optimization function provides us a unique solution for the low-dimensional embedding of spatiotemporal data $\mathrm{Y}(\boldsymbol{s}, t)$ into a network. Furthermore, network statistics provide new and effective means to quantify spatiotemporal patterns on the fractal surface. For example, the shortest path among nodes is an important measure used in network research. A popular question in social network is "What is the shortest distance among people?" Milgram's six-degrees of separation experiment in 1967 showed that people in the US were on average 6-steps from any other person [146]. Consider the weighted network $\boldsymbol{G}$ formed by the set of vertices $\{\boldsymbol{z}_l\}_{l=1}^{T}$. Let $\mathrm{d}(\boldsymbol{z}_l, \boldsymbol{z}_m)$ denote the shortest path between nodes l and m, which is calculated using the Dijkstra's algorithm [142]. If we assume $\mathrm{d}(\boldsymbol{z}_l, \boldsymbol{z}_m) = 0$ for disconnected nodes l and m, the average path length l_G is

$$l_G = \frac{1}{T(T-1)} \sum\nolimits_{l \neq m} \mathrm{d}(\boldsymbol{z}_l, \boldsymbol{z}_m) \tag{24}$$

This average path length l_G describes the average distance among all nodes in a weighted network. This measure can be used to compare networks with different ranges of weights. Similarly, we define the median path length m_G as

$$m_G = \mathrm{Median}(\{\mathrm{d}(\boldsymbol{z}_l, \boldsymbol{z}_m)\}_{l \neq m}) \tag{25}$$

6.4 Materials and Experimental Design

In this present investigation, we illustrated and evaluated the proposed methodology for simulation modeling of spiral wave dynamics and turbulent pattern on fractal surfaces. First, we studied the differences of reaction-diffusion dynamics on regular and fractal surfaces. Second, we varied the fractal dimension to generate different types of fractal surfaces. Then we investigated how fractal characteristics change spatiotemporal dynamic patterns on complex surfaces. Finally, nonlinear modeling of spiral formation and turbulent patterns was investigated on an anatomically

realistic heart model. We simulated and compared the propagation and conduction of electrical activity between a healthy heart and a heart with arrhythmia. Experimental design is detailed as follows.

6.4.1 Nonlinear Reaction-Diffusion Model

In order to simulate spatiotemporal dynamics, we utilized nonlinear reaction-diffusion model, i.e., Fitzhugh-Nagumo (FHN) model [147], which is widely used to simulate cardiac electrical conduction on 2-dimensional tissues. Eq. (4) is a general formulation of reaction-diffusion model. In this present paper, we specify the functions f and g in FHN kinetic model Eq. (4) as

$$\begin{aligned} f(u,v) &= c_1 u(1-u)(u-a) - c_2 uv \\ g(u,v) &= b(u-dv) \end{aligned} \tag{26}$$

If we substitute Eq. (26) into Eq. (4), the FHN model becomes

$$\begin{aligned} \frac{\partial u}{\partial t} &= c_1 u(1-u)(u-a) - c_2 uv + \nabla^2 u \\ \frac{\partial v}{\partial t} &= b(u-dv) \end{aligned} \tag{27}$$

where $\mathcal{D}_1 = 1$, $\mathcal{D}_2 = 0$, $a = 0.13$, $b = 0.013, c_1 = 0.26, c_2 = 0.1, d = 1.0$ (also see section 6.3.3), u is the membrane voltage, and v is the recovery variable. It may be noted that membrane voltage u is the dynamic variable used to simulate the electrical propagation and conduction on the fractal surfaces.

In addition to differentiating reaction-diffusion dynamics on regular and fractal surfaces, we generate different types of fractal surfaces with a variety of fractal dimensions (i.e., from 2.05 to 2.95, step size 0.05) to study how fractal characteristics change spatiotemporal dynamic patterns on complex surfaces (see Figure 6.9). For each fractal dimension, we generated 10 replicates of

surfaces so as to obtain the statistical distribution of modeling performance. Two stimulation protocols were used to generate spiral waves: (1) $S1$ protocol: A basic stimulation ($S1$) is applied at the center of a 33×33 grid for 20 time steps. (2) $S1-S2$ protocol: After the $S1$ stimulation, a second premature stimulation ($S2$) was applied after 4000 time steps and last for 500 time steps at 3 grids away from the center. These two protocols generate two different types of spiral waves on the fractal surfaces.

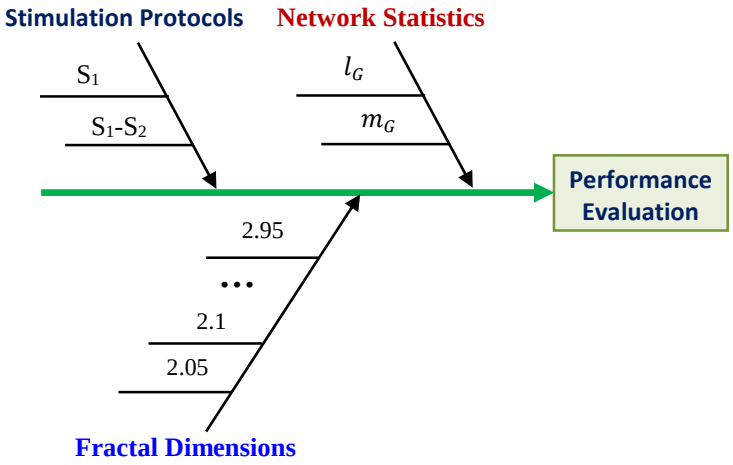


Figure 6.9 Experimental design for reaction-diffusion modeling on fractal surfaces.

6.4.2 Whole Heart Simulation

Furthermore, we studied the nonlinear modeling of spiral formation and turbulent patterns on a whole heart. Note that a healthy heart has near-periodic electrical impulses, while cardiac arrhythmia has rapid, disorganized and irregular electrical impulses. Specifically, this study focuses on the simulation modeling of electrical activities for healthy controls and arrhythmias. Recently, computer simulation of electrical conduction and propagation on a human heart is receiving increasing attentions, because it overcomes many practical and ethical limitations in real-world biomedical experiments. In addition, computer simulation offers greater flexibility for biomedical scientists to test their hypothesis and develop new hypotheses for cardiovascular research and knowledge discovery.

Cardiac electrical activity is a series of complex biochemical-mechanical reactions, which involves orchestrated transportation of large amounts of ions through various biological channels. Electrical activity will also propagate from cell to cell through the entire heart to maintain the vital living organism. In the literature, cellular automata and reaction-diffusion models were widely used to model cardiac electrical propagation and conduction processes [148-151]. A cellular automaton is a discrete model with a regular grid of cells, each has a finite number of states. Every cell has the same updating rule based on its neighboring states. Because of its simplicity and superior computational speed, cellular automata was popular in whole heart simulation. However, simplistic assumptions and rules are limited in their ability to model cardiac electrical activity, especially on the irregular surfaces. Reaction-diffusion models describe how dynamic variables change the distribution in space and time under two processes, i.e., (i) Reaction process: dynamic variables are interacting with each other for conversion, and (ii) Diffusion process: dynamic variables spread out in space. Although reaction-diffusion models provide more realistic simulation of electrical activity through the whole heart, they are more difficult to model on irregular surfaces and more computationally expensive than cellular automata.

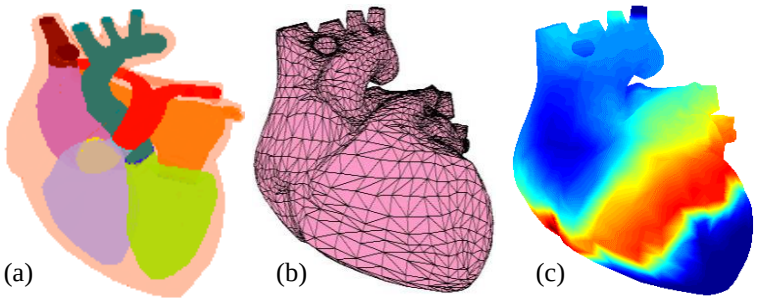


Figure 6.10 Whole heart models. (a) Colored components of the human heart, (b) Finite element meshes, (c) Electrical conduction on the surface of the human heart.

This present investigation provides a practical solution to simulate electrical propagation and conduction through the heart. However, a significant challenges resides in the representation

of complex geometry of the heart. This representation must not only characterize geometric complexities but also yield sufficient resolution to capture activation wavefronts and cell-to-cell propagation dynamics. As shown in Figure 6.10, this present investigation utilized an anatomically realistic heart geometry for reaction-diffusion modeling of cardiac electrical activity. However, traditional finite-difference methods (FDM) are effective on regular surfaces with orthogonal and regular grids by discretizing the diffusion tensor in the domain. The complex geometry and high dimensionality of heart (see Figure 6.10b) pose great challenges for FDMs. Therefore, we proposed reaction-diffusion modeling in the reduced dimension (see section 6.3.3) that is designed and developed for irregular grids of fractal surfaces. This new methodology will be demonstrated and evaluated for simulation modeling of space-time electrical dynamics on the complex geometry of a 3D heart (see Figure 6.10c).

6.5 Experimental Results

6.5.1 Spatiotemporal Simulation on Regular and Fractal Surfaces

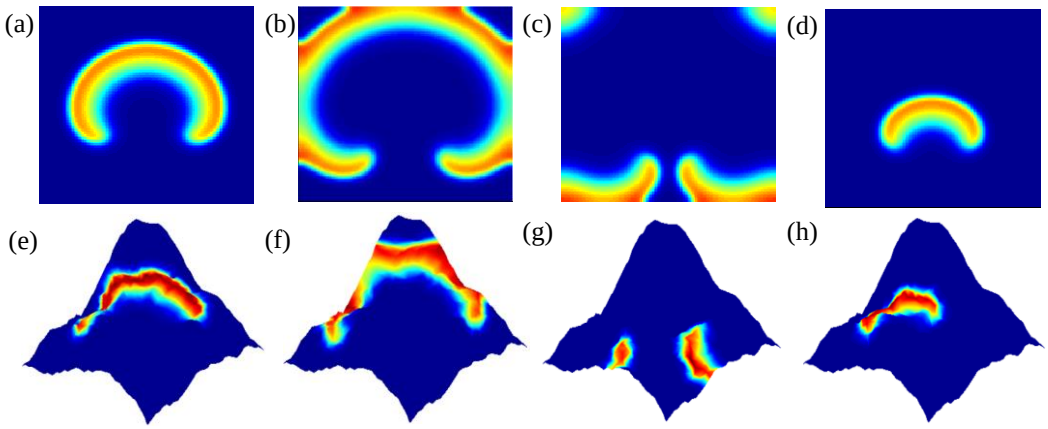


Figure 6.11 Snapshots of reaction-diffusion dynamical patterns at different time indices on the regular surface (a)-(d), and the fractal surface (e)-(h).

As shown in Figure 6.11, we simulate the spiral formation and turbulent patterns on both regular and fractal surfaces. Figure 6.11a-d shows 2-D snapshots at different time indices of space-time dynamic variable u with spiral formation (which rotates over time) on the regular surface. It

may be noted that reaction-diffusion dynamics on the regular surface exhibits the formation of spiral waves and periodic patterns. The reaction-diffusion dynamics start with a semicircular arc amid the surface (see Figure 6.11a), and then diffuses to the edges of the surface (see Figure 6.11b). The membrane voltage vanishes at the top of the surface but re-circulate at the bottom (see Figure 6.11c), and finally reforms a semicircle in the middle (see Figure 6.11d). As a result, spiral waves periodically rotate on the regular surface.

On the other hand, Figure 6.11e-h show reaction-diffusion dynamics on the fractal surface. It may be noted that spatiotemporal patterns in these snapshots are different due to the geometric complexity of the fractal surface. Although there are similar patterns as the regular surface, semicircular arcs show distinct patterns in the middle of the surface (see Figure 6.11h). The width and length of spiral waves are different. After a sufficiently long time period, spiral wave patterns become more chaotic on the fractal surface that are vastly different from periodic patterns on the regular surface.

6.5.2 Spatiotemporal Simulation on the 3-D Heart Surface

Further, we investigated the excitation and propagation of cardiac electrical activity on the 3-D heart surface. Because of the complexity of heart geometry, we leveraged the proposed methodology to address the issues of irregular node spacing, complex boundaries, and regionally dependent conductivities in the 3-D heart model. As shown in Figure 6.12, the propagation and conduction of electrical activity were simulated and compared between a healthy heart (see Figure 6.12a-f) and a heart with arrhythmia (see Figure 6.12g-l). It may be noted that the healthy heart yields regular and periodic electrical impulses. In the healthy heart, electrical activity is near-periodically paced by the sinoatrial node in the right atrium in each cardiac cycle (see Figure 6.12a). Cardiac electric activity propagates throughout the whole heart (see Figure 6.12b-e) and eventually vanishes (see Figure 6.12f), then the next cycle starts.

However, electrical excitation doesn't begin from the sinoatrial node in the simulation of cardiac arrhythmia. Instead, cardiac cycle initiates from another site of the right atrium or muscle cells in the nearby pulmonary veins. As such, the rapid and disorganized electrical impulses occur around atria (see Figure 6.12g-l). As shown in Figure 6.12, electrical conduction is different between the healthy heart and the arrhythmia one. The snapshots in Figure 6.12g-l are taken at the same time indices as Figure 6.12a-f. However, traveling patterns of spatiotemporal electrical activity are different. It may be noted that space-time patterns in the arrhythmia heart are more disorganized than the healthy heart. In addition, electrical waves can rotate and self-circulate in the arrhythmia heart without the second stimulus. As such, this causes the heart to fibrillate (commonly known as atrial fibrillations) (see Figure 6.12l). Next section will further details the pattern recognition of space-time electrical activities in the whole heart between healthy control and cardiac arrhythmia.

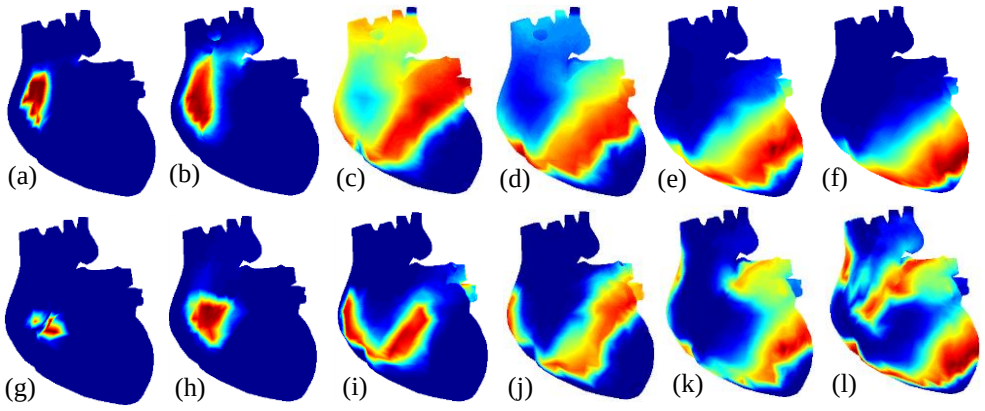


Figure 6.12 Spatiotemporal dynamics of electrical conduction at different time indices on the 3-D heart that is healthy (a)-(f), or is with cardiac arrhythmia (g)-(l).

6.5.3 Spatiotemporal Pattern Recognition

Figure 6.12 shows the frames of electrical conduction on the 3-D heart surface over time. As aforementioned, we utilized a hyper-distance matrix D_T to quantify the frame-to-frame dissimilarity. Each frame, e.g., $Y(\boldsymbol{s}, t_l)$, is treated as nodes in a network and D_T as the edge weight

between nodes (i.e., adjacency matrix). Then, a self-organizing approach is used to derive the network topology by minimizing the energy. Figure 6.13 shows geometric structures of low-dimensional networks derived from spatiotemporal dynamics of the healthy heart (Figure 6.12a-f) and a heart with arrhythmia (Figure 6.12g-l).

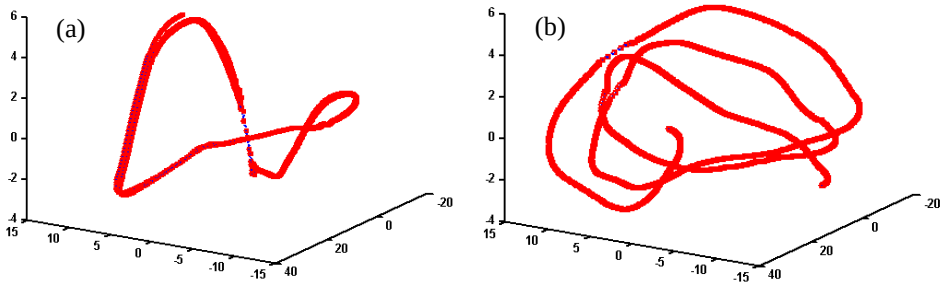


Figure 6.13 Self-organizing topology of low-dimensional networks derived from spatiotemporal electrical conduction and propagation of (a) a healthy heart and (b) a heart with arrhythmia.

As shown in Figure 6.13a, network structure of the healthy heart shows a regular and periodic pattern. However, network topology of the arrhythmia heart is disorganized and irregular (see Figure 6.13b). As such, the proposed approach of self-organizing network provides a new means to visualize spatiotemporal dynamics in low-dimensional networks, thereby facilitating the characterization and quantification of dynamical properties of the underlying complex processes.

In addition, we investigated how fractal characteristics of a surface impact the spatiotemporal patterns. Figure 6.14a shows an example of simulated fractal surfaces with the fractal dimension $D_F = 2.05$. As the fractal dimension is close to dimension 2, the simulated fractal surface is flat and similar to 2D regular surface. We used the S1 stimulation protocol to generate reaction-diffusion dynamics on the fractal surface. Figure 6.14b is the 3-dimensional topology of self-organizing network. Each green node in the network is pertinent to a frame of spatiotemporal images (e.g., Figure 6.14c-f). The movement of nodes along the network trajectory reveals the frame-to-frame differences of spatiotemporal patterns on the fractal surface. The nodes corresponding to Figure 6.14c and Figure 6.14f are very close in the network, because Figure 6.14c

is the basic stimulation (S1) at the beginning and Figure 6.14f is pertinent to the vanish waves at the end. Figure 6.14d and Figure 6.14e yield smooth waves in the simulation because the fractal dimension 2.05 is close to dimension 2 (i.e., a regular 2-D surface).

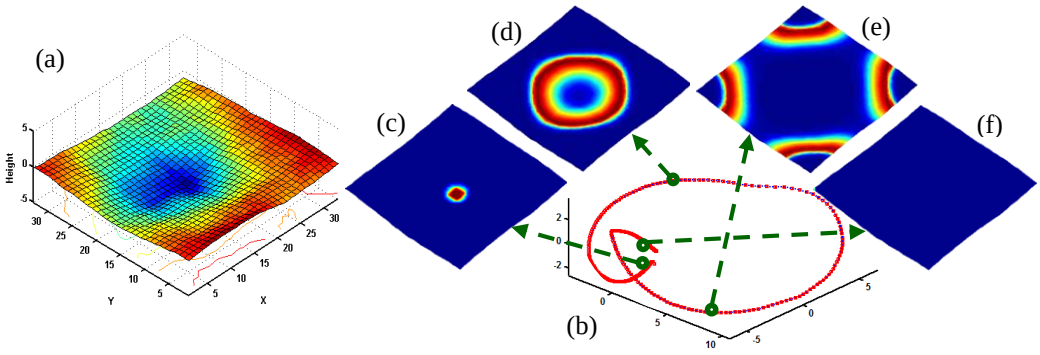


Figure 6.14 (a) A simulated fractal surface with fractal dimension $D_F = 2.05$. (b) 3-dimensional topology of self-organizing network. (c)-(f) Frames of spatiotemporal patterns on the fractal surface (S1 protocol). Note: Each green node in the network is mapped back to its associated frame.

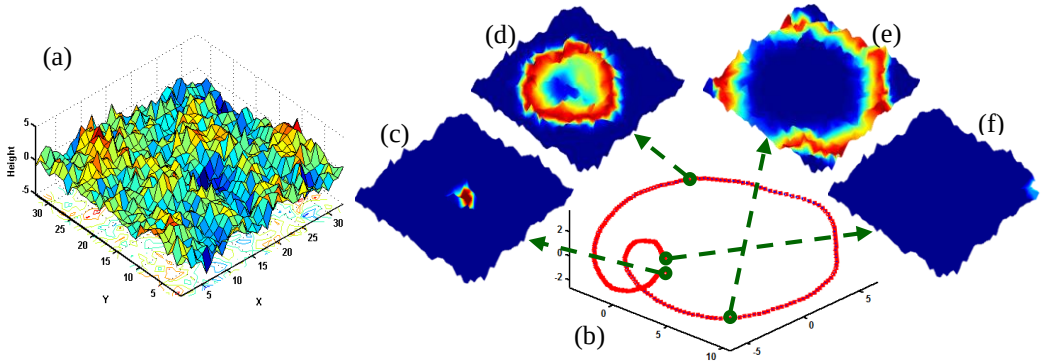


Figure 6.15 (a) A simulated fractal surface with fractal dimension $D_F = 2.95$. (b) 3-dimensional topology of self-organizing network. (c)-(f) Frames of spatiotemporal patterns on the fractal surface (S1 protocol). Note: Each green node in the network is mapped back to its associated frame.

On the other hand, Figure 6.15 shows the results from the surface with fractal dimension $D_F = 2.95$. Because the fractal dimension is away from dimension 2 (i.e., a regular 2-D surface), the simulated fractal surface is uneven and irregular. As a result, spatiotemporal dynamic patterns in Figure 6.15c-f are varied from Figure 6.14c-f, although the same S1 protocol is used. Also, the path of network trajectory in Figure 6.15b is visually different from Figure 6.14b, particular in the

beginning and end of network trajectory. However, Figure 6.14 and Figure 6.15 are only visual representations of spatiotemporal dynamics in the low-dimensional networks. It is necessary to further characterize and quantify dynamical properties of the underlying processes.

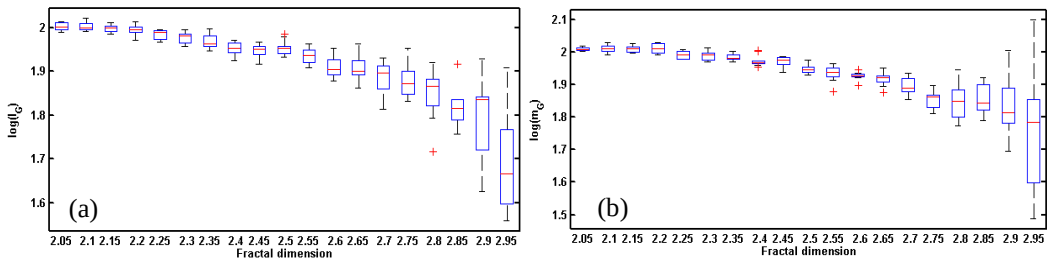


Figure 6.16 Network statistics of spatiotemporal dynamics (S1 protocol) on complex surfaces with various fractal dimensions. (a) Average path length l_G and (b) median path length m_G.

Figure 6.16 shows the distribution of network statistics, i.e., average path length l_G and median path length m_G, with respect to the variations of fractal dimensions. Notably, we generated 10 replicate surfaces for each fractal dimension. We took the logarithm of two network statistics as shown in the y-axis of the plot (see Figure 6.15). Figure 6.16a shows a decreasing trend of the average path length l_G when the fractal dimension is increased from 2.05 to 2.95. As the fractal dimension gets bigger, the variance of the average path length l_G is also increasing. This is because replicate surfaces for a bigger fractal dimension often have large variations. Figure 6.15b shows a similar trend for the variations of the median path length m_G when the fractal dimension is increasing.

In addition to the S1 protocol, we have also investigated reaction-diffusion dynamics with the S1-S2 stimulation protocol, i.e., a premature stimulation (S2) after the basic stimulation (S1). The fractal surfaces used are the same as in Figure 6.14a ($D_F = 2.05$) and Figure 6.15a ($D_F = 2.95$). However, S1-S2 protocol generates spiral waves that are more complex on the surfaces. Note that the premature stimulation (S2) is applied at the location that is 3 grids away from the center (see Figure 6.17b). As such, a semicircular arc appears in the middle of the surface (see

Figure 6.17c). This semicircular arc continues to propagate after the basic stimulation (S1) vanishes at the boundary (see Figure 6.17d). However, the semicircular arc will not completely vanish at the boundary. Instead, it will rotate back to the surface to form continuous or self-circulating "spiral waves" on the surface (see Figure 6.17e-g). It is worth mentioning that the network trajectory (See Figure 6.17a) is vastly different from the one with the S1 protocol in Figure 6.14b.

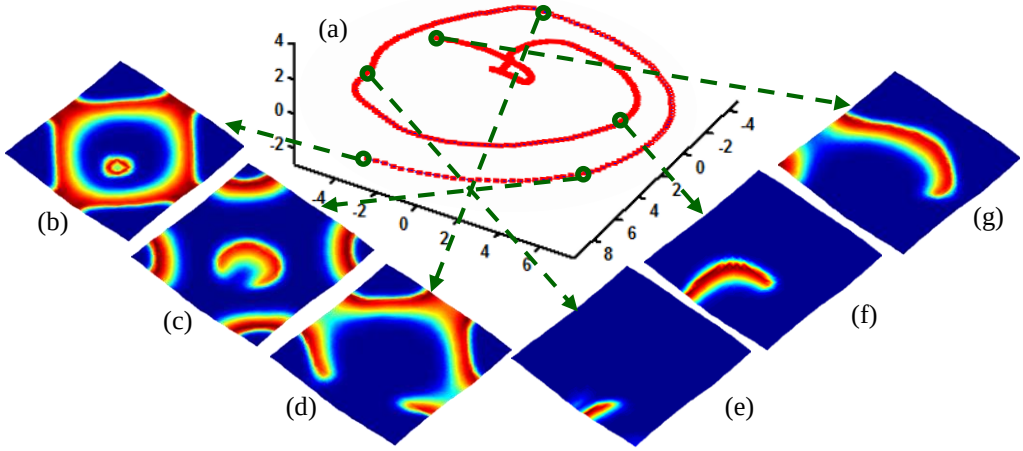


Figure 6.17 (a) 3-dimensional topology of self-organizing network with fractal dimension $D_F = 2.05$ in Figure 6.14a. (b)-(g) Frames of spatiotemporal patterns with the S1-S2 protocol on the surface. Note: Each green node in the network is mapped back to its associated frame.

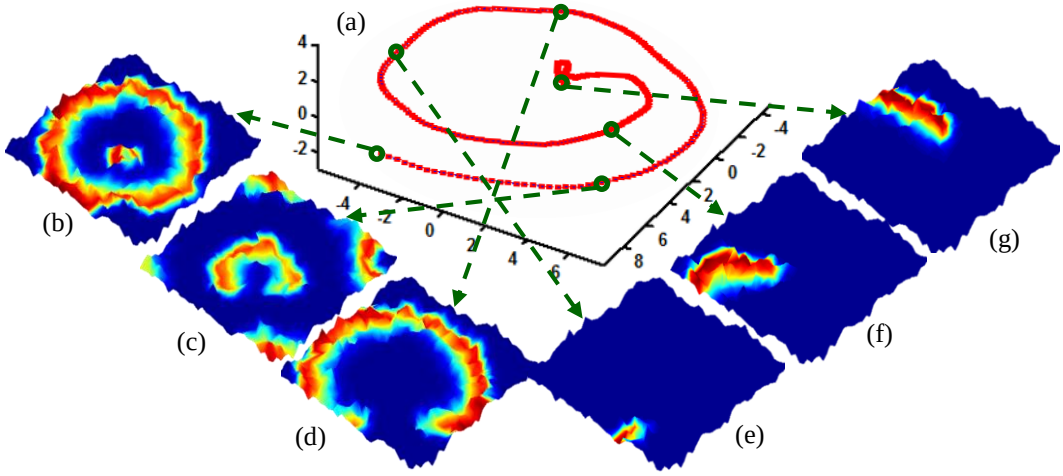


Figure 6.18 (a) 3-dimensional topology of self-organizing network with fractal dimension $D_F = 2.95$ in Figure 6.15a. (b)-(g) Frames of spatiotemporal patterns with the S1-S2 protocol on the surface. Note: Each green node in the network is mapped back to its associated frame.

Figure 6.18 shows simulation results of S1-S2 protocol on the complex surface with fractal dimensions $D_F = 2.95$ (i.e., the same surface in Figure 6.15a). Although Figure 6.18b-g are taken at exactly the same time indices as Figure 6.17b-g, wave patterns are significantly different with the same S1-S2 protocol. Note that the path of network trajectory in Figure 6.18a is varied from Figure 6.17a, particular in the end of network trajectory. Finally, we also calculated the distribution of network statistics for S1-S2 protocol, including average path length l_G and median path length m_G, when the fractal dimension is increased from 2.05 to 2.95. As show in Figure 6.19, both network statistics are decreasing monotonically with the increasing fractal dimension. The variance of network statistics is increasing as the fractal dimension gets bigger. The consistent results in Figure 6.16 and Figure 6.19 show that network statistics such as average path length l_G and median path length m_G effective characterize the complexity of fractal surfaces. As such, the approach of self-organizing network provides an effective tool for pattern recognition of spatiotemporal dynamics on the fractal surfaces.

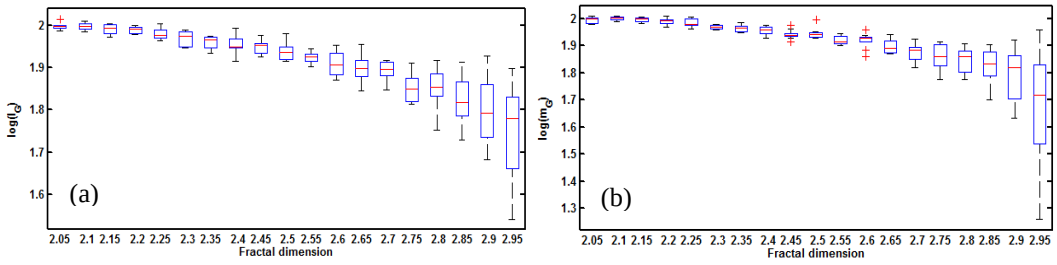


Figure 6.19 Network statistics of spatiotemporal dynamics (S1-S2 protocol) on complex surfaces with various fractal dimensions. (a) Average path length l_G and (b) median path length m_G.

6.6 Discussion and Conclusions

Simulation modeling provides a better understanding of complex phenomena, and thereby enables the prediction of system behaviors and optimal control of system operations. However, many engineered and natural systems involve irregular and self-similar geometric forms (i.e., fractal geometry). Very little has been done to investigate the simulation model of spatiotemporal

dynamics (e.g., cardiac electrical excitation and conduction) on the fractal geometry. Most of previous simulation models were developed for space-time dynamics (e.g., reaction, diffusion and propagation) on the Euclidean geometry, e.g., flat planes and rectangular volumes. This brings inaccurate approximation of real-world dynamics on fractal surfaces, due to the sensitive dependence of nonlinear dynamical systems on initial conditions. There is an urgent need to investigate the modeling differences between fractal and Euclidean geometry. This present paper developed a suite of methods and tools for simulation modeling of spatiotemporal dynamics on fractal surfaces, including fractal surface simulation, isometric graphing for surface characterization, dimensionality reduction for reaction-diffusion modeling and Spatiotemporal pattern recognition.

We illustrated and evaluated the proposed methodology for simulation modeling of spatiotemporal dynamics on fractal surfaces. *First*, we compared the differences of reaction-diffusion dynamics on regular and fractal surfaces. As the geometric complexity of the fractal surface is much higher, spatiotemporal patterns on the fractal surface are vastly different from periodic patterns on the regular surface. *Second*, we generated different types of fractal surfaces to investigate how fractal characteristics change spatiotemporal dynamic patterns on complex surfaces. We leveraged manifold learning to extract low-dimensional basis for describing space-time dynamics on the fractal surface. Each frame is treated as a network node. We utilized a self-organizing approach to derive the network topology from a hyper-distance matrix D_T of the frame-to-frame dissimilarity between nodes (i.e., adjacency matrix). We showed that the distribution of network statistics, i.e., average path length l_G and median path length m_G, yields a decreasing trend when the fractal dimension is increased from 2.05 to 2.95. The decreasing trend is consistent for two different types of spatiotemporal dynamics from S1 and S1-S2 protocols. This experiment

showed that self-organizing network provides an effective tool for pattern recognition of spatiotemporal dynamics on the fractal surfaces. In addition, network statistics such as average path length l_G and median path length m_G effective characterize the complexity of fractal surfaces.

Further, we developed the whole-heart simulation model to investigate the excitation and propagation of cardiac electrical activity between a healthy heart and a heart with arrhythmia. The proposed simulation methodology effectively addresses the complexity of heart geometry such as irregular node spacing, complex boundaries, and regionally dependent conductivities. Experimental results showed that space-time patterns in the arrhythmia heart are more disorganized than the healthy heart. In addition, electrical waves can rotate and self-circulate in the arrhythmia heart without the second stimulus. This research demonstrated that the proposed methodology outperforms traditional modeling approaches based on the Euclidean geometry, and provide effective tools to model and characterize space-time dynamics on fractal surfaces of complex systems.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

Real-time sensing brings the proliferation of big data that contains rich information of complex systems. It is well known that real-world systems show high levels of nonlinear and nonstationary behaviors in the presence of extraneous noise. This brings significant challenges for human experts to visually inspect the integrity and performance of complex systems from the collected data. My research goal is to develop innovative methodologies for modeling and optimizing complex systems, and create enabling technologies for real-world applications. Specifically, my research focuses on *Mining Dynamic Recurrences in Nonlinear and Nonstationary Systems for Feature Extraction, Process Monitoring and Fault Diagnosis*. This research will enable and assist in (i) sensor-driven modeling, monitoring and optimization of complex systems; (ii) integrating product design with system design of nonlinear dynamic processes; and (iii) creating better prediction/diagnostic tools for real-world complex processes.

My *research accomplishments* include the following.

(1) *Feature Extraction and Analysis:*

I proposed a novel multiscale recurrence analysis to not only delineate recurrence dynamics in complex systems, but also resolve the computational issues for the large-scale datasets. It was utilized to identify heart failure subjects from the 24-hour heart rate variability (HRV) time series and control the quality of mobile-phone-based electrocardiogram (ECG) signals.

(2) *Modeling and Prediction:*

I proposed the design of stochastic sensor network to allow a subset of sensors at varying locations within the network to transmit dynamic information intermittently, and a new approach

of sparse particle filtering to model spatiotemporal dynamics of big data in the stochastic sensor network. It may be noted that the proposed algorithm is very general and can be potentially applicable for stochastic sensor networks in a variety of disciplines, e.g., environmental sensor network and battlefield surveillance network.

(3) *Monitoring and Control:*

Process monitoring of dynamic transitions in complex systems is more concerned with aperiodic recurrences and heterogeneous types of recurrence variations. However, traditional recurrence analysis treats all recurrence states homogeneously, thereby failing to delineate heterogeneous recurrence patterns. I developed a new approach of heterogeneous recurrence analysis for complex systems informatics, process monitoring and anomaly detection.

(4) *Simulation and Optimization:*

Another research focuses on fractal-based simulation to study spatiotemporal dynamics on fractal surfaces of high-dimensional complex systems, and further optimize spatiotemporal patterns. This proposed algorithm is applied to study the reaction-diffusion modeling on fractal surfaces and real-world 3D heart surfaces.

The results and experience gained from my previous projects lay a solid foundation for my continued research and innovation in the interdisciplinary field of manufacturing and healthcare. I will focus both on fundamental methodology development on related topics and on solutions to challenging questions rising from cutting edge problems in my *future research*.

(1) *Modeling and Prediction of Spatiotemporal Dynamics:*

A variety of important scientific and data-driven problems involve quantities that vary over space and time. Examples include time-varying brain image and fMRI data, nanowire growth modeling at multiple spatial scales, and temporal movement of climate data. As the dynamics of

complex systems vary across both space and time, spatiotemporal data is generated from dynamic systems, where the dependence of spatial domain on time symbolizes the changes of spatial domain over time. Traditional methodologies characterize and model spatiotemporal data in two ways: (i) spatially-varying time series model, which separates the temporal analysis for each location; (ii) temporally-varying spatial model, which separates spatial analysis for each time point. However, both approaches are conditional methods studying either the space given time or time given space, and are limited in their capabilities to characterize and model space-time correlations. Space-time interactions bring substantial complexity in the scope of modeling and prediction, because of spatial correlation, temporal correlation, as well as how space and time interact. My research objective is to develop methods and algorithms that work well in practice for a wide range of spatiotemporal dynamics as well as various data types. Specifically, this research aims to address three challenging problems in the analysis of spatiotemporal data: (i) learn underlying spatiotemporal dynamics, (ii) delineate and model informative spatiotemporal patterns and (iii) provide optimal prediction for various type of dynamic systems. This research will provide researchers with powerful, principled and highly automatic methods to analyze and optimally predict spatiotemporal dynamics of complex systems.

(2) *Multiscale Analysis of Complex Systems:*

Complex systems consist of many interrelated and interdependent components linked through numerous relationships and interactions. Therefore, complex systems approaches enable researchers to study aspects of the real world for which events and actions have multiple causes and consequences, and where order and structure coexist at different scales of time, space, and organization. Moreover, with the advancement of microprocessors, miniature sensors, wired and wireless digital networks, contemporary monitoring devices collect enormous amount of datasets

exhibited from complex systems. However, data collected from complex systems typically exhibit at least one of the following properties: (i) Nonlinearity: the relationships among components are not additive; (ii) Nonstationarity: the statistical properties of the system's output change with time; (iii) Multiscale variability: systems exhibit varying patterns over a range of scales.

Therefore, complex systems defy understanding based on the traditional reductionist approach, in which one attempts to understand a system's behavior by combining all constituent parts that have been analyzed separately. My research objective is to develop nonlinear dynamic methodologies that incorporate the concept of scale explicitly, so that different behaviors of signals on varying scales will be simultaneously characterized by the same scale-dependent methodology. Specifically, this research aims to (i) develop concepts and methodologies (e.g., cellular automata, scale-free and small-world network topologies, deterministic chaos and self-organization) under the multiscale framework to investigate complex systems; (ii) study complex processes in biology, economics, engineering, and many other fields (e.g., Earth's global climate, human brain, social organization and ecosystem). This research will provide a deeper understanding of complex systems, and facilitate the development of controls and strategies to make systems more efficient.

(3) *Smart Sensor Network for Process Monitoring and Design Optimization:*

Wireless sensor network has emerged as a key technology for monitoring space-time dynamics of complex systems, e.g., environmental sensor network, battlefield surveillance network, and body area sensor network. Sensors are easily networked through wireless links, deployed in large numbers and distributed throughout complex physical systems. Distributed sensing provides an unprecedented opportunity to monitor space-time dynamics of complex systems and to improve the quality and integrity of operation and services. However, sensor failures are not uncommon in traditional sensing systems. In other words, realizing the full

potential of sensor network hinges on the development of novel information-processing algorithms to support the design and exploit the uncertain information. Moreover, traditional sensing systems lack an effective decision-support systems that can make optimal decisions with incomplete information. My research objective is to develop a smart sensor network for process monitoring and design optimization. By "smart", I mean the sensor network (i) has the ability to monitor complex systems with incomplete sensor information, and (ii) will provide customized rehabilitation when systems have ill conditions or even failed. This research will develop innovative methodologies and interventions that add intelligence to smart sensor networks for improving the functionality of complex systems.

In summary, my research is aimed at creating a rigorous body of knowledge for complex systems analysis, monitoring, modeling, optimization and decision making, which has great potentials for both academic intellectual merits and broader societal impact.

REFERENCES

[1] American Heart Association Writing Group, "Heart Disease and Stroke Statistics—2014 Update: A Report from the American Heart Association," *Circulation,* vol. 129, pp. e28-e292, 2014.

[2] H. Yang, S. T. S. Bukkapatnam and R. Komanduri, "Spatio-temporal representation of cardiac vectorcardiogram signals," *BioMedical Engineering Online,* vol. 11, 2012.

[3] Y. Chen and H. Yang, "Multiscale recurrence analysis of long-term nonlinear and nonstationary time series," *Chaos, Solitons & Fractals,* vol. 45, pp. 978-987, 2012.

[4] A. Katok and B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems.* Cambridge Univ. Press., 1995.

[5] Cardiac Arrhythmia Suppression Trial Investigators, "Preliminary Report: Effect of Encainide and Flecainide on Mortality in a Randomized Trial of Arrhythmia Suppression after Myocardial Infarction," *N. Engl. J. Med.,* vol. 321, pp. 406-12, 1989.

[6] J. Q. You and F. Nori, "Atomic physics and quantum optics using superconducting circuits," *Nature,* vol. 474, pp. 589-97, 2011.

[7] J. R. Pratt and A. H. Nayfeh, "Design and Modeling for Chatter Control," *Nonlinear Dynamics,* vol. 19, pp. 49-69, 1999.

[8] R. Roy, T. W. Murphy, T. D. Maier, Z. Gills and E. R. Hunt, "Dynamical control of a chaotic laser: Experimental stabilization of a globally coupled system," *Phys. Rev. Lett.,* vol. 68, pp. 1259-62, 1992.

[9] M. Ishikawa, "Precise fabrication of nanomaterials: A nonlinear dynamics approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 15, pp. 047503, 2005.

[10] R. Tenny, L. S. Tsimring, L. Larson and H. D. I. Abarbanel, "Using Distributed Nonlinear Dynamics for Public Key Encryption," *Phys. Rev. Lett.,* vol. 90, pp. 047903, 2003.

[11] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980, Lecture Notes in Mathematics,*Anonymous Springer-Verlag, pp. 366-81, 1981.

[12] M. B. Kennel, R. Brown and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A,* vol. 45, pp. 3403-3411, 1992.

[13] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A,* vol. 33, pp. 1134-1140, February, 1986.

[14] J. Eckmann, S. O. Kamphorst and D. Ruelle, "Recurrence Plots of Dynamical Systems," *Europhys. Lett.,* vol. 4, pp. 973, 1987.

[15] N. Marwan, R. M. Carmen, M. Thiel and J. Kurths, "Recurrence plots for the analysis of complex systems," *Phys. Rep.,* vol. 438, pp. 237-329, 2007.

[16] R. Sun and Y. Wang, "Predicting termination of atrial fibrillation based on the structure and quantification of the recurrence plot," *Med. Eng. Phys.,* vol. 30, pp. 1105-1111, 2008.

[17] J. P. Zbilut, N. Thomasson and C. L. Webber, "Recurrence quantification analysis as a tool for nonlinear exploration of nonstationary cardiac signals," *Med. Eng. Phys.,* vol. 24, pp. 53-60, 2002.

[18] N. Marwan and J. Kurths, "Nonlinear analysis of bivariate data with cross recurrence plots," *Phys. Lett. A,* vol. 302, pp. 299-307, 2002.

[19] N. Thomasson, T. J. Hoeppner, C. L. Webber Jr. and J. P. Zbilut, "Recurrence quantification in epileptic EEGs," *Phys. Lett. A,* vol. 279, pp. 94-101, 2001.

[20] Z. Wu, "Recurrence plot analysis of DNA sequences," *Phys. Lett. A,* vol. 332, pp. 250-255, 2004.

[21] F. Strozzi, J. Zaldivar and J. P. Zbilut, "Recurrence quantification analysis and state space divergence reconstruction for financial time series analysis," *Physica A,* vol. 376, pp. 487-499, 2007.

[22] H. Yang, S. T. S. Bukkapatnam and L. G. Barajas, "Local recurrence based performance prediction and prognostics in the nonlinear and nonstationary systems," *Pattern Recognit.,* vol. 44, pp. 1834-40, 2011.

[23] M. Siek and D. P. Solomatine, "Nonlinear chaotic model for predicting storm surges," *Nonlin. Processes Geophys.,* vol. 17, pp. 405-420, 2010.

[24] M. A. Riley and S. Clark, "Recurrence analysis of human postural sway during the sensory organization test," *Neurosci. Lett.,* vol. 342, pp. 45-48, 2003.

[25] W. C. Skamarock and J. B. Klemp, "A time-split nonhydrostatic atmospheric model for weather research and forecasting applications," *Journal of Computational Physics,* vol. 227, pp. 3465-3485, 3/20, 2008.

[26] N. S. Xu and S. E. Huq, "Novel cold cathode materials and applications," *Materials Science and Engineering: R: Reports,* vol. 48, pp. 47-189, 1/31, 2005.

[27] J.-P. Leduc, "Spatio-temporal wavelet transforms for digital signal analysis," *Signal Processing,* vol. 60, pp. 23-41, 1997.

[28] H. Yang, S. T. S. Bukkapatnam and R. Komanduri, "Nonlinear adaptive wavelet analysis of electrocardiogram signals," *Phys. Rev. E,* vol. 76, pp. 026214, 2007.

[29] D. B. Precival and A. T. Walden, *Wavelet Methods for Time Series Analysis.* Cambridge University Press, 2000.

[30] C. L. Webber Jr. and J. P. Zbilut, "Dynamical assessment of physiological systems and states using recurrence plot strategies," *J. Appl. Physiol.,* vol. 76, pp. 965-973, 1994.

[31] J. P. Zbilut and C. L. Webber Jr., "Embeddings and delays as derived from quantification of recurrence plots," *Phys. Rev. A,* vol. 171, pp. 199-203, 12/7, 1992.

[32] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan and J. Kurths, "Recurrence plot based measures of complexity and their application to heart-rate-variability data," *Phys. Rev. E,* vol. 66, pp. 026702, 2002.

[33] Y. Hirata, S. Horai and K. Aihara, "Reproduction of distance matrices and original time series from recurrence plots and their applications," *The European Physical Journal Special Topics,* vol. 164, pp. 13-22, 2008.

[34] G. McGuire, N. B. Azar and M. Shelhamer, "Recurrence matrices and the preservation of dynamical properties," *Physics Letters A,* vol. 237, pp. 43-47, 1997.

[35] J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika,* vol. 53, pp. 325-338, 1966.

[36] M. Thiel, M. C. Romano and J. Kurths, "How much information is contained in a recurrence plot?" *Physics Letters A,* vol. 330, pp. 343-349, 2004.

[37] R. A. Horn and I. Olkin, "When does A*A=B*B and why does one want to know?" *Amer. Math. Monthly,* vol. 103, pp. 470-482, 1996.

[38] M. Thiel, M. C. Romano, J. Kurths, R. Meucci, E. Allaria and F. T. Arecchi, "Influence of observational noise on the recurrence quantification analysis," *Physica D,* vol. 171, pp. 138-152, 10/15, 2002.

[39] K. Q. Shen, C. J. Ong, X. P. Li, Z. Hui and E. P. V. Wilder-Smith, "A Feature Selection Method for Multilevel Mental Fatigue EEG Classification," *Biomedical Engineering, IEEE Transactions on,* vol. 54, pp. 1231-1237, 2007.

[40] A. L. Goldberger, L. Amaral, L. Glass, J. Haussdorff, P. C. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng and H. E. Stanley, "PhysioBank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation,* vol. 23, pp. e215-e220, June. 13, 2000.

[41] P. C. Ivanov, L. A. N. Amaral, A. L. Goldberger, S. Havlin, M. G. Rosenblum, Z. R. Struzik and H. E. Stanley, "Multifractality in human heartbeat dynamics," *Nature,* vol. 399, pp. 461-465, 1999.

[42] S. Schinkel, O. Dimigen and N. Marwan, "Selection of recurrence threshold for signal detection," *Eur. Phys. J. Special Topics,* vol. 164, pp. 45-53, 2008.

[43] H. Yang, "Multiscale recurrence quantification analysis of spatial cardiac vectorcardiogram (VCG) signals," *IEEE Trans. Biomed. Eng.,* vol. 58, pp. 339-347, February, 2011.

[44] J. J. Chen, C. A. Tsai, J. F. Young and R. L. Kodell, "Classification ensembles for unbalanced class sizes in predictive toxicology," *SAR and QSAR in Environmental Research,* vol. 16, pp. 517-529, 2005.

[45] G. D. Clifford, F. Azuaje and P. E. McSharry, *Advanced Methods and Tools for ECG Data Analysis.* London: Artech House, 2006.

[46] J. N. F. Mak, Y. Hu and K. D. K. Luk, "An automated ECG-artifact removal method for trunk muscle surface EMG recordings," *Med. Eng. Phys.,* vol. 32, pp. 840-848, 10, 2010.

[47] H. Park, D. Jeong and K. Park, "Automated detection and elimination of periodic ECG artifacts in EEG using the energy interval histogram method," *IEEE Trans. Biomed. Eng.,* vol. 49, pp. 1526-1533, 2002.

[48] I. Silva, G. B. Moody and L. Celi, "Improving the quality of ECGs collected using mobile phones: the PhysioNet/Computing in cardiology challenge 2011," *Comput. Cardiol.,* vol. 38, pp. 273-276, 2011.

[49] D. Hayn, B. Jammerbund and G. Schreier, "ECG quality assessment for patient empowerment in mHealth applications," in *Computing in Cardiology, 2011*, pp. 353-356, 2011.

[50] J. Kuzilek, M. Huptych, V. Chudacek, J. Spilka and L. Lhotska, "Data driven approach to ECG signal quality assessment using multistep SVM classification," in *Computing in Cardiology, 2011*, pp. 453-455, 2011.

[51] S. Zaunseder, R. Huhle and H. Malberg, "CinC challenge — assessing the usability of ECG by ensemble decision trees," in *Computing in Cardiology, 2011*, pp. 277-280, 2011.

[52] L. Johannesen, "Assessment of ECG quality on an android platform," in *Computing in Cardiology, 2011*, pp. 433-436, 2011.

[53] C. Liu, P. Li, L. Zhao, F. Liu and R. Wang, "Real-time signal quality assessment for ECGs collected using mobile phones," in *Computing in Cardiology, 2011*, pp. 357-360, 2011.

[54] V. Chudacek, L. Zach, J. Kuzilek, J. Spilka and L. Lhotska, "Simple scoring system for ECG quality assessment on android platform," in *Computing in Cardiology, 2011*, pp. 449-451, 2011.

[55] H. Xia, G. A. Garcia, J. C. McBride, A. Sullivan, T. De Bock, J. Bains, D. C. Wortham and Xiaopeng Zhao, "Computer algorithms for evaluating the quality of ECGs in real time," in *Computing in Cardiology, 2011*, pp. 369-372, 2011.

[56] I. Jekova, V. Krasteva, I. Dotsinsky, I. Christov and R. Abacherli, "Recognition of diagnostically useful ECG recordings: Alert for corrupted or interchanged leads," in *Computing in Cardiology, 2011*, pp. 429-432, 2011.

[57] N. Kalkstein, Y. Kinar, M. Na'aman, N. Neumark and P. Akiva, "Using machine learning to detect problems in ECG data collection," in *Computing in Cardiology, 2011*, pp. 437-440, 2011.

[58] P. Langley, L. Y. Di Marco, S. King, D. Duncan, C. Di Maria, W. Duan, M. Bojarnejad, Dingchang Zheng, J. Allen and A. Murray, "An algorithm for assessment of quality of ECGs acquired via mobile telephones," in *Computing in Cardiology, 2011*, pp. 281-284, 2011.

[59] A. C. Maan, E. W. van Zwet, S. Man, S. M. M. Oliveira-Martens, M. J. Schalij and C. A. Swenne, "Assessment of signal quality and electrode placement in ECGs using a reconstruction matrix," in *Computing in Cardiology, 2011*, pp. 289-292, 2011.

[60] B. E. Moody, "Rule-based methods for ECG quality control," in *Computing in Cardiology, 2011*, pp. 361-363, 2011.

[61] K. Noponen, M. Karsikas, S. Tiinanen, J. Kortelainen, H. Huikuri and T. Seppanen, "Electrocardiogram quality classification based on robust best subsets linear prediction error," in *Computing in Cardiology, 2011*, pp. 365-368, 2011.

[62] V. Starc, "Could determination of equivalent dipoles from 12 lead ECG help in detection of misplaced electrodes," in *Computing in Cardiology, 2011*, pp. 445-448, 2011.

[63] T. H. C. Tat, Chen Xiang and Lim Eng Thiam, "Physionet challenge 2011: Improving the quality of electrocardiography data collected using real time QRS-complex and T-wave detection," in *Computing in Cardiology, 2011*, pp. 441-444, 2011.

[64] R. B. Govindan, K. Narayanan and M. S. Gopinathan, "On the evidence of deterministic chaos in ECG: Surrogate and predictability analysis," *Chaos*, vol. 8, pp. 495-502, 1998.

[65] H. Kantz and T. Schreiber, "Human ECG: nonlinear deterministic versus stochastic aspects," *Science, Measurement and Technology, IEE Proceedings* -, vol. 145, pp. 279-284, 1998.

[66] B. Pilgram and D. T. Kaplan, "Nonstationarity and 1/f noise characteristics in heart rate," *American Journal of Physiology - Regulatory, Integrative and Comparative Physiology*, vol. 276, pp. R1-R9, January 01, 1999.

[67] P. S. Addison, "Wavelet Transforms and the ECG: A Review," *Physiol. Meas.*, vol. 26, pp. 155-199, 2005.

[68] G. E. Dower, A. Yakush, S. B. Nazzal, R. V. Jutzy and C. E. Ruiz, "Deriving the 12-lead electrocardiogram from four (EASI) electrodes," *J. Electrocardiol.*, vol. 21, pp. S182-7, 1988.

[69] D. Dawson, H. Yang, M. Malshe, S. T. S. Bukkapatnam, B. Benjamin and R. Komanduri, "Linear affine transformations between 3-lead (Frank XYZ leads) vectorcardiogram and 12-lead electrocardiogram signals," *J. Electrocardiol.*, vol. 42, pp. 622-30, 12, 2009.

[70] H. Yang, M. Malshe, S. T. S. Bukkapatnam and R. Komanduri, "Recurrence quantification analysis and principal components in the detection of myocardial infarction from vectorcardiogram signals," in *Proceedings of the 3rd INFORMS Workshop on Data Mining and Health Informatics (DM-HI 2008)*, Washington, DC, USA, pp. 1-6, 2008.

[71] L. L. Trulla, A. Giuliani, J. P. Zbilut and C. L. Webber Jr., "Recurrence quantification analysis of the logistic equation with transients," *Phys. Rev. A*, vol. 223, pp. 255-260, December, 1996.

[72] T. Kohonen, *Self-Organizing Maps.* New York: Springer, 1997.

[73] M. Hagenbuchner and A. C. Tsoi, "A supervised training algorithm for self-organizing maps for structures," *Pattern Recog. Lett.*, vol. 26, pp. 1874-1884, 9, 2005.

[74] E. Byon, A. K. Shrivastava and Y. Ding, "A classification procedure for highly imbalanced class sizes," *IIE Transactions*, vol. 42, pp. 288, 2010.

[75] Y. Ding, E. A. Elsayed, S. Kumara, J. C. Lu, F. Niu and J. Shi, "Distributed Sensing for Quality and Productivity Improvements," *IEEE Transactions on Automation Science and Engineering*, vol. 3, pp. 344-359, 2006.

[76] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, pp. 2688-2710, 2010.

[77] Y. Rudy and E. Burnes, "Noninvasive Electrocardiographic Imaging," *Annals of Noninvasive Electrocardiology*, vol. 4, pp. 340-359, 1999.

[78] R. N. Ghanem, P. Jia, C. Ramanathan, K. Ryu, A. Markowitz and Y. Rudy, "Noninvasive electrocardiographic imaging (ECGI): comparison to intraoperative mapping in patients," *Heart Rhythm*, vol. 2, pp. 339-354, 2005.

[79] Y. Rudy, "Cardiac repolarization: Insights from mathematical modeling and electrocardiographic imaging (ECGI)," *Heart Rhythm*, vol. 6, pp. S49-S55, 2009.

[80] Y. Chen and H. Yang, "Self-organized neural network for the quality control of 12-lead ECG signals," *Physiological Measurement*, vol. 33, pp. 1399, 2012.

[81] H. Yang, C. Kan, G. Liu and Y. Chen, "Spatiotemporal differentiation of myocardial infarctions," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, pp. 938-47, 2013.

[82] P. Zarychta, F. E. Smith, S. T. King, A. J. Haigh, A. Klinge, D. Zheng, S. Stevens, J. Allen, A. Okelarin, P. Langley and A. Murray, "Body surface potential mapping for detection of myocardial infarct sites," in *Computers in Cardiology, 2007*, pp. 181-184, 2007.

[83] X. Descombes, F. Kruggel and D. Y. von Cramon, "Spatio-temporal fMRI analysis using Markov random fields," *Medical Imaging, IEEE Transactions on,* vol. 17, pp. 1028-1039, 1998.

[84] S. Das, R. Ghanem and S. Finette, "Polynomial chaos representation of spatio-temporal random fields from experimental measurements," *Journal of Computational Physics,* vol. 228, pp. 8726-8751, 12/10, 2009.

[85] M. W. Woolrich, M. Jenkinson, J. M. Brady and S. M. Smith, "Fully Bayesian spatio-temporal modeling of FMRI data," *Medical Imaging, IEEE Transactions on,* vol. 23, pp. 213-231, 2004.

[86] F. D. Bowman, "Spatiotemporal Models for Region of Interest Analyses of Functional Neuroimaging Data," *Journal of the American Statistical Association,* vol. 102, pp. 442-453, 06/01; 2014/03, 2007.

[87] Q. Huang, "Physics-driven Bayesian hierarchical modeling of the nanowire growth process at each scale," *IIE Transactions,* vol. 43, pp. 1-11, 2011.

[88] J. R. Stroud, P. Muller and B. Sanso, "Dynamic Models for Spatiotemporal Data," *J. R. Statist. Soc. B,* vol. 63, pp. 673-689, 2001.

[89] C. K. Wikle, R. F. Milliff, D. Nychka and L. M. Berliner, "Spatiotemporal Hierarchical Bayesian Modeling: Tropical Ocean Surface Winds," *Journal of the American Statistical Association,* vol. 96, pp. 382-397, Jun., 2001.

[90] L. A. Waller, B. P. Carlin, H. Xia and A. E. Gelfand, "Hierarchical Spatio-Temporal Mapping of Disease Rates," *Journal of the American Statistical Association,* vol. 92, pp. 607-617, Jun., 1997.

[91] J. Kelsall and J. Wakefield, "Modeling Spatial Variation in Disease Risk: A Geostatistical Approach," *Journal of the American Statistical Association,* vol. 97, pp. 692-701, Sep., 2002.

[92] N. Serban, "A space–time varying coefficient model: The equity of service accessibility," *The Annals of Applied Statistics,* vol. 5, pp. 1699-2264, 2011.

[93] J. Mateu, F. Montes and M. Plaza, "The 1970 US draft lottery revisited: a spatial analysis," *Journal of the Royal Statistical Society: Series C (Applied Statistics),* vol. 53, pp. 219-229, 2004.

[94] U. Feige, V. Mirrokni and J. Vondrak, "Maximizing Non-monotone Submodular Functions," *SIAM J. Comput.,* vol. 40, pp. 1133-1153, 2007.

[95] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *Neural Networks, IEEE Transactions on,* vol. 5, pp. 989-993, 1994.

[96] A. Krause, A. Singh and C. Gauestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *The Journal of Machine Learning Research,* vol. 9, pp. 235-284, 2008.

[97] M. Scholz, "Validation of Nonlinear PCA," *Neural Processing Letters,* vol. 36, pp. 21-30, 2012.

[98] Z. Kong, O. Beyca, S. T. Bukkapatnam and R. Komanduri, "Nonlinear Sequential Bayesian Analysis-Based Decision Making for End-Point Detection of Chemical Mechanical Planarization (CMP) Processes," *Semiconductor Manufacturing, IEEE Transactions on,* vol. 24, pp. 523-532, 2011.

[99] Z. Kong, A. Oztekin, O. F. Beyca, U. Phatak, S. T. S. Bukkapatnam and R. Komanduri, "Process Performance Prediction for Chemical Mechanical Planarization (CMP) by Integration of Nonlinear Bayesian Analysis and Statistical Modeling," *Semiconductor Manufacturing, IEEE Transactions on,* vol. 23, pp. 316-327, 2010.

[100] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo in Practice,* A. Doucet and N. D. Freitas, Eds. New York: Springer-Verlag, 2001, pp. 197-223.

[101] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos and Z. Ghahramani, "Kronecker graphs: an approach to modeling networks," *Journal of Machine Learning Research,* vol. 11, pp. 985-1042, 2010.

[102] P. Rao, S. T. S. Bukkapatnam, O. Beyca, Z. Kong and R. Komanduri, "Real-time Identification of Incipient Surface Morphology Variations in Ultra-Precision Machining Process," *J. Manuf. Sci. Eng.,* vol. 136, pp. 021008, 2014.

[103] S. T. S. Bukkapatnam, A. Lakhtakia and S. Kumara, "Analysis of sensor signals shows turning on a lathe exhibits low-dimensional chaos," *Physical Review E,* vol. 52, pp. 2375-87, 1995.

[104] S. T. S. Bukkapatnam, S. Kumara and A. Lakhtakia, "Analysis of acoustic emission signals in machining," *ASME Transactions, Journal of Manufacturing Science and Engineering,* vol. 121, pp. 568-73, 1999.

[105] S. T. S. Bukkapatnam, S. Kumara and A. Lakhtakia, "Real-time chatter control in machining using chaos theory," *CIRP Journal of Manufacturing Systems,* vol. 29, pp. 321-6, 1999.

[106] M. Gültekin, E. A. Elsayed, J. R. English and A. S. Hauksdóttir, "Monitoring automatically controlled processes using statistical control charts," *Int. J. Prod. Res.,* vol. 40, pp. 2303-20, 2002.

[107] G. Singer and I. Ben-Gal, "The funnel experiment: The Markov-based SPC approach," *Qual. Reliab. Eng. Int.,* vol. 23, pp. 899-913, 2007.

[108] N. Marwan, S. Schinkel and J. Kurths, "Recurrence plots 25 years later - Gaining confidence in dynamical transitions," *Europhys. Lett.,* vol. 101, pp. 20007, 2013.

[109] N. Ruschin-Rimini, I. Ben-Gal and O. Maimon, "Fractal geometry statistical process control for non-linear pattern-based processes," *IIE Transactions,* vol. 45, pp. 355-373, 2013.

[110] X. Zhang, H. Wang and Q. Huang, "Statistical and experimental analysis of correlated time-varying process variables for conditions diagnosis in chemical–mechanical planarization," *Semiconductor Manufacturing, IEEE Transactions on,* vol. 22, pp. 512-21, 2009.

[111] H. Wang, X. Zhang, A. Kumar and Q. Huang, "Nonlinear dynamics modeling of correlated functional process variables for condition monitoring in chemical–mechanical planarization," *Semiconductor Manufacturing, IEEE Transactions on,* vol. 22, pp. 188-95, 2009.

[112] X. Zhang and Q. Huang, "Analysis of interaction structure among multiple functional process variables for process control in semiconductor manufacturing," *Semiconductor Manufacturing, IEEE Transactions on,* vol. 23, pp. 263-72, 2010.

[113] H. Yang, S. T. S. Bukkapatnam and L. G. Barajas, "Continuous flow modelling of multistage assembly line system dynamics," *Int. J. Comput. Integr. Manuf.,* vol. 26, pp. 401-11, 2013.

[114] H. Yang and Y. Chen, "Heterogeneous recurrence monitoring and control of nonlinear stochastic processes," *Chaos,* vol. 24, pp. 013138, 2014.

[115] Y. J. Kim and J. M. Petal, "Performance Comparison of the R*-Tree and the Quadtree for kNN and Distance Join Queries," *IEEE Trans. Knowl. Data Eng.,* vol. 22, pp. 1014-27, 2010.

[116] C. H. Weiß, "Visual analysis of categorical time series," *Statistical Methodology,* vol. 5, pp. 56-71, 2008.

[117] C. H. Weiß and R. Göb, "Discovering patterns in categorical time series using IFS," *Comput. Stat. Data Anal.,* vol. 52, pp. 4369-4379, 2008.

[118] C. Cheng, Z. Wang , W. Huang, S. T. S. Bukkapatnam and R. Komanduri, "Ultra-Precision Machining Process Dynamics and Surface Quality Monitoring," *Procedia Manufacturing,* pp. 1-12, 2015.

[119] D. Dornfeld, S. Min and Y. Takeuchi, "Recent Advances in Mechanical Micromachining," *CIRP Ann. Manuf. Technol.,* vol. 55, pp. 745-768, 2006.

[120] R. V. Donner, Y. Zou, J. F. Donges, N. Marwan and J. Kurths, "Recurrence networks—a novel paradigm for nonlinear time series analysis," *New J. Phys.*, vol. 12, pp. 033025, 2010.

[121] J. Zhang and M. Small, "Complex Network from Pseudoperiodic Time Series: Topology versus Dynamics," *Phys. Rev. Lett.*, vol. 96, pp. 238701, Jun, 2006.

[122] J. B. Gao, "Recurrence Time Statistics for Chaotic Systems and Their Applications," *Phys. Rev. Lett.*, vol. 83, pp. 3178-3181, Oct, 1999.

[123] T. B. Tenenbaum, V. de Silva and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, 2000.

[124] Y. Chen and H. Yang, "Sparse Modeling and Recursive Prediction of Space-time Dynamics in Stochastic Sensor Network," *IEEE Transactions on Automation Science and Engineering*, vol. 13, pp. 215-226, 2016.

[125] Y. Chen, G. Liu and H. Yang, "Sparse particle filtering for modeling space-time dynamics in distributed sensor networks," in *Proceedings of 2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Taipei, Taiwan, pp. 626-631, 2014.

[126] B. B. Mandelbrot, *The Fractal Geometry of Nature*. New York: Freeman, 1982.

[127] C. K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley and A. L. Goldberger, "Mosaic organization of DNA nucleotides," *Phys. Rev. E.*, vol. 49, pp. 1685-9, 1994.

[128] J. M. Haussdorff, Y. Ashkenazy, C.-K. Peng, P. C. Ivanov, H. E. Stanley and A. L. Goldberger, "When human walking becomes random walking: fractal analysis and modeling of gait rhythm fluctuations," *Physica A: Statistical Mechanics and its Applications*, vol. 302, pp. 138-47, 2001.

[129] W. Dierking, "Quantitative roughness characterization of geological surfaces and implications for radar signature analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, pp. 2397-2412, 1999.

[130] X. Yang, X. Ning and J. Wang, "Multifractal analysis of human synchronous 12-lead ECG signals using multiple scale factors," *Physica A: Statistical Mechanics and its Applications*, vol. 384, pp. 413-22, 2007.

[131] A. Turiel and C. J. Perez-Vicente, "Multifractal geometry in stock market time series," *Physica A: Statistical Mechanics and its Applications*, vol. 322, pp. 629-49, 2003.

[132] S. Blacher, F. Brouers and G. Ananthakrishna, "Multifractal analysis, a method to investigate the morphology of materials," *Physica A: Statistical Mechanics and its Applications*, vol. 185, pp. 28-34, 1992.

[133] N. C. Kenkel and D. J. Walker, "Fractals in the biological sciences," *Coenoses*, vol. 11, pp. 77-100, 1996.

[134] N. S. Lam, H. Qiu, D. A. Quattrochi and C. W. Emerson, "An evaluation of fractal methods for characterizing image complexity," *Cartography and Geograpghyic Information Science,* vol. 29, pp. 25-35, 2002.

[135] A. Fournier, D. Fussel and L. Carpenter, "Computer rendering of stochastic models," *Communications of the ACM,* vol. 25, pp. 371-384, June, 1982.

[136] D. Saupe, "Algorithms for random fractals," in *The Science of Fractal Images*, H. O. Peitgen and D. Saupe, Eds. New York: Spriger-Verlag, pp. 71-136, 1988.

[137] D. C. Lin and R. L. Hughson, "Modeling Heart Rate Variability in Healthy Humans: A Turbulence Analogy," *Phys Rev Lett,* vol. 86, pp. 1650-1653, 2001.

[138] D. C. Lin and R. L. Hughson, "A phenomenology model of normal sinus rhythm in healthy humans," *Biomedical Engineering, IEEE Transactions on,* vol. 49, pp. 97 - 109, 2002.

[139] I. T. Jolliffe, *Principal Component Analysis.* New York: Springer-Verlag, 1989.

[140] T. Cox and M. Cox, *Multidimensional Scaling.* London: Chapman & Hall, 1994.

[141] H. Yang, Y. Chen and F. Leonelli, "Characterization and monitoring of nonlinear dynamics and chaos in complex physiological systems," in *Healthcare Analytics: From Data to Knowledge to Healthcare Improvement*, H. Yang and E. K. Lee, Eds. New York: Wiely, 2016.

[142] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.

[143] J. M. Rogers and A. D. McCulloch, "A collocation-Galerkin finite element model of cardiac action potential propagation," *Biomedical Engineering, IEEE Transactions on,* vol. 41, pp. 743-757, 1994.

[144] H. Yang and G. Liu, "Self-organized topology of recurrence-based complex networks," *Chaos,* vol. 23, pp. 043116, 2013.

[145] G. Liu and H. Yang, "A self-organizing method for predictive modeling with highly-redundant variables," in *2015 IEEE International Conference on Automation Science and Engineering (CASE),* Gothenburg, Sweden, 2015, pp. 1084-1089.

[146] S. Milgram, "The Small World Problem," *Psychology Today,* vol. 2, pp. 60-67, 1967.

[147] A. M. Pertsov, J. M. Davidenko, R. Salomonsz, W. T. Baxter and J. Jalife, "Spiral waves of excitation underlie reentrant activity in isolated cardiac muscle," *Circulation Research,* vol. 72, pp. 632-650, 1993.

[148] A. J. Pullan, M. L. Buist and L. K. Cheng, *Mathematically Modelling the Electrical Activity of the Heart: From Cell to Body Surface and Back again.* Singapore: World Science, 2005.

[149] F. B. Sachse, *Computational Cardiology: Modeling of Anatomy, Electrophysiology, and Mechanics*. Berlin, Germany: Springer, 2004.

[150] D. Yu, D. Du, H. Yang and Y. Tu, "Parallel computing simulation of electrical excitation and conduction in the 3D human heart," in *Proceedings of Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, Chicago, IL, 2014, pp. 4315-9.

[151] Y. Zhang and C. Bajaj, "Adaptive and quality quadrilateral/hexahedral meshing from volumetric data," *Comput Methods Appl Mech Eng.*, vol. 195, pp. 942-960, 200

[152] Y. Chen and H. Yang, "Heterogeneous recurrence T^2 charts for monitoring and control of nonlinear dynamic processes," *Proceedings of 2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Gothenburg, Sweden, p. 1066-1071, 2015

APPENDIX A: COPYRIGHT PERMISSIONS

Below is permission for the use of material in Chapter 2.

This is a License Agreement between Yun Chen ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

Supplier	Elsevier Limited The Boulevard,Langford Lane Kidlington,Oxford,OX5 1GB,UK
Registered Company Number	1982084
Customer name	Yun Chen
Customer address	14327 Hanging Moss Circle, Apt. 101 TAMPA, FL 33613
License number	3836631390156
License date	Mar 26, 2016
Licensed content publisher	Elsevier
Licensed content publication	Chaos, Solitons & Fractals
Licensed content title	Multiscale recurrence analysis of long-term nonlinear and nonstationary time series
Licensed content author	Yun Chen,Hui Yang
Licensed content date	July 2012
Licensed content volume number	45
Licensed content issue number	7
Number of pages	10
Start Page	978
End Page	987
Type of Use	reuse in a thesis/dissertation
Portion	full article
Format	electronic
Are you the author of this Elsevier article?	Yes
Will you be translating?	No
Title of your thesis/dissertation	Mining Dynamic Recurrences in Nonlinear and Nonstationary Systems for Feature Extraction, Process Monitoring and Fault Diagnosis
Expected completion date	May 2016

Below is permission for the use of material in Chapter 3.

PAPER

Self-organized neural network for the quality control of 12-lead ECG signals

Yun Chen and Hui Yang[1]

Published 17 August 2012 • 2012 Institute of Physics and Engineering in Medicine
Physiological Measurement, Volume 33, Number 9

Physiological Measurement

JOURNAL

ISSN: 0967-3334
Publication year(s): 1993 - present
Publisher: IOP Publishing
Language: English
Country of publication: United Kingdom of Great Britain and Northern Ireland

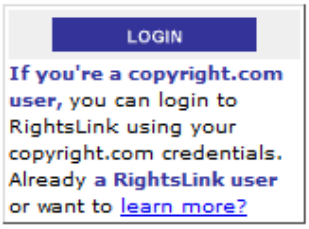
ADDITIONAL AUTHOR RIGHTS AFTER PUBLICATION – SUBSCRIPTION ARTICLES

Final Published Version and Version of Record – Subscription Articles

After the copyright in the article has transferred to IOP Publishing, may I still use the article in a thesis or dissertation?
Yes - upon transfer of copyright, IOP Publishing and/or the copyright owner grants back to authors a number of rights. These include the right to include the Final Published Version of the article in research theses or dissertations. Please include citation details and, for online use, a link to the Version of Record.

Below is permission for the use of material in Chapter 4.

Title: Sparse Modeling and Recursive Prediction of Space–Time Dynamics in Stochastic Sensor Networks
Author: Y. Chen; H. Yang
Publication: Automation Science and Engineering, IEEE Transactions on
Publisher: IEEE
Date: Jan. 2016



Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

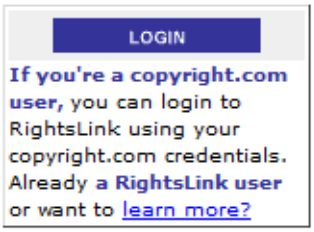
Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Below is permission for the use of material in Chapter 5.

Title:	Heterogeneous recurrence T^2 charts for monitoring and control of nonlinear dynamic processes
Conference Proceedings:	Automation Science and Engineering (CASE), 2015 IEEE International Conference on
Author:	Y. Chen; H. Yang
Publisher:	IEEE
Date:	24-28 Aug. 2015



Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

APPENDIX B: PROOF OF COROLLARY

Corollary: The Laplacian operator of u_i in the domain Ω with the triangular mesh is

$$\nabla^2 u_i = -\sum_{j=1}^{N} u_j \cdot \left(\int_{\Omega} \phi_i(\tilde{\boldsymbol{s}}) d\Omega \right)^{-1} \cdot \left(\int_{\Omega} \nabla\phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla\phi_j(\tilde{\boldsymbol{s}}) d\Omega \right) = \boldsymbol{\nabla}_{u_i}^2 \cdot \boldsymbol{U}$$

where $\boldsymbol{U} = [u_1, u_2, \dots, u_N]^T$.

Proof: If $\Phi, u \in H(\Omega)$, where $H(\Omega) = \left\{ f : \Omega \to \mathbb{R}, \int_{\Omega} f^2 d\Omega < \infty \right\}$ is a function space that all the functions are bounded (i.e., quadratic integrable). The Chain rule from calculus gives that

$$\nabla(\Phi \cdot \nabla u) = \nabla\Phi \cdot \nabla u + \Phi \cdot \nabla^2 u \tag{B.1}$$

Furthermore, using Gauss's theorem on $\nabla(\Phi \cdot \nabla u)$ we have

$$\int_{\Omega} \nabla(\Phi \cdot \nabla u) d\Omega = \int_{\partial\Omega} (\Phi \cdot \nabla u) \cdot \vec{\mathbf{n}} dS \tag{B.2}$$

where $d\Omega = \mathrm{dx} \cdot \mathrm{dy}$ is a surface element in Ω, $\vec{\mathbf{n}}$ is the unit normal direction pointing outward at the boundary $\partial\Omega$ with line element dS. Then, we integrate Eq. (B.1) on both sides and apply Eq. (B.2) to get

$$\int_{\Omega} \nabla(\Phi \cdot \nabla u) d\Omega = \int_{\Omega} \Phi \cdot \nabla^2 u d\Omega + \int_{\Omega} \nabla\Phi \cdot \nabla u d\Omega = \int_{\partial\Omega} (\Phi \cdot \nabla u) \cdot \vec{\mathbf{n}} dS \tag{B.3}$$

Moreover, in our presented research, we assume that the derivatives of concentration variables u and v vanish at the boundary, i.e., $\nabla u = \nabla v = 0$ at $\partial\Omega$. Therefore $\int_{\partial\Omega} (\Phi \cdot \nabla u) \cdot \vec{\mathbf{n}} dS = 0$ and Eq. (B.3) becomes to

$$\int_{\Omega} \Phi \cdot \nabla^2 u d\Omega = -\int_{\Omega} \nabla\Phi \cdot \nabla u d\Omega \tag{B.4}$$

Let $\Phi(\tilde{\boldsymbol{s}}) = \phi_i(\tilde{\boldsymbol{s}})$, then Eq. (B.4) can be rewritten as

$$\int_{\Omega} \phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla^2 u(\tilde{\boldsymbol{s}}) d\Omega = -\int_{\Omega} \nabla\phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla u(\tilde{\boldsymbol{s}}) d\Omega = -\int_{\Omega} \nabla\phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla\left(\sum_{j=1}^{N} u_j \phi_j(\tilde{\boldsymbol{s}})\right) d\Omega$$
$$= -\sum_{j=1}^{N} u_j \cdot \left(\int_{\Omega} \nabla\phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla\phi_j(\tilde{\boldsymbol{s}}) d\Omega\right) \tag{B.5}$$

It may be noted that $\phi_i(\tilde{\boldsymbol{s}}_j) = 0$ for all $j \neq i$. Then, $\nabla^2 u_i$ from Eq. (B.5) becomes

$$\nabla^2 u_i = -\sum_{j=1}^{N} u_j \cdot \left(\int_{\Omega} \phi_i(\tilde{\boldsymbol{s}}) d\Omega\right)^{-1} \cdot \left(\int_{\Omega} \nabla\phi_i(\tilde{\boldsymbol{s}}) \cdot \nabla\phi_j(\tilde{\boldsymbol{s}}) d\Omega\right) = \boldsymbol{\nabla}^2_{u_i} \cdot \boldsymbol{U} \tag{B.6}$$

where $\boldsymbol{U} = [u_1, u_2, \ldots, u_N]^T$.

APPENDIX C: NUMERICAL CALCULATION

Numerical calculation of the *diagonal* lumped mass matrix, $\boldsymbol{M} = [M_{ii}]_{i=1}^{N} = \left[\int_{\Omega} \phi_i d\Omega\right]_{i=1}^{N}$, and the *symmetric* stiffness matrix, $\boldsymbol{K} = \left[K_{ij}\right]_{i,j=1}^{N} = \left[\int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j d\Omega\right]_{i,j=1}^{N}$.

First, the integration on domain Ω is approximated by the sum of integrations on each nonoverlap triangulation Γ_l, i.e., $M_{ii} = \int_{\Omega} \phi_i d\Omega = \sum_{l=1}^{L} \int_{\Gamma_l} \phi_i d\Omega$ and $K_{ij} = \int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j d\Omega = \sum_{l=1}^{L} \int_{\Gamma_l} \nabla\phi_i \cdot \nabla\phi_j d\Omega$.

Second, we illustrate the integration of a triangle Γ_l with vertices $\tilde{\boldsymbol{s}}_i = (x_i, y_i)$, $\tilde{\boldsymbol{s}}_j = (x_j, y_j)$ and $\tilde{\boldsymbol{s}}_k = (x_k, y_k)$ as shown in Figure 6.6(b). First, the basis function $\phi_i(\cdot)$ associated with vertex $\tilde{\boldsymbol{s}}_i$ can be expressed as

$$\phi_i(\tilde{\boldsymbol{s}}) = \frac{\psi_{jk}(\tilde{\boldsymbol{s}})}{\psi_{jk}(\tilde{\boldsymbol{s}}_i)} \tag{C.1}$$

where $\psi_{jk}(\tilde{\boldsymbol{s}}) = (x - x_k)(y_j - y_k) - (y - y_k)(x_j - x_k)$, and the basis functions for vertices $\tilde{\boldsymbol{s}}_j$ and $\tilde{\boldsymbol{s}}_k$ are defined analogously. Therefore, we have elements of two $N \times N$ sparse matrices $\boldsymbol{M}(\Gamma_l) = Sparse\{M_{pp}(\Gamma_l)\}_{p=i,j,k}$ and $\boldsymbol{K}(\Gamma_l) = Sparse\{K_{pq}(\Gamma_l)\}_{p,q=i,j,k}$ as

$$\begin{aligned} M_{pp}(\Gamma_l) &= \int_{\Gamma_l} \phi_p d\Omega = \frac{|\Gamma_l|}{3}, \qquad p = i, j, k \\ K_{pq}(\Gamma_l) &= \int_{\Gamma_l} \nabla\phi_p \cdot \nabla\phi_q d\Omega = \nabla\phi_p \cdot \nabla\phi_q \cdot |\Gamma_l|, \qquad p, q = i, j, k \end{aligned} \tag{C.2}$$

where $|\Gamma_l| = \left|x_j y_k - x_k y_j - x_i y_k + x_k y_i + x_i y_j - x_j y_i\right|/2$ is the area of the triangle Γ_l. And elementary calculations yield

$$\nabla\phi_k \cdot \nabla\phi_i = \frac{(y_j - y_i)(y_k - y_j) - (x_i - x_j)(x_j - x_k)}{\psi_{ji}(\tilde{\boldsymbol{s}}_k)\psi_{kj}(\tilde{\boldsymbol{s}}_i)} \tag{C.3}$$

With similar expressions obtained for $\nabla\phi_k \cdot \nabla\phi_j$, $\nabla\phi_i \cdot \nabla\phi_j$, $|\nabla\phi_i|^2$, $|\nabla\phi_k|^2$ and $\left|\nabla\phi_j\right|^2$.

Finally, the matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ can be obtained by summing matrices through all triangles $\{\Gamma_l\}_{l=1}^{L}$ as $\boldsymbol{M} = \sum_{l=1}^{L} \boldsymbol{M}(\Gamma_l)$ and $\boldsymbol{K} = \sum_{l=1}^{L} \boldsymbol{K}(\Gamma_l)$.

ABOUT THE AUTHOR

Yun Chen was born in Wuxi, Jiangsu Province, China. He received the B.S. degree in Applied Mathematics from the University of Science and Technology of China, Hefei, Anhui, China. He entered the Ph.D. program in Industrial and Management Systems Engineering at University of South Florida, USA, 2010.

His areas of expertise are feature extraction and analysis, process monitoring and control, modeling and prediction, simulation and optimization, big data analytics with applications in healthcare and manufacturing systems. He has published 6 journal articles and has 2 more journal articles under review. His research received the Best Paper Award at *IIE Annual Conference 2015* (Computer and Information Systems Division), and Best Student Paper Finalists at *IIE Annual Conference 2015* (Process Industries Division) and *IEEE CASE Conference 2014*. One of his published papers was also selected as the cover article by *Physiological Measurement* journal.