Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations | Graduate School

2012

PREDICTION OF RESPIRATORY MOTION

Suk Jin Lee
Virginia Commonwealth University

Follow this and additional works at: https://scholarscompass.vcu.edu/etd

 Part of the Engineering Commons

© The Author

Downloaded from
https://scholarscompass.vcu.edu/etd/336

This Dissertation is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

www.manaraa.com

School of Engineering
Virginia Commonwealth University

This is to certify that the dissertation prepared by Suk Jin Lee entitled PREDICTION OF RESPIRATORY MOTION has been approved by his committee as satisfactory completion of the dissertation requirement for the degree of Doctor of Philosophy in Engineering

---

Yuichi Motai, Ph.D., Dissertation Director, School of Engineering

---

Martin J. Murphy, Ph.D., Committee Member, School of Medicine

---

Ashok Iyer, Ph.D., Committee Member, School of Engineering

Alen Docef, Ph.D., Committee Member, School of Engineering

Hongsik Choi, Ph.D., Committee Member, Information Technology, Georgia Gwinnett College

---

Supriyo Bandyopadhyay, Ph.D., Graduate Program Coordinator, School of Engineering

---

Rosalyn Hobson, Ph.D., Associate Dean for Graduate Studies, School of Engineering

---

J. Charles Jennett, Ph.D., Dean, School of Engineering

F. Douglas Boudinot, Ph.D., Dean, School of Graduate Studies

Date

www.manaraa.com

© Suk Jin Lee 2012
All Rights Reserved

www.manaraa.com

PREDICTION OF RESPIRATORY MOTION

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in School of Engineering at Virginia Commonwealth University.

by

Suk Jin Lee

Director: Yuichi Motai, Ph.D.

Department of Electrical and Computer Engineering
School of Engineering
Virginia Commonwealth University
Richmond, Virginia
March 2012

www.manaraa.com

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 REVIEW: PREDICTION OF RESPIRATORY MOTION 5

2.1 TOOLS FOR MEASURING TARGET POSITION DURING RADIOTHERAPY 6
2.1.1 RADIOGRAPHS 6
2.1.2 FIDUCIAL MARKERS 7
2.1.3 FLUOROSCOPY 7
2.1.4 COMPUTED TOMOGRAPHY 7
2.1.5 MAGNETIC RESONANCE IMAGING 8
2.1.6 OPTICAL IMAGING 9
2.2 TRACKING-BASED DELIVERY SYSTEMS 10
2.2.1 LINEAR ACCELERATOR 10
2.2.2 MULTILEAF COLLIMATOR 12
2.2.3 ROBOTIC COUCH 13
2.3 PREDICTION ALGORITHMS FOR RESPIRATORY MOTION 15
2.3.1 MODEL-BASED PREDICTION ALGORITHMS 16
2.3.2 MODEL-FREE PREDICTION ALGORITHMS 26
2.3.3 HYBRID PREDICTION ALGORITHMS 30
2.4 OPEN QUESTIONS FOR PREDICTION OF RESPIRATORY MOTION 39
2.4.1 CHANGES OF RESPIRATORY PATTERNS 39
2.4.2 TUMOR DEFORMATION AND TARGET DOSIMETRY 39
2.4.3 IRREGULAR PATTERN DETECTION 40
2.5 SUMMARY 41

CHAPTER 3 PHANTOM: PREDICTION OF HUMAN MOTION WITH DISTRIBUTED BODY SENSORS 42

3.1 INTRODUCTION 43
3.2 RELATED WORK 47
3.2.1 KALMAN FILTER 47
3.2.2 INTERACTING MULTIPLE MODEL FRAMEWORK 48
3.2.3 CLUSTER NUMBER SELECTION USING GAUSSIAN MIXTURE MODEL (GMM) AND EXPECTATION-MAXIMIZATION (EM) ALGORITHM 50
3.3 PROPOSED GROUPING CRITERIA WITH DISTIBUTED SENSORS 53
3.3.1 COLLABORATIVE GROUPING WITH DISTRIBUTED BODY SENSORS 53

www.manaraa.com

3.3.2 Estimated Parameters used for Interacting Multiple Model Estimator (IMME) 55
3.4 Sensors Multi-channel (MC) IMME: Proposed System Design 58
3.4.1 MC Mixed Initial Condition and the Associated Covariance 59
3.4.2 MC Likelihood Update 60
3.4.3 Switching Probability Update 60
3.4.4 Feedback from Switching Probability Update to Stage 1 for Grouping Criteria with Distributed Sensors 61
3.4.5 Combination of MC conditioned estimates and covariance 61
3.4.6 Computational Time 62
3.5 Experimental Results 65
3.5.1 Motion Data 65
3.5.2 Collaborative Grouping Initialization 67
3.5.3 Comparison of Grouping Methods with Other Techniques 71
3.5.4 Multi-channel (MC) IMME 73
3.5.5 Prediction Overshoot 79
3.5.6 Computational Time 80
3.6 Summary 82

CHAPTER 4 RESPIRATORY MOTION ESTIMATION WITH HYBRID IMPLEMENTATION 83

4.1 Introduction 84
4.2 Related Work 88
4.2.1 Recurrent Neural Network (RNN) 88
4.2.2 Extended Kalman Filter for Recurrent Neural Networks 90
4.3 Multi-channel Coupled EKF-RNN 94
4.3.1 Decoupled Extended Kalman Filter (DEKF) 94
4.3.2 Hybrid Estimation based on EKF for Neural Network (HEKF) 96
4.3.3 Optimized Group Number for Recurrent Multilayer Perceptron (RMLP) 99
4.3.4 Prediction Overshoot Analysis 101
4.3.5 Comparisons on Computational Complexity and Storage Requirement 102
4.4 Experimental Results 105
4.4.1 Motion Data Captured 105
4.4.2 Optimized Group Number for RMLP 105
4.4.3 Prediction Overshoot Analysis 106
4.4.4 Comparison on Estimation Performance 108

www.manaraa.com

4.4.5 Error Performance over Prediction Time Horizon 110
4.4.6 Comparisons on Computational Complexity 112
4.5 Summary 114

CHAPTER 5 CUSTOMIZED PREDICTION OF RESPIRATORY MOTION 115

5.1 Introduction 116
5.2 Prediction Process for Each Patient 119
5.3 Proposed Filter Design for Multiple Patients 122
5.3.1 Grouping Breathing Pattern for Prediction Process 123
5.3.2 Neuron Number Selection 125
5.4 Experimental Results 127
5.4.1 Breathing Motion Data 127
5.4.2 Feature Selection Metrics 127
5.4.3 Comparison on Estimation Performance 130
5.4.4 Prediction Accuracy with Time Horizontal Window 131
5.4.5 Prediction Overshoot Analysis 133
5.4.6 Comparisons on Computational Complexity 136
5.5 Summary 138

CHAPTER 6 IRREGULAR BREATHING CLASSIFICATION FROM MULTIPLE PATIENT DATASETS 139

6.1 Introduction 140
6.2 Related Work 144
6.2.1 Expectation-Maximization (EM) based on Gaussian Mixture model 144
6.2.2 Neural Network (NN) 145
6.3 Proposed Algorithms on Irregular Breathing Classifier 147
6.3.1 Feature Extraction from Breathing Analysis 147
6.3.2 Clustering of Respiratory Patterns based on EM 150
6.3.3 Reconstruction Error for Each Cluster using NN 151
6.3.4 Detection of Irregularity based on Reconstruction Error 152
6.4 Evaluation Criteria for Irregular Breathing Classifier 156
6.4.1 Sensitivity and Specificity 156
6.4.2 Receiver Operating Characteristics (ROC) 158
6.5 Experimental Results 160
6.5.1 Breathing Motion Data 160

www.manaraa.com

6.5.2 Selection of the Estimated Feature Metrics ($\hat{x}$) 161
6.5.3 Clustering of Respiratory Patterns based on EM 162
6.5.4 Breathing Pattern Analysis to Detect Irregular Pattern 163
6.5.5 Classifier Performance 168
6.6 Summary 172

CHAPTER 7 CONCLUSIONS AND CONTRIBUTIONS 173

7.1 Conclusions 173
7.1.1 Hybrid Implementation of Extended Kalman Filter 173
7.1.2 Customized Prediction of Respiratory Motion with Clustering 173
7.1.3 Irregular Breathing Classification from Multiple Patient Datasets 174
7.2 Contributions 176

APPENDIX A 178

A.1 Acronyms Definitions 178
A.2 Symbol Definitions 181

APPENDIX B 185

B.1 Matlab Codes for Neural Network 185
B.2 Matlab Codes for Adaptive Neural Network 188
B.3 Matlab Codes for Kalman Filter 189
B.4 Matlab Codes for Decoupled Extended Kalman Filter 191
B.5 Matlab Codes for Feature Extraction 193
B.6 Matlab Codes for Reconstruction Error 195
B.7 Matlab Codes for Irregular Detection 199
B.8 Matlab Codes for Detection of True Positive and True Negative 202
B.9 Matlab Codes for ROC Curves 205

REFERENCES 208

www.manaraa.com

LIST OF TABLES

Table 1. Instrumentations for Radiation Therapy 12
Table 2. Model-based Prediction Algorithms of Respiratory Motion 26
Table 3. Model-free Prediction Algorithms of Respiratory Motion 30
Table 4. Hybrid Prediction Algorithms of Respiratory Motion 37
Table 5. Comparison of the Computational Complexity (KF vs IMME vs MC-IMME). 62
Table 6. Characteristics of the Motion Data 66
Table 7. Comparison of grouping number methods with AIC values 72
Table 8. Comparison of Overall Velocity error averaged among 8 channels 78
Table 9. Prediction Overshoot Comparison listed in Table 6 80
Table 10. CPU Time Used among the Datasets 81
Table 11. Prediction Overshoot Analysis (HEKF versus DEKF) 108
Table 12. Error Performance among Prediction Time Horizon (HEKF versus DEKF) . 111
Table 13. CPU Time Used in the Target Estimation 112
Table 14. Feature selection metrics with description 123
Table 15. The characteristics of the breathing datasets 127
Table 16. Average Error Performance among a variety of Prediction Time (CNN vs RNN) 132
Table 17. Averaged Prediction Overshoot (CNN vs RNN) 136
Table 18. Comparisons on Computational Complexity 136
Table 19. Feature Extraction metrics including the formula and notation 148
Table 20. Classifier Studies of Irregular Breathing Detection 171

www.manaraa.com

LIST OF FIGURES

Figure 1. CyberKnife System. 11
Figure 2. A multileaf collimator (MLC) with a desired field shape. 13
Figure 3. External view of robotic couch with six degree of freedom. 14
Figure 4. Variable prediction algorithms for respiratory motion. 15
Figure 5. Linear predictor with tapped-delay line. 16
Figure 6. Roles of the variables in the Kalman filter. 18
Figure 7. Explanation of signal history length (SHL). 19
Figure 8. Finite state model with regular breathing cycles. 21
Figure 9. Parameters for Support vector regression. 23
Figure 10. Probabilistic predictive model based on Hidden Markov model. 24
Figure 11. Basic adaptive filtering process for prediction. 27
Figure 12. An artificial neural network with bias input and one hidden layer. 28
Figure 13. Adaptive Neuro Fuzzy Inference System with the total five layers. 31
Figure 14. Adaptive tumor tracking system with two independent signals. 33
Figure 15. An interactive multiple model for respiratory motion prediction. 34
Figure 16. Closed-loop feedback system incorporating EKF for RNN. 36
Figure 17. Prediction Overshoot of IMME. 43
Figure 18. Interacting Multiple Model Estimator 49
Figure 19. Brute-force search algorithm to select the group number. 52
Figure 20. Collaborative group number selection with the adaptive hyper-parameter. 55
Figure 21. General block diagram for the proposed MC-IMME. 58
Figure 22. System design for distributed body sensors has two stages. 59
Figure 23. Polhemus Liberty AC magnetic tracker. 66
Figure 24. Hyper-parameter values based on the target motion data and group number. 68
Figure 25. The difference ($\Delta(G)$) with non-collaborative grouping. 69
Figure 26. The difference ($\Delta_{ADT}(G)$) with collaborative grouping. 70
Figure 27. Sensory Position and Grouping of Motion Data. 71
Figure 28. Comparison of motion tracking estimation for Head_1 dataset. 73
Figure 29. Comparison of accumulated position error of each channel for Head_1. 74
Figure 30. Overall performance of accumulated error among the datasets 74
Figure 31. Error performance among prediction time horizon. 76
Figure 32. Comparison of average velocity estimation of group number 1 for Head_1. . 77
Figure 33. Comparison of the velocity estimations with no feedback/forward vs. feedback/forward. 78
Figure 34. Prediction overshoot comparison between IMME and MC-IMME. 79

www.manaraa.com

Figure 35. A fully connected recurrent neural network with external inputs. 89
Figure 36. Closed-loop feedback system embodying the RMLP and the EKF 92
Figure 37. Decoupled Extended Kalman Filter (DEKF) for RNN. 95
Figure 38. Prediction overshoots with DEKF. 96
Figure 39. Hybrid motion estimation based on EKF (HEKF) for RNN. 99
Figure 40. Comparison of objective function values between HEKF and DEKF. 106
Figure 41. Comparison of prediction overshoot between HEKF and DEKF. 107
Figure 42. Target estimation between HEKF and DEKF. 109
Figure 43. Comparison of position error between HEKF and DEKF 110
Figure 44. Multiple marker interactions for the individual patient. 119
Figure 45. Interactive process for multiple patients 122
Figure 46. Dominant feature selection with Feature Combination Vector. 128
Figure 47. Clustering of 130 patients datasets with the dominant class number. 129
Figure 48. Comparison on estimation performance of DB89 with 192 ms latency. 130
Figure 49. Error Performance with different Prediction Time Horizons (CNN vs RNN). 132
Figure 50. Prediction Overshoot Comparison 134
Figure 51. Prediction overshoot comparison over all the patients with 192ms latency. 135
Figure 52. Irregular Breathing Pattern Detection with the proposed algorithm. 147
Figure 53. Reconstruction Error to detect the irregular pattern using NN 152
Figure 54. Detection of regular/irregular patterns using the threshold value (ξ_m) 154
Figure 55. True positive range (R^{TP}) vs. True negative range (R^{TN}). 157
Figure 56. Frequency distribution of recording times for the breathing datasets. 160
Figure 57. Objective functions for selection of feature metrics 162
Figure 58. Quantitative model analysis for the selection of cluster number. 163
Figure 59. Frequency distribution of breathing cycle (BC_i) for the breathing datasets. . 164
Figure 60. Frequency distribution of the number of irregular patterns ($\Sigma_j \psi_{ij}$) 165
Figure 61. Frequency distribution of ratio (γ_i). 166
Figure 62. Representing regular breathing patterns. 166
Figure 63. Representing gray-level breathing patterns. 167
Figure 64. Representing irregular breathing patterns. 167
Figure 65. ROC graph of irregular detection with different observation period. 168
Figure 66. ROC graph of irregular detection with different regular thresholds and observation period 169
Figure 67. Area under the ROC curve. 170

www.manaraa.com

PREFACE

This dissertation is mainly based on the following papers which will be referred to in the text by their Roman numerals.

I. Suk Jin Lee, Y. Motai, A. Docef, E. Weiss, G. D. Hugo, and M. Murphy, "Prediction of Respiratory Motion: a Review," In preparation.

II. Suk Jin Lee, Y. Motai, and H. Choi, "Distributed Sensory System with Multi-channel Interacting Multiple Model," In Review, 2011.

III. Suk Jin Lee, Y. Motai, and M. Murphy, "Respiratory Motion Estimation with Hybrid Implementation of Extended Kalman Filter," IEEE Trans. Industrial Electronics, vol. PP, no. 99, pp. 1 – 1, 2011.

IV. Suk Jin Lee, Y. Motai, E. Weiss, and S. S. Sun, "Customized Prediction of Respiratory Motion with Clustering from Multiple Patient Interaction," In Review, 2011.

V. Suk Jin Lee, Y. Motai, E. Weiss, and S. S. Sun, "Irregular Breathing Classification from Multiple Patient Datasets," In Review, 2012.

www.manaraa.com

ABSTRACT

PREDICTION OF RESPIRATORY MOTION

By SUK JIN LEE

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2012.

Major Director: Yuichi Motai, Ph.D.
Assistant Professor, Department of Electrical and Computer Engineering

Radiation therapy is a cancer treatment method that employs high-energy radiation beams to destroy cancer cells by damaging the ability of these cells to reproduce. Thoracic and abdominal tumors may change their positions during respiration by as much as three centimeters during radiation treatment. The prediction of respiratory motion has become an important research area because respiratory motion severely affects precise radiation dose delivery. This study describes recent radiotherapy technologies including tools for measuring target position during radiotherapy and tracking-based delivery systems.

In the first part of our study we review three prediction approaches of respiratory motion, *i.e.*, *model-based* methods, *model-free* heuristic learning algorithms, and *hybrid* methods. In the second part of our work we present a phantom study—prediction of human motion with distributed body sensors—using a Polhemus Liberty AC magnetic tracker. In the third part of our work we propose respiratory motion estimation with hybrid implementation of extended Kalman filter. The proposed method uses the recurrent neural network as the role of the predictor and the extended Kalman filter as the role of the corrector. In the fourth part of our work we further extend our research work to

www.manaraa.com

present customized prediction of respiratory motion with clustering from multiple patient interactions. For the customized prediction we construct the clustering based on breathing patterns of multiple patients using the feature selection metrics that are composed of a variety of breathing features. In the fifth part of our work we retrospectively categorize breathing data into several classes and propose a new approach to detect irregular breathing patterns using neural networks. We have evaluated the proposed new algorithm by comparing the prediction overshoot and the tracking estimation value. The experimental results of 448 patients' breathing patterns validated the proposed irregular breathing classifier.

www.manaraa.com

CHAPTER 1 INTRODUCTION

Rapid developments in radiotherapy systems open a new era for the treatment of thoracic and abdominal tumors with accurate dosimetry [1]. For accurate treatment planning and target motion acquisition, radiotherapy systems should take into consideration not only technical limitations, but also physiological phenomena, especially respiratory motion [1] [2]. The delivery system cannot respond instantaneously to target position measurement since this measurement itself takes some time. Target prediction method due to respiratory motion is proposed as a solution to increase targeting precision before or during radiation treatments [1] [3]. The significant merit of predicting respiratory motion is that radiotherapy can be delivered more accurately to target locations, reducing the volume of healthy tissue receiving a high radiation dose [1]. The objective of this study is to deliver a comprehensive review of current prediction methods for respiratory motion and propose a new prediction method of respiratory motion.

Respiratory motion severely affects precise radiation dose delivery because thoracic and abdominal tumors may change locations by as much as three centimeters during radiation treatment [3] [79] [80]. A number of methods to mitigate the effect of respiratory motion are widely used in radiotherapy systems [1]. Respiratory gating methods can deliver radiation treatment within a specific part of the patient's breathing cycle (referred to as *gate*), where radiation is activated only when the respiratory motion is within a predefined amplitude or phase level [2] [81]. Breath-hold methods, exemplified by the deep inspiration breath hold, have been prominently used for lung cancer radiotherapy, where the therapists may turn on the beam only if the target breath-hold level is reached; otherwise, the treatment is withheld [1].

www.manaraa.com

Real-time tumor tracking is a more advanced technique to dynamically adjust the radiation beam according to the change of tumor motion [1], where variations in tumor motion caused by respiratory motion should be minimized with the precise patient positioning system [58]. If the acquisition of tumor position and the repositioning of the radiation beam are not well synchronized, a large volume of healthy tissue may be irradiated unnecessarily and the tumor may be underdosed [20] [21] [89] [90] [91] [92]. There exists a finite time delay (or system latency) between measuring and responding to real-time measurement [1] [47] [51] [54]. Due to the magnitude of the time delay, for real-time tumor tracking, the tumor position should be predicted, so that the radiation beams can be adjusted accordingly to the predicted target position during radiation treatment [8] [37] [93].

The state-of-the-art prediction of respiratory motion has been widely addressed [4] [5] [7] [31] [32] [33] [34] [35] [36] [37]. Although there have been many proposed methodologies of prediction algorithms for respiratory motion, they lack the overall survey or benchmark studies among the methods [44]. The main problem of comparing all the studies is involving the complexities from a combination of radiation technologies and algorithms, which makes it hard to identify which approach is the best one [4] [5] [35] [43] [45]. Thus, in this study, we intend to list all of the relevant items in a systematic manner so that the reader can get to know all significant and representative approaches [44].

Research studies on the prediction of respiratory motion were carried out in the areas of medical physics or biology to give precise treatments to remove tumor or cancer cells without unnecessarily irradiating healthy tissues in intensity-modulated radiation therapy

www.manaraa.com

[1] [2] [3] [4] [5] [6] [8] [9] [10] [11] [12] [13] [23] [24] [25] [26] [27] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43]. Several IEEE Transactions journals, including Transactions on Medical Imaging [14] [15] [16] [17] [100] [101], Biomedical Engineering [19] [20] [21] [22] [78], and Nuclear Science [76], have presented a variety of prediction and modeling methods based on fiducial markers and computed tomography (CT) images. For example, Sarrut *et al*. showed a strategy and criteria to determine the correctness of breathing motion tracking from CT imaging [16]. By providing background information, this research will stimulate the interest of readers in biomedical applications and encourage collaborative research activities in the biomedical and medical physics areas [102].

The objective of this study is to deliver a comprehensive review of current prediction methods of respiratory motion and propose a new method to predict respiratory motion with variable breathing features. Before we start to describe the prediction methods, we will present basic radiotherapy technologies for the brief understanding of radiotherapy and previous prediction methods of respiratory motion in Chapter 2. This study will show three prediction methods of respiratory motion, including *model-based*, *model-free*, and *hybrid* prediction algorithms in Chapter 2. In the following chapter, we will show a phantom study—prediction of human motion with distributed body sensors—using a Polhemus Liberty AC magnetic tracker. In Chapter 4, we propose hybrid implementation based on EKF (HEKF) for respiratory motion estimate. Here, the recurrent neural network (RNN) performs the role of the predictor and the extended Kalman filter (EKF) performs the role of the corrector. In Chapter 5, we further extend our research work to present customized prediction of respiratory motion with multiple patient interactions

www.manaraa.com

using neural network (CNN). For the pre-procedure of prediction for individual patient, we construct the clustering based on breathing patterns of multiple patients using the feature selection metrics that are composed of a variety of breathing features. In the intra-procedure, the proposed CNN used neural networks (NN) for a part of the prediction and EKF for a part of the correction. In Chapter 6, we retrospectively categorize breathing data into several classes and propose a new approach to detect irregular breathing patterns using neural networks, where the reconstruction error can be used to build the distribution model for each breathing class. The sensitivity, specificity and receiver operating characteristic (ROC) curve of the proposed irregular breathing pattern detector was analyzed.

www.manaraa.com

CHAPTER 2 REVIEW: PREDICTION OF RESPIRATORY MOTION

Radiation therapy is a cancer treatment method that employs high-energy radiation beams to destroy cancer cells by damaging the ability of these cells to reproduce [55]. In external beam radiotherapy (EBRT), specific parts of the patient's body are exposed to the radiation emanating from a treatment machine [50] [55] [122]. The X-ray beams have to penetrate other body tissues to reach the target area during treatment process. This leads to unnecessary irradiation of healthy tissues around the tumors. Accordingly, prediction of respiratory motion is a very critical issue in EBRT. Radiation technologies can consist of two major approaches: 1) tools for measuring target position during radiotherapy [11] [18] [19] [20] [39] [92], where patient-specific treatment parameters including acquisition of respiratory patterns, treatment simulation, and target area planning are determined for treatment preparation, and 2) tracking-based delivery systems [43] [79] [86] [106] [107], where the patient is placed under the linear accelerator and radiation is delivered using real-time tracking methods under free breathing conditions.

www.manaraa.com

2.1 Tools for Measuring Target Position during Radiotherapy

Measuring target position for treatment planning in radiotherapy is heavily dependent on image processing and patient-specific interpretation methods for medical data and images [7] [14] [15] [16] [17] [70] [71] [72] [73] [76] [78]. There exist several measuring tools for the target position. Once the target is identified, it is easy to track this defined target in most imaging modalities [1]. A number of medical imaging, such as radiographs, fluoroscopy, computed tomography (CT), magnetic resonance imaging (MRI), and optical imaging can provide real-time information in company with outstanding visualization to improve the treatment results during beam delivery [73]. It is difficult to detect the target directly in images. The fiducial markers are often employed to act as surrogates for optical signal tracking.

2.1.1 Radiographs

Radiographs (referred to as plain X-rays) are photographic images produced by the activity of X-ray or nuclear radiation to view a non-uniformed physical object. The rays may penetrate the human body through the different density and structure of the object. The rays that pass through are recorded behind the object with a detector which can display the different density and structure of the body. Generally, radiographies are generated by X-ray beams, whereas in nuclear medicine gamma rays are involved [151]. Radiographs are unceasingly used and employed as a major tool to detect and measure the target position [54].

www.manaraa.com

2.1.2 Fiducial Markers

Fiducial markers located around the tumor position are often employed to act as surrogates for optical signal tracking, to synchronize the internal and external breathing motion signals, and to provide real-time information during beam delivery [1] [2] [6] [49] [93] [108] [109]. In real-time tumor tracking, multiple implanted fiducial markers are detected as surrogate on the images of fluoroscopy systems for accurate tumor location, but their use can be limited due to the risk of pneumothorax during marker implantation [6, 108-109]. External fiducial markers are also attached on the patient's chest for respiratory gated radiotherapy, where they can be used to correlate internal breathing motion with external optical signal based on the infrared tracking system [1] [50] [159].

2.1.3 Fluoroscopy

Fluoroscopy is a method for obtaining real-time moving images of deep body structures using fluoroscope [1]. A patient is placed between an X-ray tube and fluorescent screen during fluoroscopic procedures. Modern fluoroscopes are associated with an image intensifier and video camera so that they can display a continuous series of images with maximum 25-30 images per second [152]. Fluoroscopy is often used not only to watch the digestive track but also to track moving organs during therapeutic procedures [49] [54] [58].

2.1.4 Computed Tomography

Computed Tomography (CT) [11] [34] [38] [73] is a specialized X-ray imaging method employing a series of individual small X-ray sensors with computer processing. Here,

www.manaraa.com

medical data come together with multiple angles, and a computer treats this information to generate an image (referred to as "cut"). The vision of body images is similar to the vision of a sliced bread loaf. CT images are widely used for diagnostic purposes, especially for diagnosing a variety of tumors including lung, pancreas, liver, and other thoracic and abdominal tumors, because using CT images can not only validate that tumors exist, but they also determine tumor position and size to provide clear images for radiation treatment planning [18] [39] [90]. X-ray computed tomography (CT) including computed axial tomography (CAT) and cone beam CT (CBCT) uses rotating X-ray equipment with a digital computer to produce a clear medical image for all types of tissues [117].

2.1.5 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a medical imaging method that uses the property of nuclear magnetic resonance, instead of radiative delivery to the patient to visualize the internal organs and tissues for diagnosis and therapy planning. MRI aligns the protons in the water atoms within the patient using a strong magnetic field. Then, a very sensitive radio antenna detects the resonance signal of the protons that are activated by the electromagnetic pulse of the scanner [151]. In MRI, the picture of body images looks similar to a "cut" in CT. MRI provides good contrast between the different soft tissues compared with X-ray CT, so that it can create a highly detailed image of the scanned body structures in the soft tissues [118]. The integrated and hybrid MRI modalities also proposed to improve the treatment outcome [73] [118].

www.manaraa.com

2.1.6 Optical Imaging

Optical Imaging is a non-invasive imaging method that takes photographs of biological tissues or organs using visible, ultraviolet, and infrared wavelengths for clinical diagnosis [153]. Unlike X-ray photons that penetrate the entire biological tissue, optical photons interact with biological tissue medium by the property of absorption and elastic scattering [154]. Advanced optical imaging modalities have been recently developed, so they can provide cost-effective and much higher resolution images than current CT and MRI images [153]. Optical imaging system consisting of infrared cameras and external markers can also provide accurate position of target tracking during the treatment process in real-time [103].

www.manaraa.com

2.2 Tracking-based Delivery Systems

Conventional radiotherapy systems used linear accelerators with gantry mechanism to delivery the radiation beam to the targeting areas [10]. Due to the breathing-induced tumor motion, breath-holds and gating methods are used to reduce underdosing of tumor parts and overdose to surrounding critical parts [155]. Multileaf collimator (MLC)-based and couch-based tracking methods also have been developed for real-time tumor tracking under free breathing conditions [42] [43] [49] [86] [142] [155] [156] [157] [158].

2.2.1 Linear Accelerator

Linear Accelerator (Linac) is the medical device to generate the therapeutic beam for EBRT treatment [10]. Linacs accelerate electrons by high-voltage electric fields, and then let these electrons collide with source target to produce high-energy X-ray beams. Linacs may be equipped with specialized blocks or a multileaf collimator (MLC) in the head of machine to conform fields to the shape of the patient's tumor. Finally, the customized beam can be delivered by a gantry mechanism (such as robotic arms) to specific parts of the patient to destroy the malignant tumors [10] [92].

For example, CyberKnife is a well-known image-guided radiosurgery system for Linac applications [56]. The two main elements of the system are the linear particle accelerator to generate radiation for treatment, and a robotic arm to allow the radiation to be delivered at any target area of the body with six degrees of freedom [54]. Advanced image guidance technology, *e.g.*, X-ray sources to generate orthogonal X-ray images, is used to detect the bony landmarks location, implanted fiducials or soft tissue tumors. IR

www.manaraa.com

tracking system synchronized with the tumor motion can reduce safety margins for respiratory gating or breath-hold techniques, as shown in Fig. 1 [10] [50] [124].

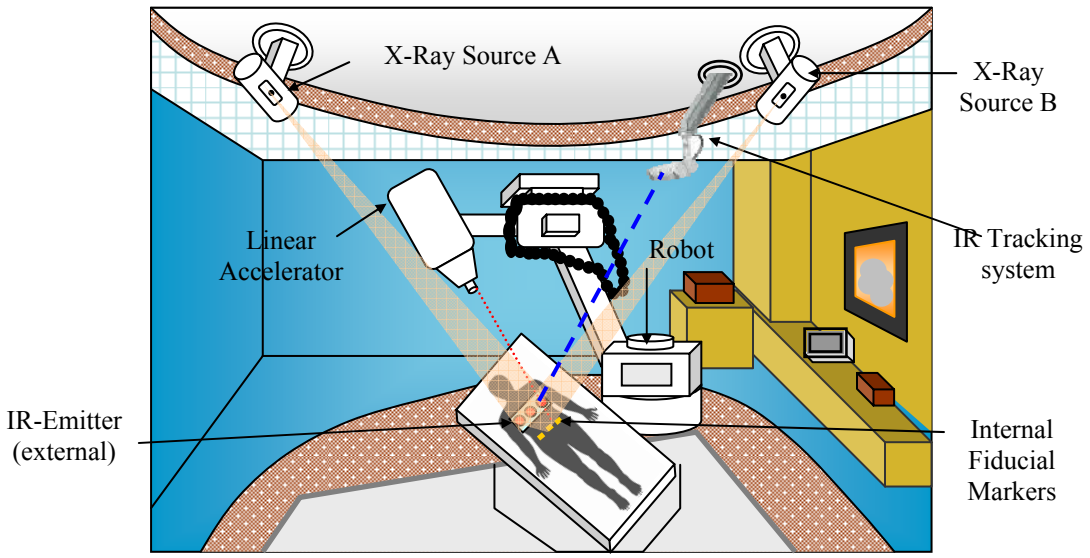

Figure 1. CyberKnife System.
X-ray source with low energy is used to detect soft tissue tumors or implanted fiducial markers during the treatment. IR tracking system synchronized with the tumor motion can reduce safety margins for respiratory gating or breath-hold techniques [10] [56].

The simple treatment process includes planning, repetition of verification and targeting, and treatment delivery. In the planning process, X-ray image scanning and advanced treatment planning are prepared. In the repetition of verification and targeting process, the image-guided radiosurgery system verifies clinical tumor location. If any variation is detected in the tumor position, the robotic arm is replaced according to the tumor movement based on a frame. In the treatment process, the sophisticated radiation beam for radiosurgery is delivered to the tumor [56]. The synchrony respiratory tracking system is widely used to continuously synchronize the delivery of radiation beam to the motion of the tumor for real-time tumor tracking [32] [33] [36] [84] [85] [86] [124] [136] [138].

www.manaraa.com

Table 1. Instrumentations for Radiation Therapy

Radiotherapy systems	Development
CyberKnife robotic treatment [56]	Accuray, Inc., Sunnyvale, CA
Varian Real-time Position Management system [57] [75]	RPM system, Varian Medical, Palo Alto, CA
Real-time tumor-tracking system [51] [52]	RTRT system, Mitsubishi Electronics Co., Ltd., Tokyo
Elekta system [41] [42] [43] [77]	Elekta Ltd, Stockholm, Sweden
Siemens Radiation Oncology system [66] [67]	Siemens AG, Munich, Germany

There are many radiation therapy equipments to support prediction of respiratory motion with advanced radiotherapy technologies [144]. The outline of all the radiotherapy systems is out of scope in this study. Among many radiation therapy systems, some radiotherapy equipments are widely used for the management of respiratory motion [1], such as CyberKnife robotic treatment device (Accuray, Inc., Sunnyvale, CA) [56] [124], Real-time Position Management system (RPM system, Varian Medical, Palo Alto, CA) [57] [75], Real-time tumor-tracking system (RTRT, Hokkaido University) [51] [52] [74], Elekta system (Elekta Ltd, Stockholm, Sweden) [41] [42] [43] [77], and Siemens Radiation Oncology system (Siemens AG, Munich, Germany) [66] [67]. Therefore, we describe five main radiotherapy equipments as shown in Table 1.

2.2.2 Multileaf Collimator

Multileaf collimator (MLC) is a sophisticated system for radiation therapy dose delivery, made up of separate leaves that can move independently in and out of a particle beam path to form a desired field shape as shown in Fig. 2. The advantage of MLC is that it can simply change an individual leaf for the field shape with controlling remote computer and save treatment preparation time by eliminating clinician's entering the treatment room [142].

www.manaraa.com

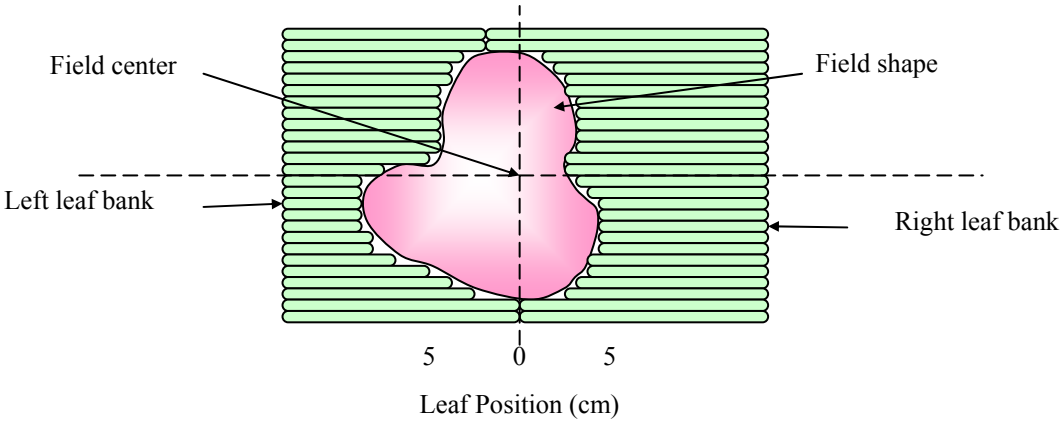

Figure 2. A multileaf collimator (MLC) with a desired field shape.
MLC is made up of separate leaves that can move independently in and out of a particular beam path to form a desired filed shape.

Sawant *et al.* proposed an integrated system by combining an independent position monitoring system and an intensity-modulated radiotherapy (IMRT) delivery system based on dynamic MLC (DMLC). In [86], they investigated two important parameters, *i.e.*, system latency and geometric accuracy. To reduce the system latency, the tracking algorithm used a modified linear adaptive filter with continuous megavoltage X-ray images of three implemented transponders at approximately seven frames per second. The geometric accuracy was calculated by comparing the aperture center of each image frame with the target motion trajectories. MLC-based tracking method may increase the treatment accuracy and decrease the treatment time compared to breath-holds and gating methods [156] [157].

2.2.3 Robotic Couch

A robotic couch can be used to compensate for breathing-induced tumor motion with extra degree of precision for patients in real time [42] [43]. For the couch-based tracking method, a robotic couch system consists of stereoscopic infrared cameras and the couch

www.manaraa.com

system moves in response to any changes in angle and position of organ motion detected by the cameras during treatment delivery [49] [155].

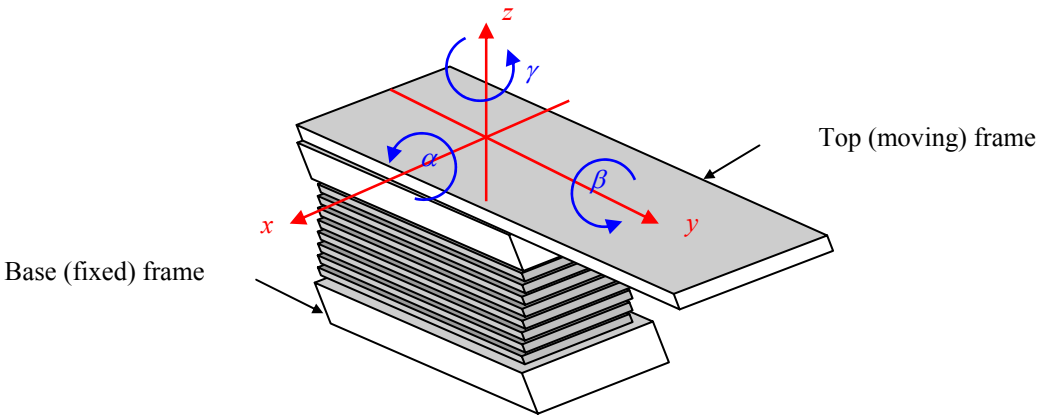

Figure 3. External view of robotic couch with six degree of freedom.
The couch system consists of top (moving) frame linked with a fixed base frame using independent mechanical legs. Here the top platform is defined by six independent position-orientation variables – coordinates (x, y, z, α, β, γ) [43].

Fig. 3 shows HexaPOD robotic couch with six degrees of freedom. The couch system consists of top (moving) frame linked with a fixed base frame using independent mechanical legs. Here the top platform is defined by six independent position-orientation variables – coordinates (x, y, z, α, β, γ) [43] [158]. The commercially available robotic couches can arrange the patient position according to the treatment procedure with highly accurate level; however, they lack compensation for the respiratory and cardiac motion [42-43].

www.manaraa.com

2.3 PREDICTION ALGORITHMS FOR RESPIRATORY MOTION

A number of prediction methods for respiratory motion have been investigated based on surrogate markers and tomography images [12] [14] [18] [31] [32] [33] 34] [37] [46] [47] [48] [53]. The previous methods can be categorized into three approaches: 1) *model-based* approach [31] [32] [33] [34] [46] [48] which uses a specific biomechanical or mathematical model for respiratory motion functions or models; 2) *model-free* approach [35] [36] [37] [41] [46] heuristic learning algorithms that are trained based on the observed respiratory patterns ; 3) *hybrid* approach [40] [45], which uses united methods to combine more than two methods, resulting in outperforming the previous solitary method. These three approaches are described in the following Chapters 2.3.1, 2.3.2, and 2.3.3 respectively. Fig. 4 shows the key studies, which have more than 30 references in the last 10 years, representing the salient algorithms covered.

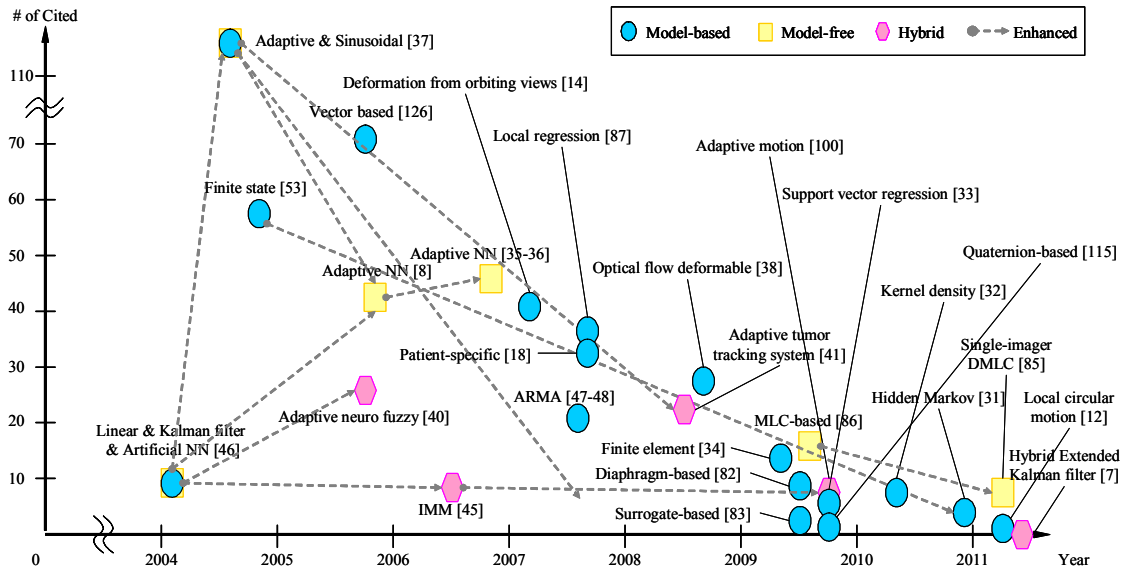

Figure 4. Variable prediction algorithms for respiratory motion.
This figure shows the key studies, which have more than 30 references in the last 10 years, representing to the salient algorithms covered.

www.manaraa.com

2.3.1 Model-based Prediction Algorithms

Generally, *model-based* methods include 1) linear prediction [31] [37] [44] [46], 2) Kalman filter [4] [12] [31] [44] [45] [46] [128], 3) sinusoidal Model [37] [44], 4) finite state model [31] [44] [53], 5) autoregressive moving average model [44] [47] [48], 6) support vector machine [20] [33] [44] [137] [138] [139] [140], and 7) hidden Markov model [31] [53]. Especially, linear approaches and Kalman filters are widely used for the fundamental prediction approach of respiratory motion among a variety of investigated methods [14] [18] [31] [32] [33] [34] [37] [38] [46] [47] [48] [53] [82] [83] [87] [115] [126].

1) Linear Prediction

A linear prediction is a mathematical system operation where future output values are estimated as a linear function of previous values and predictor coefficients, as follows [44] [46]:

$$\hat{x}(t) = a_0 + a_1 x(t-1) + \cdots + a_n x(t-n) = \sum_{i=0}^{n} a_i x(t-i), \qquad (1)$$

where $\hat{x}(t)$ is the predicted value or position at time t.

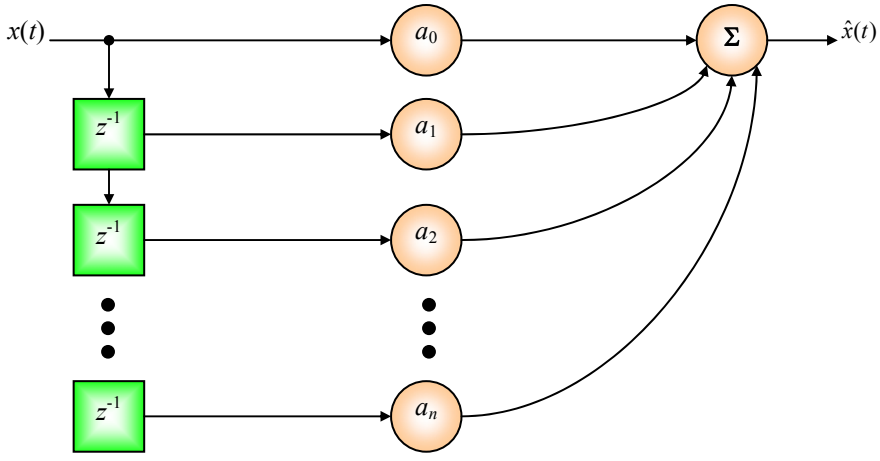

Figure 5. Linear predictor with tapped-delay line.
The predicted value is a linear combination of previous observations $x(t\text{-}n)$ and predictor coefficients a_n that are not changing over time.

www.manaraa.com

The predicted value is a linear combination of previous observations $x(t–n)$ and predictor coefficients a_n that are not changing over time, as shown in Fig. 5. In a linear prediction, it is a significant task to solve a linear equation to find out the coefficients a_n that can minimize the mean squared error between the predicted values and previous values [46]. The linear model is widely used in the early stage to compare the prediction performance with other models, *e.g.* neural network prediction and Kalman filtering [31] [46]. Sharp *et al*. revealed that the root mean squared error (RMSE) for the prediction accuracy is around 2.2mm with 200ms latency [46]. The limitation of this model is that it is not robust to some changes from one linear state to another [31]. This model can be enhanced into nonlinear (sinusoidal) and adaptive models as shown in Fig. 4 [37].

2) Kalman Filter

The Kalman filter (KF) is one of the most commonly used prediction methods in real-time filtering technologies [4] [12] [31] [44] [45] [46]. KF provides a recursive solution to minimize mean square error within the class of linear estimators, where linear process and measurement equations to predict a tumor motion can be expressed as follows [128]:

$$\hat{x}(t) = Fx(t-1) + Bu(t-1) + W, \quad z(t) = H\hat{x}(t) + V, \tag{2}$$

where we denote the state transition matrix as F, the control-input matrix as B, and the measurement matrix as H. $u(t)$ is an n-dimensional known vector, and $z(t)$ is a measurement vector. The random variables W and V represent the process and measurement noise with the property of the zero-mean white Gaussian noise with covariance, $E[W(t)W(t)^T] = R(t)$ and $E[\mathrm{V}(t)V(t)^T] = Q(t)$, respectively. The matrices F, B, W, H, and V are assumed known and possibly time-varying.

www.manaraa.com

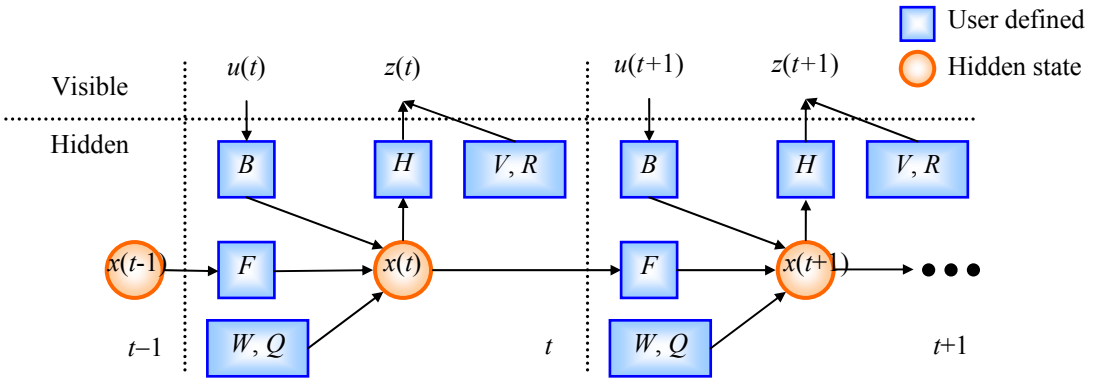

Figure 6. Roles of the variables in the Kalman filter.
$u(t)$ is an n-dimensional known vector, and $z(t)$ is a measurement vector. The next state is calculated based on the dynamic equation, such as $x(t+1)=Fx(t) + Bu(t) + V$. Here, V and W are process noise and measurement noise with covariance R and Q.

In KF, the predicted position $\hat{x}(t)$ can be derived from the previous state $x(t\text{-}1)$ and the current measurement $z(t)$ [44] [128]. Sharp *et al.* showed that RMSE for the prediction accuracy is around 2.5mm with 200ms latency [46]. Because of state update process with new data, KF is effective for linear dynamic systems, but prediction accuracy is degraded when breathing patterns change from one linear state to another [31]. KF was enhanced to interactive multiple model (IMM) filter with constant velocity (CV) and constant acceleration (CA) based on KF by Putra *et al*. in Fig. 4 [4] [45]. Hong *et al*. also suggested the first-order extended Kalman filter (EKF) can be used to process and update the state estimate [12].

3) Sinusoidal Model

Regular respiratory motion shows a continuous sinusoidal pattern with respect to the time sequence. This sinusoidal curve can be adjusted to respiratory motion over signal history length (SHL). We show Fig. 7 to clarify the ideas of SHL, response time (Δ), and prediction error for a single point of respiratory motion trace. Let $x(t)$ denote the actual respiratory motion curve at time t after SHL. Vedam *et al*. represented a sinusoidal wave model to estimate the predicted position for a given response time (Δ), as follows [37]:

www.manaraa.com

$$x_{pred}(t+\Delta) = x_{act}(t) + \left[x_{SHL}(t+\Delta) - x_{SHL}(t)\right], \quad (3)$$

where $x_{SHL}(t)$ is a fitted sinusoidal curve including SHL, given by $x_{SHL}(t)=A\sin(Bt+C)+D$ with time sequences from t-SHL to t (t>SHL, and A, B, C, and D are the parameters of sinusoidal waveform model) [37].

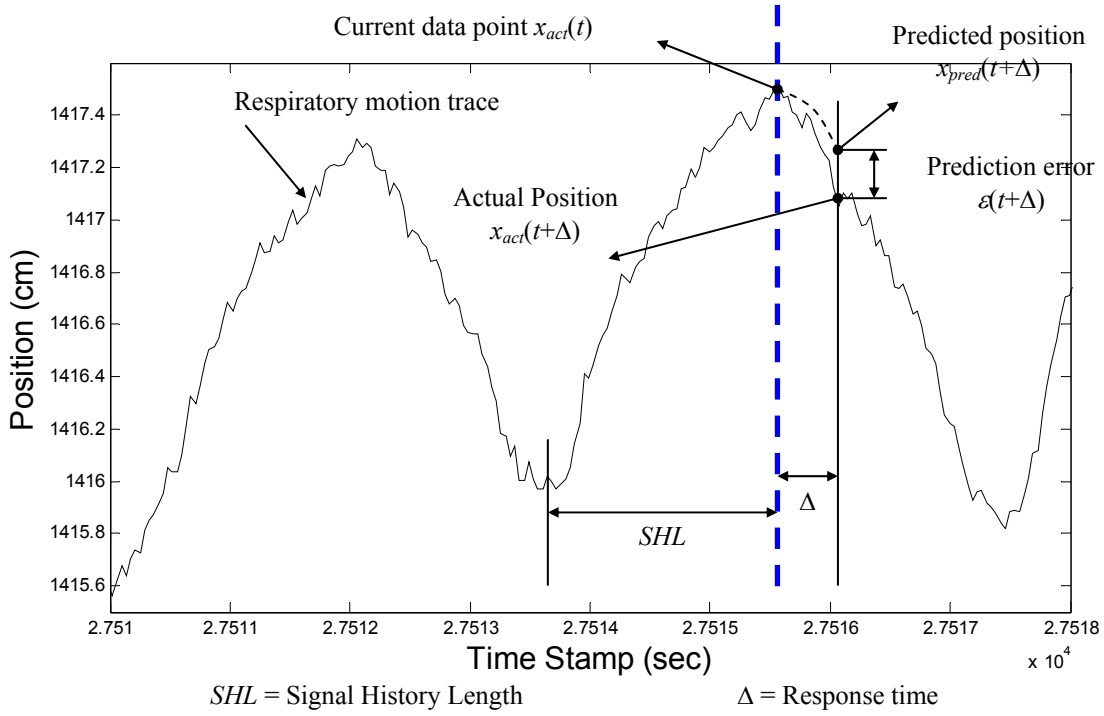

Figure 7. Explanation of signal history length (SHL)
Explanation of SHL, response time (Δ) and prediction error with respect to the current data point. Let $x(t)$ denote the actual respiratory motion curve at time t after SHL. The predicted position $x_{pred}(t+\Delta)$ can be calculated based on the sinusoidal curve fit model over SHL.

Vedam *et al*. evaluated that the prediction error with 200ms latency is less than 2mm. This model also has a limitation with 1-dimensional prediction and the prediction accuracy degrades with long latency [37] [44].

4) Finite State Model

www.manaraa.com

The breathing motion can be analyzed based on its natural understanding of breathing states [53]. In finite state model (FSM), a regular respiratory motion is subdivided into three states – exhale (EX), end-to-exhale (EOE), and inhale (IN), as shown in Fig. 8 [44] [53]. The other motions are categorized as irregular breathing (IRR) except the above three states in this approach. Wu *et al*. represented the finite state automation for the transition from one state to another [53]. Line segments for finite states in Fig. 8 are determined by the velocity of tumor motion and the average amplitude for two connected directed line segments. Let $X(t) = \{x_0, x_1,..., x_n\}$ as an n-dimensional vector point at time t. The length of a directed line segment from $X(t_0)$ to $X(t_1)$ is expressed as follows:

$$\| \overrightarrow{X_0 X_1} \| = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{0i})^2} \,. \tag{4}$$

The velocity of tumor motion is calculated with two vector points ($X(t_0)$ and $X(t_1)$), as follows:

$$v(t_0 \to t_1) = \frac{\| \overrightarrow{X_0 X_1} \|}{t_0 - t_1} \,. \tag{5}$$

This method provides not only a statistically quantitative analysis of motion characteristics, but also good prediction results, *i.e.*, average RMS error less than 1mm. However, the study on FSM is restricted to a one dimension model. This method was enhanced into a three dimension version with hidden Markov model by Kalet *et al.* [31].

www.manaraa.com

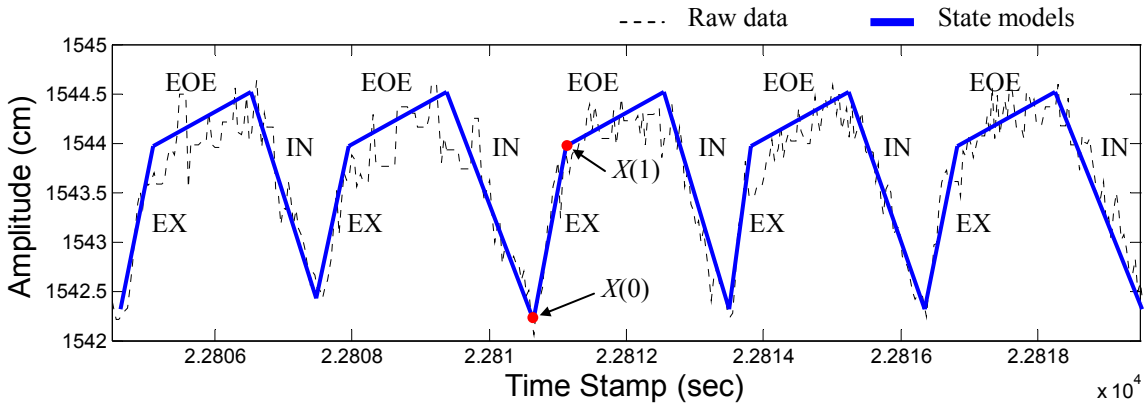

Figure 8. Finite state model with regular breathing cycles.
The breathing motion patterns are modeled with irregular (IRR), exhale (EX), end-to-exhale (EOE), and inhale (IN) breathing states [53]. State transitions are initiated by the velocity of tumor motion with two vector points $X(t_0)$ and $X(t_1)$.

5) Autoregressive moving average model

Autoregressive moving average (ARMA) model is a mathematical generalization of the linear model with time series data and signal noise, and widely used to predict motion patterns of a time series from past values [44] [47] [48]. ARMA consists of two models: 1) an autoregressive (AR) model represented by a weighted sum of the present and past positions with a polynomial order p, *i.e.*, $\varphi_1 x(t-1) + \cdots + \varphi_p x(t-p)$, and 2) a moving average (MA) model represented by a weighted sum of the present and past signal noise with a polynomial order q, *i.e.*, $\theta_1 \varepsilon(t-1) + \cdots + \theta_q \varepsilon(t-q)$ [44] [47]. The mathematical notation ARMA (p, q) with polynomial orders of p AR and q MA is expressed as follows [48]:

$$\hat{x}(t) = \varepsilon(t) + \sum_{i=1}^{p} \varphi_i x(t-i) + \sum_{i=1}^{q} \theta_i \varepsilon(t-i), \tag{6}$$

where we define φ_i as the parameter of the AR model, and θ_i as the parameter of MA model, respectively. The error terms $\varepsilon(t)$ are the white noise assuming to be independent and identically distributed random variables. The order of ARMA model was built on the combination of p and q with maximizing the Akaike information criterion. There is no

www.manaraa.com

limitation with sampling data and processing time to select the orders p and q. However, McCall *et al.* demonstrated that up to ARMA (4, 4) models were preferred and the ARMA (2, 1) models achieved the optimized mean prediction errors over all the latency investigated [48]. Ren *et al*. also showed that the standard deviation of the position is below 2.6mm with prediction in contrast with 4.6mm without prediction [47].

6) Support Vector Machine

Support vector machines (SVMs) are supervised learning methods that are widely used for classification and regression analysis [33] [137] [138] [139] [140]. For medicine applications, they have been used to predict lung radiation-induced pneumonitis from patient variables and compute the future location of tumors from patient geometry and clinical variables [20] [44] [140]. Let define $G(x)$ as an unknown function (truth) with d-dimensional input vector x,= $[x_1,...,x_d]$, $F(x, \hat{w})$ as a function with estimation $\hat{w}$ derived from minimizing a measurement error between $G(x)$ and $F(x, \hat{w})$. Using N training samples v_i, i = 1,..., N, the primal objective function with a loss function $L(\cdot)$ can be expressed, as follows [139]:

$$C\sum_{i=1}^{N} L[y_i - F(v_i, \hat{w})] + \left\|\hat{w}\right\|^2, \tag{7}$$

where, C is a control value to adjust a balance, y_j is the observation of $G(x)$ in the presence of noise. The function $L(\cdot)$ is a general loss function with user defined threshold ε, as shown in Fig. 9, *i.e.*, if the observation is within the threshold ($|y_i\text{-}F(x_i, \hat{w})|<\varepsilon$), the loss is zero; otherwise, the loss is the amount of the difference between the predicted value and the threshold ε, such as ($|y_i\text{-}F(x_i, \hat{w})|-\varepsilon$) [137] [139]. Based on the loss function and the threshold, the objective function (7) is calculated by solving the optimization problem as follows:

www.manaraa.com

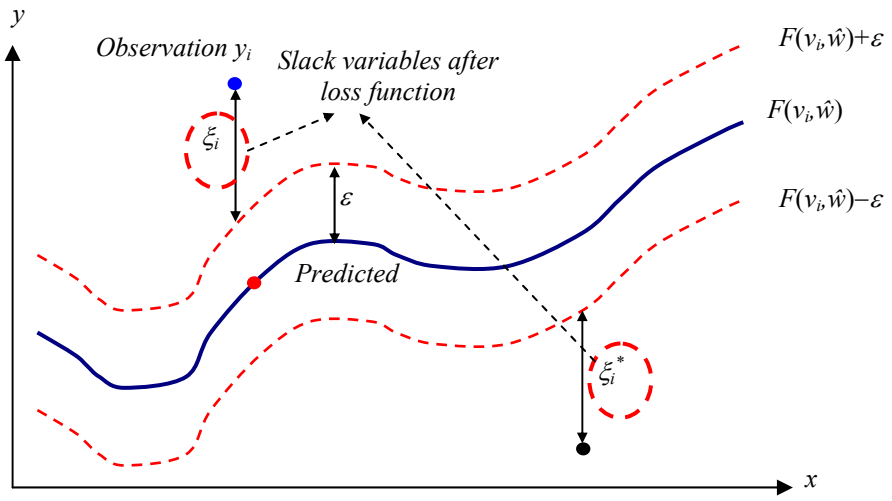

Figure 9. Parameters for Support vector regression.

Let define ε as a user defined threshold, and v_i (i=1,..., N) as N training samples. The loss function is defined using the threshold ε, such as if the observation is within the threshold, the loss is zero; otherwise, the loss is the amount of the difference between the predicted value and the threshold ($|y_i-F(v_i, \hat{w})|-\varepsilon$).

$$\min_{w} C\left(\sum_{i=1}^{N}\xi_i^* + \sum_{i=1}^{N}\xi_i\right) + \frac{1}{2}(w^t w), \tag{8}$$

where ξ_i and ξ_i^* are slack variables as shown in Fig. 9. A control value C is used to adjust the balance between the error term and the weight concentration [139]. This optimization problem can be resolved by the Lagrangian relaxation using Lagrangian multipliers [137] [139].

Riaz *et al.* implemented an SVM regression model to predict the future location of the tumor, and showed that the prediction performance of RMSE was less than 2mm at 1000ms latency [33]. However, the prediction error using machine learning increased monotonically with fewer data points in the training samples. In addition, initial model parameters at the beginning of a treatment required to be adjusted due to the pattern change of a patient respiration [33] [138]. That resulted in the high computational complexity and the slow response time of prediction [137].

www.manaraa.com

7) Hidden Markov model

A hidden Markov model (HMM) is a statistical probability model with invisible state transition, where states are not directly visible, but a particular state can generate one of observations based on observation probabilities [31]. In [31], state distributions of the finite state model (FSM) – irregular (IRR), exhale (EX), end-to-exhale (EOE), and inhale (IN) in three dimension [53] – are used to create HMM with transition state matrix (A) and current state probability (B) based on the fractional time of a particular breathing cycle. Each state is determined by the previous state, and is distinguished with velocity (v_i). We denote a_{ij} as the transition state probability from the present state i to the next state j, such that $\Sigma_j a_{ij}=1$, b_i as the current state probability to be calculated based on the time percent in a particular breathing cycle, such that $\Sigma_i b_i(t)=1$, as shown in Fig. 10 [31]. The transition probability in Fig. 10 assumes that there is no possibility of physical movement from EOE state to EX state, or from IN state to EOE state, and so on. To eliminate these transition elements, the transition state matrix can be expressed by replacing those values with zero, as follows:

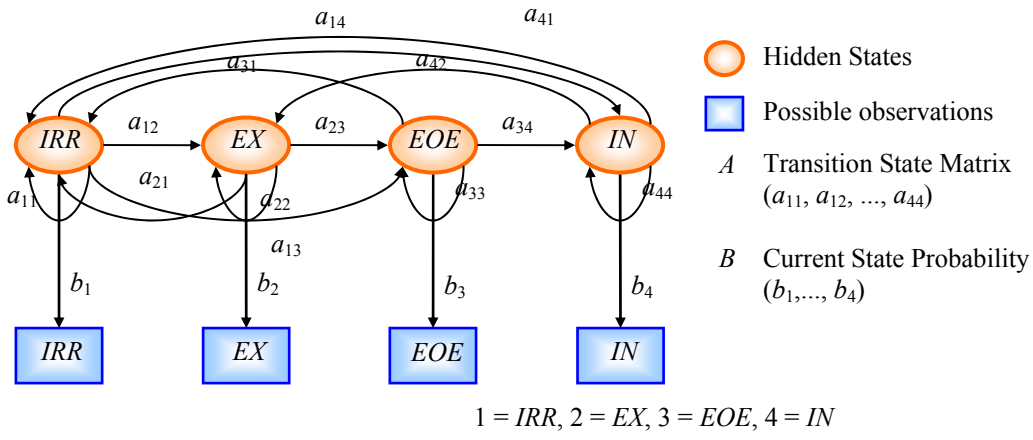

Figure 10. Probabilistic predictive model based on Hidden Markov model.
The transition state probability a_{ij} from the present state i to the next state j summarized to unity, such that $\Sigma_j a_{ij}=1$. The current state probability is calculated based on the time percent in a particular breathing cycle, such that $\Sigma_i b_i(t)=1$.

www.manaraa.com

$$A(t) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & 0 & a_{33} & a_{34} \\ a_{41} & a_{42} & 0 & a_{44} \end{pmatrix}, \quad (9)$$

To predict motion with HMM, the future position of an observation is calculated using the velocity parameter (v_i) based on FSM,

$$\hat{x}(t) = x(t-1) + \sum_l v_l \tau \,, \quad (10)$$

where variable τ (= 1/RT) consists of the sampling rate (R) and the estimated cycle period (T), and l represents the dimension. Kalet *et al*. showed that the RMSEs of ideal HMM and linear prediction are 1.88mm and 2.27mm with 200ms latency. The limitation of this model is that the implemented algorithm is based on stochastic process so that the prediction results can be different even with the same data [31]. We summarized the prediction accuracy and a representative feature for each method of the *model-based* approach, as shown in Table 2.

www.manaraa.com

Table 2. Model-based Prediction Algorithms of Respiratory Motion

Methods	Prediction error and Evaluation metrics	Features (System)
Linear Predictor [46]	Around 2.2mm with 200ms latency, RMSE	RMSE at 10 Hz (RTRT)
Kalman filter [46]	Around 2.5mm with 200ms latency, RMSE	RMSE at 10 Hz (RTRT)
Sinusoidal Model [37]	Less than 2mm with 200ms latency, Standard deviation	1- Dimensional prediction (RPM)
Finite state model [53]	Less than 1.5mm, RMSE	Three line segments (EX-EOE-IN) (RTRT)
Vector model based on tidal volume and airflow [126]	0.28–1.17mm	Standard deviation (Digital spirometer)
Patient-specific model using PCA [18]	Around 2–3mm Standard deviation	Respiration-correlated CT (RPM)
Autoregressive moving average model [47, 48]	0.8mm with 200ms latency, Standard deviation	Image rate: 1.25-10 Hz (RTRT, RPM)
Deformation from orbiting views [14]	2.5mm (LR), 1.7mm(SI) Standard Deviation	Cone-beam CT
Local regression method [87]	2.5mm	Local weighted regression, RMSE (RPM)
Optical flow deformable algorithm [38]	1.9mm	Standard deviation (Philips CT scanner)
Finite element method [34]	3mm (end expiration – end inspiration), 2mm (end expiration – midrespiration)	Patient-specific Models (Philips CT Scanner)
Surrogate-based Method [83]	2.2–2.4mm(carina), 3.7–3.9mm(diaphragm)	Standard Deviation (RPM)
Diaphragm-based Method [82]	2.1mm	Standard Deviation (RPM)
Support vector Regression Method [33]	Less than 2mm at 1000ms latency, RMSE	30 Hz sample frequency (CyberKnife)
Quaternion-based method [115]	2.5 (Standard Deviation)	Phantom Matching Error (PME)
Hidden Markov Model [31]	1.88ms at 200ms latency, RMSE	Various latency: 33 ms ~ 1000 ms (RTRT)
Kernel density estimation-based [32]	1.08mm at 160ms, 2.01mm at 570ms, RMSE	Multidimensional Prediction (CyberKnife)
Local circular motion model [12]	Less than 0.2 (nRMSE) at 200ms Normalized RMSE	First-order EKF, 5, 10, 15, 20 Hz (RPM)

2.3.2 Model-free Prediction Algorithms

Model-free heuristic learning algorithms, exemplified by linear adaptive filters and neural networks variables, can be used for the respiratory prediction for compensating for the impaired breathing signal with a variety of breathing patterns [8] [35] [36] [37]. These heuristic learning algorithms can adjust their coefficients/weights or configurations to

www.manaraa.com

reproduce newly arrived breathing signals without a priori models of signal history [8]. In this Chapter, we will explain two representative learning algorithms and adaptive systems for tumor prediction including 1) adaptive filters [3] [8] [35] [36] [37] [129], and 2) artificial neural network [8] [35] [36] [44] [46].

1) Adaptive Filters

An adaptive filter is a self-adaptive system that can adjust its coefficient values over time according to an optimization process incurred by an error signal, such as least mean squares (LMS) and recursive least squares (RLS) algorithms [130]. The adaptive filter depicted in Fig. 11 shows the basic adaptive filtering process for prediction.

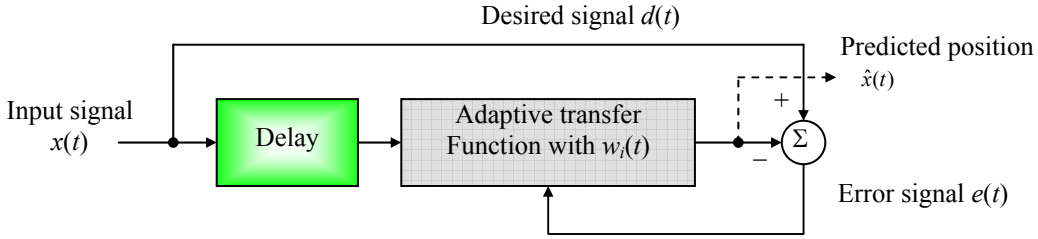

Figure 11. Basic adaptive filtering process for prediction.
The predicted position is calculated using the combination of previous respiratory motion $x(t\text{-}i)$ multiplied by its coefficient values $w_i(t)$. Here the coefficient values are time-variable according to an optimization process incurred by an error signal $e(t)$.

The predicted position $\hat{x}(t)$ can be expressed by a vector of previous respiratory motion $x(t-i)$ and a vector of filter coefficients $w_i(t)$, as follows:

$$\hat{x}(t) = \sum_{i=1}^{n} w_i(t)x(t-i)\,, \tag{11}$$

where filter coefficients change over time. Adaptive filters were widely used to predict the tumor motion [8] [35] [36] [37] [129]. Vedam *et al*. proved that adaptive filter models have the prediction accuracy with less than 2mm and outperform sinusoidal models [37]. Although the adaptive filter has a limitation with 1-dimensional prediction, it is extended

www.manaraa.com

into multi-dimensional adaptive filer [33]. Adaptive models can also be adjusted to update the weights of neural networks to improve the prediction accuracy [3] [35] [36].

2) Artificial Neural Network

An artificial neural network (ANN), commonly called neural network (NN), is a mathematical or computational function technique that is inspired by the biological neuron process [46]. A neural network consists of input, hidden, and output layers interconnected with directed weights (w), where we denote w_{ij} as the input-to-hidden layer weights at the hidden neuron j and w_{jk} as the hidden-to-output layer weights at the output neuron k, as shown in Fig. 12 [44] [46].

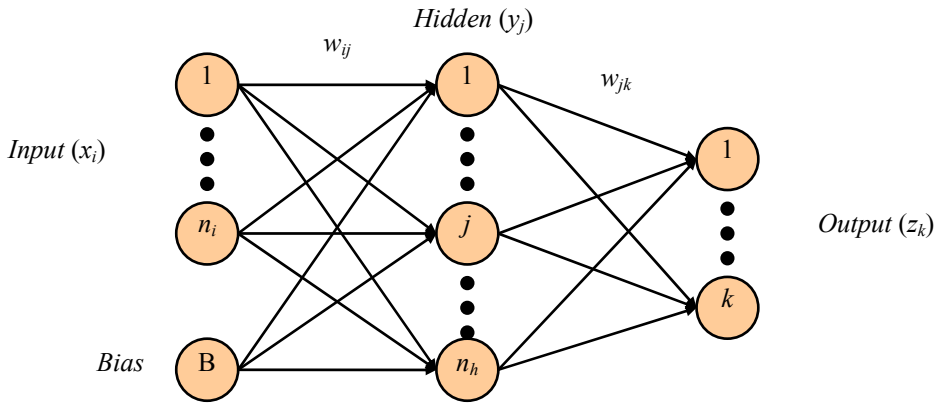

Figure 12. An artificial neural network with bias input and one hidden layer.
The network consists of input, hidden, and output layers interconnected with directed weights (w), where we denote w_{ij} as the input-to-hidden layer weights at the hidden neuron j and w_{jk} as the hidden-to-output layer weights at the output neuron k.

In Fig. 12, the input layer is a sequence history of breathing motions (n_i) with 3-dimensional positions. In the hidden layer, the intermediate value (y_j) is calculated with the history of breathing motions ($3n_i$) and bias unit using the nonlinear activation function, as follows [46]:

$$y_j = \frac{1}{1+\exp\left(-\sum_{i=1}^{3n_i+1} w_{ij} x_i\right)}, \tag{12}$$

www.manaraa.com

where we denote x_i as input values, and y_i as hidden values, respectively. The additional input unit (bias) is used to bias the linear portion of the computation. The practical prediction of respiratory motion is calculated with hidden values in the output neuron (z_k), as follows:

$$z_k = \sum_{j=1}^{n_h} w_{jk} y_j \ , \qquad (13)$$

where output values z_k denote predictions of breathing motions, and neural weights (w_{ij} and w_{jk}) in the network are generally resolved by numeric optimization. Sharp *et al*. showed that the RMSE of NN predictor is less than 2 mm with low latency (33 ms) [46]. But they only considered the form of stationary prediction.

For the adaptive filter training, Isaksson *et al.* used a feed-forward neural network with two input neurons and one output neuron using the least mean square scheme [8]. Here, the external markers were used as surrogates to predict the tumor motion. This two-layer feed-forward neural network was used for predicting irregular breathing pattern by Murphy *et al*. as well [35] [36] [44]. The network was trained by a signal history from the beginning of the patient data record using back-propagation algorithm, and kept updating the network weights with new test data samples to adjust newly arrived breathing signals [35] [36]. This adaptive filter showed much better prediction error than stationary filter, *e.g.*, RMSE of 0.5–0.7mm for the most predictable cases and of 1.4–1.7mm for the hardest cases with 200ms latency [36].

We summarized the prediction accuracy and a representative feature for each method of the *model-free* approach, as shown in Table 3.

www.manaraa.com

Table 3. Model-free Prediction Algorithms of Respiratory Motion

Methods	Prediction error and Evaluation metrics	Features (System)
Adaptive filter [37]	Less than 2mm with 200ms latency, Standard deviation	1-Dimensional prediction (RPM)
Artificial neural Networks [46]	Around 2.5mm with 200ms latency, RMSE	RMSE at 10 Hz (RTRT)
Adaptive neural network [8] [35] [36]	1.4–1.7mm with 200ms latency, Normalized RMSE	30 Hz sample Frequency (CyberKnife)

2.3.3 Hybrid Prediction Algorithms

Hybrid prediction algorithms used united methods to combine more than two methods or approaches to obtain outstanding results, compared to a previous solitary method. This method includes 1) adaptive neuro-fuzzy interference system (ANFIS) [40] [143], 2) hybrid model with adaptive filter and nonlinear model (Adaptive Tumor Tracking System) [41] [141], and 3) interacting multiple model (IMM) filter [4] [12] [45].

1) Adaptive Neuro-Fuzzy Inference System

A adaptive neuro-fuzzy inference system (ANFIS) is a hybrid intelligent system with combining both learning capabilities of a neural network and fuzzy logic reasoning, to find a specific model in association with input breathing motion and target prediction. The proposed neuro-fuzzy model ANFIS in [40] is a multilayer neural network-based fuzzy system in combination with two layers of adaptive nodes (layer 1 and 4) and three layers of fixed nodes (layer 2, 3, and 5), as shown in Fig, 9 [40].

www.manaraa.com

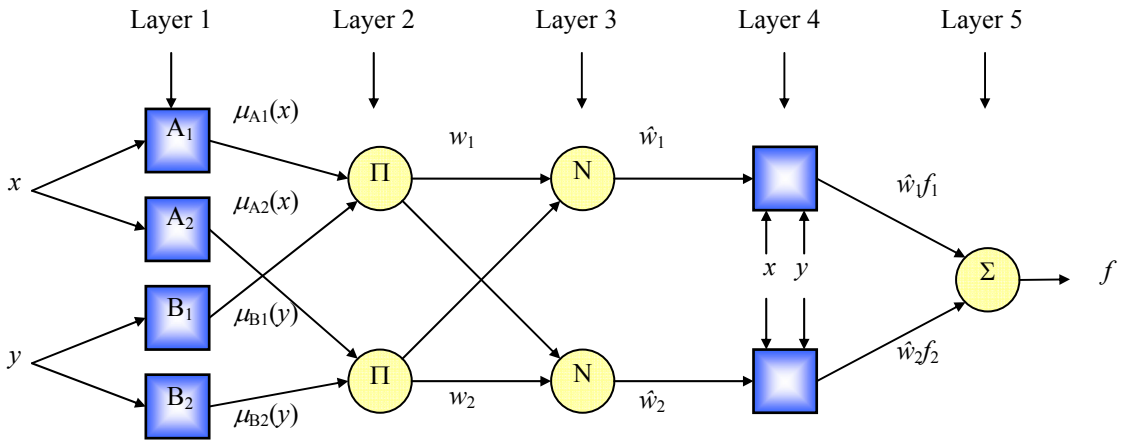

Figure 13. Adaptive Neuro Fuzzy Inference System with the total five layers.
Based on the incoming elements (x and y), this system is composed with two layers of adaptive nodes (layers 1 and 4) and three layers of fixed nodes (layers 2, 3, and 5). The layer 1 is characterized by a membership function $\mu(\cdot)$ that assigns each incoming element to a value between 0 and 1. Layer 4 is trained by a least squares method.

The first layer is distinguished by a fuzzy set (A_1, A_2, B_1, B_2) that is expressed by a membership function to assign each incoming element to a membership value between 0 and 1, as the following equation:

$$\mu_U(I;a_i,b_i,c_i)=\frac{1}{1+\left|\frac{I-c_i}{a_i}\right|2b_i},\quad A,B\in U,\quad x,y\in I,\quad i=1,2, \tag{14}$$

where I (x and y) are incoming elements, and three parameters (a_i, b_i, c_i) (referred to as premise parameters) are continuously updated by training samples using a gradient descent method [40] [143]. Each node in the second layer is a fixed node, characterized by the product (Π) of all the incoming signals, such as $w_i = \mu_{Ai}(x)\cdot\mu_{Bi}(y)$, i = 1, 2. Each node in the third layer is a fixed node, characterized by the normalized ratio (N), such as $\hat{w}_i = w_i/(w_1+w_2)$, i = 1, 2. Each node in the fourth layer is an adaptive node with a node function, such as $\hat{w}_i f_i = (p_i x + q_i y + r_i)$, i = 1, 2, where the parameter set (p_i, q_i, r_i) (referred to as consequent parameters) are trained by a least squares method. The single node in the last layer calculates the overall output by aggregating all incoming signals, such as $f = \Sigma_i$

www.manaraa.com

$\hat{w}_i f_i$, i = 1, 2. Kakar *et al*. validated that the prediction accuracy (RMSE) of respiratory motion for breast cancer patients was 0.628mm with coached breathing and 1.798mm with free breathing. This method required simpler and fewer remodeling decorations to implement its nonlinear ability in comparison to neural networks. However, for other conditions, exemplified by lung patients and respiration monitoring using spirometry or abdominal straps, it should associate the breathing signal with the target motion [40].

2) Hybrid Model with Adaptive Filter and Nonlinear Model

To compensate breathing tumor motion in the lung, an adaptive tumor-tracking system (ATTS) was proposed by Ma *et al*. with an adaptive filter and a nonlinear method [141]. Instead of only one signal, this adaptive system used two independent signals to detect the lung tumor motion during irradiation: 1) direct signal, *i.e.*, imaging of irradiated region using megavoltage imaging of the treatment beam [147], and 2) indirect signal, *i.e.*, optical marker with an infrared camera, as shown in Fig. 14 [41] [141]. The tumor position is directly visualized and located by the acquired portal image (direct signal) using a tumor tracking algorithm without internal fiducial markers [147]. Infrared camera signals (indirect signal) are used to predict respiratory signals using the adaptive filter, and these respiratory signals are correlated with the portal image to predict the tumor motion. A nonlinear dynamic system is reconstructed by the system history based on the previous measurement [141].

www.manaraa.com

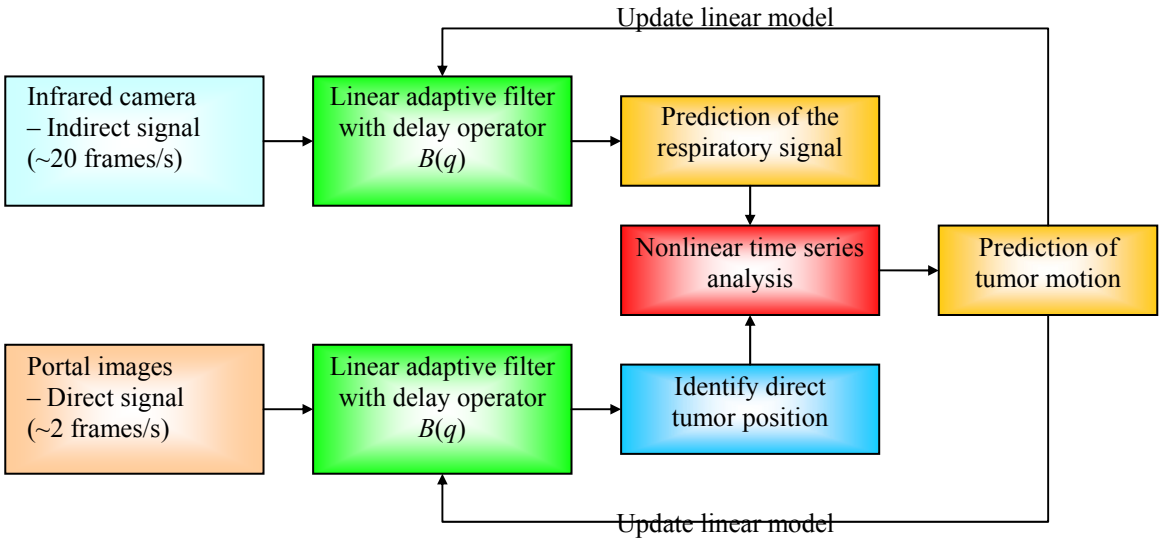

Figure 14. Adaptive tumor tracking system with two independent signals.
The tumor position is directly visualized and located by the acquired portal image (direct signal) using a tumor tracking algorithm without internal fiducial markers [41] [147]. Infrared camera signals (indirect signal) are used to predict respiratory signals using the adaptive filter, and these respiratory signals are correlated with the portal image to predict the tumor motion [141].

The adaptive filter continuously updated the coefficient parameters using least mean square method to predict the respiratory motion, as follows:

$$y(t) = B(q)u(t)\,, \tag{15}$$

where $y(t)$ is prediction of the respiratory motion, $B(q)$ is a linear model including the delay operator q with $B(q)=b_0q^0+b_1q^{-1}+\cdots+b_{n-1}q^{-n+1}$, and $u(t)$ is the history information including the past n samples of the infrared camera. In addition, ATTS modeled the correlation between two signals using means of nonlinear methods to determine the tumor position. That means dynamic nonlinear system examines the current indirect signal in the past samples using x(and y)-coordinate motion range (mm), maximum velocity of x(and y)-coordinate (mm/s), and mean cycle period (s) and then the best-fitting direct signals were adapted to predict the tumor motion [41]. Wilbert *et al*. showed that the maximum standard deviation was 0.8mm for x-coordinate and 1.0mm for y-coordinate. However, there are limits in velocity range between 8.5mm/s (y(and z)-

www.manaraa.com

coordinate) and 9.5mm/s (x-coordinate), so that the amplitude acquired below these limits will not lead to efficient prediction with such a linear model [41].

3) Interacting Multiple Model Filter

An interacting multiple model (IMM) filter can be used as a suboptimal hybrid filter for respiratory motion prediction to combine different filter models with improved control of filter divergence [4] [12] [45]. It makes the overall filter recursive by modifying the initial state vector and covariance of each filter through a probability weighted mixing of all the model states and probabilities, as shown in Fig. 15. [4] [45].

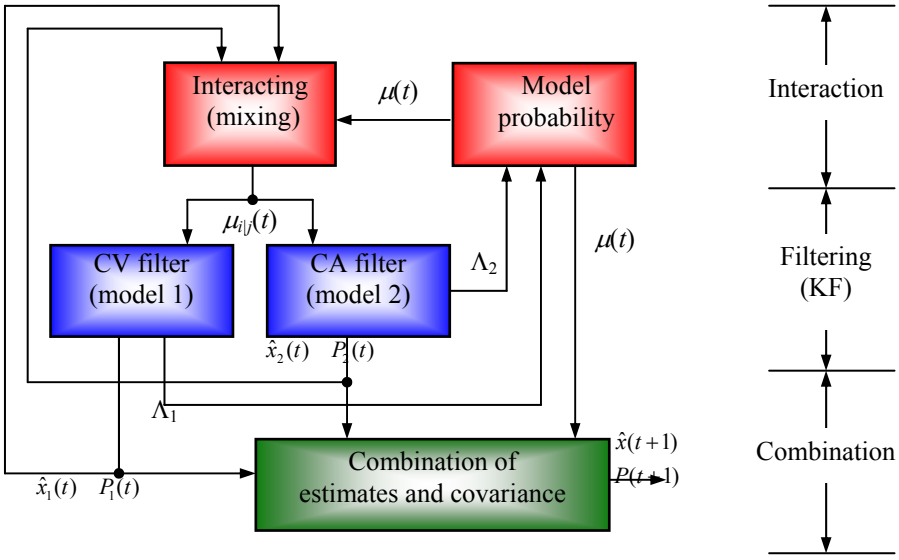

Figure 15. An interactive multiple model for respiratory motion prediction.
In the interaction step, model and mixing probabilities are initialized and updated. In the filtering step, the mixed filtering prediction (x_i) of target position and the associated covariance (P_i) are updated within each model. In the combination step, the actual prediction of target position is computed for output purposes with the mixing probability.

Fig. 15 shows a recursive filter of IMM with a constant velocity (CV) model and a constant acceleration (CA) model, where three steps – interaction, filtering, and combination – are repeated by each time instant t. In the interaction step, model probability ($\mu_j(t)$) and mixing probability ($\mu_{i|j}(t)$) are initialized and updated based on a 2×2 Markovian transition matrix (Π) with its component π_{ij} that represents the transition

www.manaraa.com

probability from model i to model j, satisfied with $\Sigma_j \pi_{ij} = 1$ for $i = 1, 2$, as follows [4] [45]:

$$\mu_j(t) = \sum_{i=1}^{2} \pi_{ij} \mu_i(t-1), \quad \mu_{i|j}(t) = \pi_{ij} \mu_i(t-1) / \mu_j(t), \tag{16}$$

where we denote $\mu_j(t)$ as the predicted probability for model j at time step t, and $\mu_{i|j}(t)$ as the weight for the conditional transition probability from model i for the previous time step $t-1$ to model j for the current time step t. In the filtering step, the mixed filtering prediction of target position ($\hat{x}_j(t)$) and the associated covariance ($P_j(t)$) are updated with Kalman gain, likelihood update (Λ_j) and model probability ($\mu_j(t)$), shown in Fig. 15 [45]. In combination step, the actual prediction of target position, *i.e.*, combination of estimates and covariance, is computed for output purposes with the mixing probability, such as estimation $\hat{x}(t+1) = \Sigma_j \hat{x}_j(t+1)\mu_j(t)$, and covariance $P(t+1) = \Sigma_j \{P_j(t)+[\hat{x}_j(t)- \hat{x}(t)][\hat{x}_j(t)- \hat{x}(t)]^T\}\mu_j(t)$ [4].

Putra *et al*. showed that the prediction of IMM filter was better than the prediction of the Kalman filters with CV and CA model, and that the errors of the IMM filter were less than 0.98mm with 200ms latency [45]. The limitation of this method is that the above hybrid method was proposed for dynamic iteration in one dimensional prediction, so that independent parallel filters should be implemented for 3-dimensional motions [4]. Furthermore, IMM method was investigated to compare with a prediction method based on the first-order extended Kalman filter by Hong *et al*. [12]. Breathing variation, such as deep or fast breathing, results in a relatively low accuracy of breathing motion prediction. King *et al.* showed that a multiple sub-model method based on breathing amplitude can provide an adaptive motion model with adjusting basic sub-models [100]. They validated

www.manaraa.com

that the combined models with multiple sub-models can show the prediction errors of 1.0–2.8mm.

4) Hybrid Extended Kalman Filter (HEKF)

Kalman filters are widely used for training nonlinear function of the state estimation and prediction for desired input-output mappings [4] [12] [45]. Kalman filter can also be used for supervised training framework of recurrent neural networks using nonlinear sequential state estimators. The prediction and correction property is an intrinsic property of Kalman filter. In Hybrid Extended Kalman filter (HEKF), recurrent neural network (RNN) performs a role of the predictor with network nonlinear function including input vector (u), recurrent network activities (v), and adaptive weight state vectors (w), whereas EKF performs a role of the corrector with innovation process in a recursive manner, as shown in Fig. 16 [145] [146].

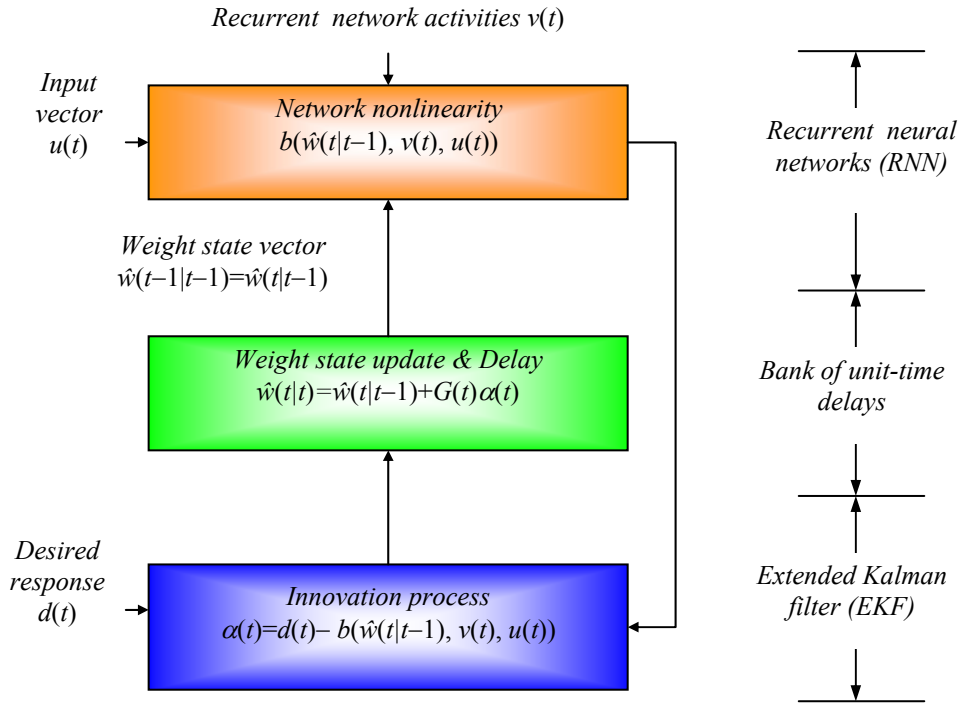

Figure 16. Closed-loop feedback system incorporating EKF for RNN.
RNN performs a role of the predictor with network nonlinear function, whereas EKF performs a role of the corrector with innovation process in a recursive manner in this system.

www.manaraa.com

The recurrent network is expressed by the network nonlinearity function $b(\cdot,\cdot,\cdot)$ with input vectors $u(t)$, the internal state of the recurrent network activities $v(t)$, and the weight state vector $\hat{w}(t|t-1)$. The innovation process $\alpha(t)$ of EKF is expressed as follows:

$$\alpha(t) = d(t) - b(\hat{w}(t \mid t-1), v(t), u(t)), \tag{17}$$

where $b(\cdot,\cdot,\cdot)$ is the network nonlinear function of vector-value measurement. The weight state vector is updated with the Kalman gain $G(t)$ and the innovation process [146].

Puskorius *et al*. proposed a Decoupled EKF (DEKF) as a practical solution for the computational resource management of covariance value with EKF for RNN [145]. Suk *et al*. applied DEKF to the prediction of respiratory motion. They evaluated that the prediction accuracy of the proposed HEKF and DEKF were less than 0.15 and 0.18 (nRMSE) with 200ms latency, respectively. They also validated that HEKF can improve the average prediction overshoot more than 60%, compared with DEKF. This method comprehensively organized the multiple breathing signals with adapting the coupling technique to compensate the computational accuracy, whereas the computational requirements were increased to improve the prediction accuracy [7]. We summarized the prediction accuracy and a representative feature for each method of the hybrid approach, as shown in Table 4.

Table 4. Hybrid Prediction Algorithms of Respiratory Motion

Methods	Prediction error and Evaluation metrics	Features (System)
Adaptive neuro-fuzzy inference system [40]	0.628mm (coached), 1.798mm (non-coached), RMSE	25 Hz sample Frequency (RPM)
Adaptive tumor tracking system [41]	0.8mm (x-max), 1.0mm (y-max), Standard deviation	Megavoltage imaging with infrared system (ELEKTA)
Interacting multiple	0.98mm with 200ms latency for 5Hz, RMSE	Kalman CV and CA,

www.manaraa.com

model filter [45]		Markovian transition (RPM)
Adaptive Motion Model [100]	1.0–2.8mm	Standard deviation
Hybrid extended Kalman filter [7]	Less than 0.15 with 200ms latency, Normalized RMSE	26 Hz sample frequency (CyberKnife)

www.manaraa.com

2.4 Open Questions for Prediction of Respiratory Motion

Variable open questions on the prediction of respiratory motion are still remained to be solved in a foreseeable future. In this Chapter, we will point out general open questions for the advanced radiotherapy technology, but open issues are not limited to the following issues described in this study.

2.4.1 Changes of Respiratory Patterns

The respiratory patterns identified in the treatment preparation may be changed before or during the treatment delivery. A real-time tracking method may compensate for changes of respiratory pattern during treatment delivery, but this method can be interrupted by other parameters, *e.g.*, cardiac and gastrointestinal motion, baseline shifts, tumor deformation, highly fluctuating amplitudes of respiratory motion, and so on [1]. Therefore, it requires clinical solutions to adjust or construct changes of respiratory patterns.

2.4.2 Tumor Deformation and Target Dosimetry

Lung deformation derived from respiration may change tumor shapes, or a tumor may change its own shape by itself [150]. Some studies investigated that irregular breathing patterns required more extended clinical target volume compared with regular breathing patterns [3]. Sophisticated target dosimetry based on tumor deformation also should be considered for the optimized treatment delivery.

www.manaraa.com

2.4.3 Irregular Pattern Detection

A real-time tumor-tracking method, where the prediction of irregularities really becomes relevant [35], has yet to be clinically established. In the thoracic radiotherapy, other parameters including cardiac and gastrointestinal motion can affect the prediction of respiratory patterns. Respiratory patterns of some patients may have dramatically irregular motions of peaks and valleys position, compared with others [148]. It requires a new strategy or standard for irregular breathing classification depending on a degree of breathing irregularity for each patient. Irregular pattern detection may be used to adjust a margin value, *e.g.*, the patients assigned with regular patterns would be dealt with tight margins to prevent health tissues from irradiating by high-dose treatment. For the patients assigned with irregular patterns, safety margins should be determined by patient-specific irregularity to compensate for the baseline shifts or highly fluctuating amplitudes that are not covered by standard safety margins [3] [149].

www.manaraa.com

2.5 SUMMARY

In this Chapter, we have showed current radiotherapy technologies including tools for measuring target position during radiotherapy and tracking-based delivery systems including Linacs, MLC, and robotic couch. We have also explained three prediction approaches including *model-based*, *model-free*, and *hybrid* prediction algorithms. In the previous Chapter, we have described some questions that still remain to be solved in the future, exemplified by changes of respiratory patterns, tumor deformation target dosimetry, and irregular pattern detection. Open questions are not limited to the issues described in the study.

www.manaraa.com

CHAPTER 3 PHANTOM: PREDICTION OF HUMAN MOTION WITH DISTRIBUTED BODY SENSORS

Tracking human motion with distributed body sensors has the potential to promote a large number of applications such as health care, medical monitoring, and sports medicine. In distributed sensory systems, the system architecture and data processing cannot perform the expected outcomes because of the limitations of data association. For the collaborative and complementary applications of motion tracking (Polhemus Liberty AC magnetic tracker), we propose a distributed sensory system with multi-channel interacting multiple model estimator (MC-IMME). To figure out interactive relationships among distributed sensors, we used a Gaussian mixture model (GMM) for clustering. With a collaborative grouping method based on GMM and expectation-maximization (EM) algorithm for distributed sensors, we can estimate the interactive relationship of multiple sensor channels and achieve the efficient target estimation to employ a tracking relationship within a cluster. Using multiple models with improved control of filter divergence, the proposed MC-IMME can achieve the efficient estimation of the measurement as well as the velocity from measured datasets with distributed sensory data. We have newly developed MC-IMME to improve overall performance with a Markov switch probability and a proper grouping method. The experiment results showed that the prediction overshoot error can be improved in the average 19.31% with employing a tracking relationship.

www.manaraa.com

3.1 INTRODUCTION

Prediction human motion with distributed body sensors has the potential to improve the quality of human life and to promote a large number of application areas such as health care, medical monitoring, and sports medicine [7] [160] [161]. The information provided by distributed body sensors are expected to be more accurate than the information provided by a single sensor [7] [162]. In distributed sensory systems, however, the system architecture and data processing cannot perform the expected outcomes because of the limitations of data association [163] [164] [165] [166] [167] [168] [169] [170] [171] [172]. As shown in Fig. 17, individual sensory system using IMME shows the position estimate values of benign motion for the human chest. The typical problem showed in this figure is that the prediction overshoots at the beginning of tracking estimation can result in a significant prediction error. This initial estimate error has motivated us to develop an appropriate method that would reduce the initial prediction estimate error. Therefore, we propose a new method to reduce the initial prediction estimate error by employing a tracking relationship of data association [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183].

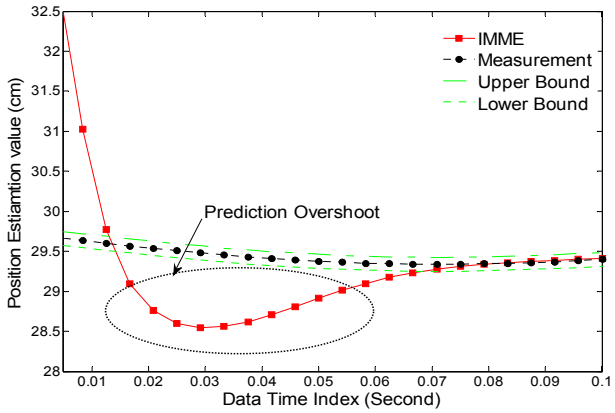

Figure 17. Prediction Overshoot of IMME.
This figure shows the position estimation of benign motion for the human chest. The upper bound and lower bound can be derived from adding the marginal value to the measurement and subtracting the marginal value from the measurement, respectively [26].

www.manaraa.com

As a unique solution to prevent significant prediction overshoots from initial estimate error, we adopt multiple sensory systems with grouping method based on GMM for clustering. Clustering is a method that enables a group to assign a set of distributed sensors into subsets so that distributed sensors in the subset are executed in a similar way. A variety of studies have been investigated for clustering methods based on *k*-means, spectral clustering, or expectation-maximization (EM) algorithm [160] [185] [186] [187] [188] [189] [190] [191] [192] [193] [194] [195] [196]. However, a known limitation of these clustering methods is that the cluster number must be predetermined and fixed. Recently, Bayesian nonparametric methods with Dirichlet process mixture have become popular to model the unknown density of the state and measurement noise [197] [198]. But, because of the relatively small set of samples, it will not adequately reflect the characteristics of the cluster structure [188]. For the time sequential datasets of distributed body sensors, we would like to add a *prior* distribution on the cluster association probability [199] [200]. We refer to this *prior* information as hyper-parameters [199]. Therefore, we proposed a new collaborative grouping method for distributed body sensors.

Multiple models (MM) may have multiple possible dynamic models for multi-sensor systems with Markov model switches. In such a hybrid system, the possible models make multiple sensors supply the information about the interested variable, and thus are collaborative and complementary. The basic idea of all MM approaches is that complicated target movements are made up of random variations originating from basic (straight-line) target motion. Due to the difficulty in representing this motion simply with a single model, MMs including potentially dynamic models operate in a parallel way with

www.manaraa.com

Markov switch probability [173]. The proposed solution is to employ a tracking relationship among distributed body sensors by adding switching probability for multiple models and grouping method to figure out the interactive relation within the sensors. IMME algorithm can be used to combine different filter models with improved control of filter divergence. As a suboptimal hybrid filter [173] [174] [175], IMME makes the overall filter recursive by modifying the initial state vector and covariance of each filter through a probability weighted mixing of all the model states and probabilities [176] [177] [178] [179] [180] [181] [182] [183].

The overall contribution of this research is to minimize the prediction overshoot originating from the initialization process by newly proposed Multi-channel IMME (MC-IMME) algorithm with the interactive tracking estimation. MC-IMME can estimate the object location as well as the velocity from measured datasets using multiple sensory channels. For this MC-IMME, we have extended the IMME to improve overall performance by adding switching probability to represent the conditional transition probability and a collaborative grouping method to select a proper group number based on the given dataset. The technical contributions of this study are twofold: First, we propose a cluster number selection method for distributed body sensors based on Dirichlet hyper-prior on the cluster assignment probabilities. Second, we present a new prediction method to reduce the initial estimate error by employing a tracking relationship among distributed sensory data. For the performance improvement, we added switching probability to represent the conditional transition probability from a previous channel state to a current channel state and a collaborative transition probability to select a proper group number based on the given datasets.

www.manaraa.com

This Chapter is organized as follows. In Chapter 3.2, the theoretical background for the proposed algorithm is briefly discussed. In Chapters 3.3 and 3.4, the proposed grouping criteria with distributed sensors placement based on *EM* algorithm and the proposed estimate system design for distributed body sensors are presented in detail, respectively. Chapter 3.5 presents and discusses experimental results of proposed methods—grouping methods and adaptive filter design. A summary of the performance of the proposed method is presented in Chapter 3.6.

www.manaraa.com

3.2 Related Work

3.2.1 Kalman Filter

The Kalman filter (KF) provides a general solution to the recursive minimized mean square estimation problem within the class of linear estimators [201] [202]. Use of the Kalman filter will minimize the mean squared error as long as the target dynamics and the measurement noise are accurately modeled. Consider a discrete-time linear dynamic system with additive white Gaussian noise that models unpredictable disturbances. The problem formulation of dynamic and the measurement equation are as follows,

$$\begin{aligned} x(k+1) &= F_i(k)x(k) + G_i(k)u(k) + v_i(k), \\ z(k) &= H_i(k)x(k) + w_i(k) \end{aligned} \tag{18}$$

where $x(k)$ is the n-dimensional state vector and $u(k)$ is an n-dimensional known vector (which is not used in our application). The subscript i denotes quantities attributed to model M_i. $v(k)$ and $w(k)$ are process noise and measurement noise with the property of the zero-mean white Gaussian noise with covariance, $E[v(k)v(k)^T] = Q(k)$ and $E[w(k)w(k)^T] = R(k)$, respectively. The matrices F, G, H, Q, and R are assumed known and possibly time-varying. That means that the system can be time-varying and the noise non-stationary.

The Kalman filter estimates a process by using a form of feedback control. So the equations for the Kalman filter divide into two groups: time update equations and measurement update equations. The estimation algorithm starts with the initial estimate $\hat{x}(0)$ of $x(0)$ and associated initial covariance $P(0)$. The problem formulation of the predicted state and the state prediction covariance can be written as:

$$\begin{aligned} \hat{x}(k+1) &= F(k)\hat{x}(k) + G(k)u(k), \\ P(k+1) &= F(k)P(k)F(k)^T + Q(k). \end{aligned} \tag{19}$$

www.manaraa.com

For the proposed MC-IMME, we use Eqs. (18) and (19) with a different model of filters, *i.e.*, a constant velocity model and a constant acceleration model.

3.2.2 Interacting Multiple Model Framework

Multiple model algorithms can be divided into three generations: autonomous multiple models (AMM), cooperating multiple models (CMM), and variable structure multi-models (VSMM) [172] [173]. The AMM algorithm uses a fixed number of motion models operating autonomously. The AMM output estimate is typically computed as a weighted average of the filter estimates. The CMM algorithm improves on AMM by allowing the individual filters to cooperate. The VSMM algorithm has a variable group of models cooperating with each other. The VSMM algorithm can add or delete models based on performance, eliminating poorly performing ones and adding candidates for improved estimation. The well-known IMME algorithm is part of the CMM generation [173].

The main feature of the interacting multiple model (IMM) is the ability to estimate the state of a dynamic system with several behavior models. For the IMM algorithm, we have implemented two different models based on Kalman filter (KF): 1) a constant velocity (CV) filter in which we use the direct discrete-time kinematic models, and 2) a constant acceleration (CA) filter in which the third-order state equation is used [181] [182] [183] 201] [203] [204] [205]. The IMME is separated into four distinct steps: interaction, filtering, mode probability update, and combination [201]. Fig. 18 depicts a two-filter IMM estimator, where $\hat{x}$ is the system state, P is the filter estimate probability, z is the measurement data, and μ are mixing probabilities. Note that the previous state of each

www.manaraa.com

model is reinitialized by the interaction stage each time the filter iterates. In IMME, at time k the state estimate is computed under each possible current model using CV or CA.

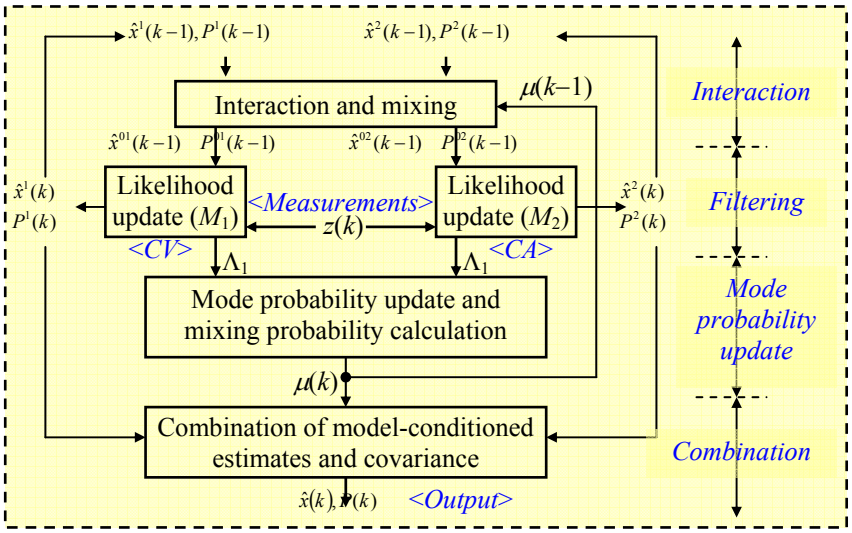

Figure 18. Interacting Multiple Model Estimator
The IMME has a four-step process in a way that different state models are combined into a single estimator to improve performance.

In Fig. 18, the mixing probability (μ_{ij}) represents the conditional transition probability from state i to state j. With an initial state of each model ($\hat{x}^i(k-1)$), new filter state is computed to estimate the mixed initial condition ($\hat{x}^{0i}(k-1)$) and the associated covariance ($P^{0i}(k-1)$) according to the mixing probability. The above estimates and the covariance are used as input to the likelihood update matched to $M_j(k)$, which uses the measurement data ($z(k)$) to yield $\hat{x}^i(k)$ and $P^i(k)$. The likelihood function corresponding to each model i (Λ_i) is derived from the mixed initial condition ($\hat{x}^{0i}(k-1)$) and the associated covariance ($P^{0i}(k-1)$). After mode probability update based on a likelihood function (Λ_i), combination of the model-conditioned estimates and covariance is computed for output purposes with the mixing probability. For our distributed sensory system of target estimation, each filter state of IMM is dedicated for each sensor, and distributed target estimations independently progress according to each IMME.

www.manaraa.com

3.2.3 Cluster Number Selection using Gaussian Mixture Model (GMM) and Expectation-Maximization (EM) Algorithm

For industrial applications of motion tracking, distributed body sensors placed on target surface with different positions and angles can have specific correlation with others. That means distributed body sensors can cooperate with each other as a group with clustering. Recently, several clustering algorithms have been developed to partition the observations (L) into several subsets (G) [185] [186] [187] [188] [189] [190] [191] [192] [193] [194] [195] [196]. The most notable approaches are a mean square error (MSE) clustering and a model-based approach. The MSE clustering typically is performed by the well-known *k*-means clustering. In general, *k*-means clustering problem is *NP*-hard [185], so a number of heuristic algorithms are generally used [191] [193] [194].

A model-based approach to deal with the clustering problem consists of certain models, *e.g.*, a Gaussian or a Poisson model for clusters and attempting to optimize the fit between the data and the model. The most widely used clustering method of this kind is a Gaussian mixture model (GMM) [188] [189] [190]. In GMM, the joint probability density that consists of the mixture of Gaussians $\phi(z; m_y, \sum_y)$, where y=1...G, should be solved [187] [188]. Assume a training set of independent and identically distributed points sampled from the mixture, and our task is to estimate the parameters, *i.e.*, *prior* probability (α_y), mean (m_y) and covariance ($\sum_y$) of the clustering components (G) that maximize the log-likelihood function $\delta(\cdot)$ based on EM algorithm [188] [190]. Given an initial estimation (α_0, m_0, $\sum_0$), EM algorithm calculates the *posterior* probability $p(y|z_j)$ in E-step. Based on the estimated result we can calculate the *prior* probability (α_y), mean

www.manaraa.com

(m_y) and covariance ($\sum_y$) for the next iteration, respectively, in the following M-step [206] [207] [208].

The log-likelihood for the incomplete datasets in the EM algorithm can never be decreased (see Chapter 1.6 in [209]), because the EM algorithm iterates the computations E-step and M-step until the convergence to a local maximum of the likelihood function. That means that the consecutive log-likelihood functions monotonically increase and could be very similar but not identical. We define the discrepancy of consecutive values as the difference (Δ). Now, we can define the difference (Δ) as follows:

$$\Delta(G) \cong \delta(\Theta, G) - \delta(\Theta, G-1)\,. \tag{20}$$

Once we estimate the parameter $\Theta \equiv \{\alpha_y, m_y, \sum_y\}_{y=1}^{G}$ we can find the optimal cluster number (G*) with the conventional Brute-force search algorithm by introducing $\Delta(G)$ that is a log-likelihood function after parameter learning with the following equation: $G^* = \text{argmin}_g \Delta(G)$. In practice, we can set a threshold (Δ_{th}) that is almost closed to zero to save the redundant iteration step. We can start with $G = 2$ for a first candidate solution for cluster number selection, estimate the set of finite mixture model parameter $\Theta^* \equiv \{\alpha^*_y, m^*_y, \sum^*_y\}_{y=1}^{G}$ using EM algorithms based on the sample data, and calculate $\Delta(G)$. After checking whether a candidate G is an appropriate cluster number for L, we can use the cluster number G as an appropriate cluster number as shown in Fig. 19.

The search algorithm based on the log-likelihood function in Fig. 19 can only work in the static data model, but cannot guarantee to work in the time sequential data because of the lack of adaptive parameter for the time sequential data. Thus, it has the following two limitations: 1) it can only work within limitation of the initial dataset, and 2) it cannot guarantee the global optimal based on the time sequential data because of the lack of

www.manaraa.com

adaptive parameter for the time sequential data. To overcome such static grouping limitations, we introduce distributed grouping using the multi-channel (MC) selection for the time sequence data of distributed body sensors in the next Chapter.

```
Input : # sensor (L)
set G ← first(L);          // first(L) : generate a first candidate solution for cluster number selection
Temp = Infinity;
while ( G ≠ L)             // L : # sensors
{
        estimate Θ*;       // Θ* : the set of finite mixture model parameter
        CV = Δ(G);         // Δ(·) : discrepancy of consecutive log-likelihood functions
        if  valid(L, G)    // check whether candidate G is an appropriate cluster number for L
                then break
        else
                if Temp > CV
                        then Temp = CV; O_t = G;
        G ← G + 1                  // update the next candidate G for L
}
Output : cluster number (O_t)
```

Figure 19. Brute-force search algorithm to select the group number.
This figure shows the Brute-force search algorithm to select the group number (G) based on the log-likelihood function.

www.manaraa.com

3.3 Proposed Grouping Criteria with Distibuted Sensors

The distributed measurements can provide more reliable estimation of target tracking. The motivation of this Chapter is to prepare interactive relationships with distributed sensory data for clustering, *i.e.*, how to collaborate with distributed measurements to achieve better performances compared to the single measurement. In Chapter 3.3.1, we will show how to initialize the hyper-parameter presenting a hypothetical *prior* probability for background knowledge, and can select the collaborative cluster number using EM iteration. In Chapter 3.3.2, we will calculate switching probability representing the conditional transition probability from channel *a* to channel *b* within a cluster number.

3.3.1 Collaborative Grouping with Distributed Body Sensors

The cluster number selection using GMM works well in the distributed means model as well as in the static data model. But it only works within limitation of the initial dataset. The tracking estimate system with distributed body sensors has time sequential data. That means the measured information from each sensor can be changed depending on the applications from time to time. To make the collaborative grouping system, we introduce some background knowledge that can be presented as a hypothetical *prior* probability (β_y) that we call hyper-parameter [199] [200]. Suppose that $\alpha_y(k)$ is an initial *prior* probability at time k. The initial hyper-parameter β_y can be found as follows:

$$\beta_y(0) = \lim_{k\to\infty} \alpha_y(k) \tag{21}$$

In practice, we can get the hyper-parameter (β_y) using sample training data instead of the infinite training data with respect to time. After calculating the hyper-parameter with sample training data using (21), the hyper-parameter (β_y) should be adaptive with respect

www.manaraa.com

to time k. Please note that the hyper-parameter can be selected based on the global information of sample data. This parameter is selected for corresponding to the steady state. It can be accomplished using the switching probability that will be explained in detail in Chapter 3.4.3. The adaptive hyper-parameter can be increased or decreased based on the current switching probability comparing to the previous switching probability, and can be calculated as follows:

$$\beta_y(k) = \beta_y(k-1) + \Delta\mu^y, \tag{22}$$

where $\Delta\mu^y$ is the difference between the current switching probability and the previous one. $\Delta\mu^y$ can be calculated using a switching probability at time k, *i.e.*, $\mu^y(k)$ indicating the switching weight of group y. We will describe how to select the difference ($\Delta\mu^y$) in detail in Chapter 3.4.4. After calculating the adaptive hyper-parameter, the adaptive (ADT) *posterior* probability $p_{ADT}(y|z_j)$ is calculated at time k in E-step as follows:

$$p_{ADT}(y \mid z_j)(k) = \frac{\alpha_y^{(t)}\phi(z_j; m_y^{(t)}, \Sigma_y^{(t)}) + \beta_y(k)}{\sum_{y=1}^{G}\alpha_y^{(t)}\phi(z_j; m_y^{(t)}, \Sigma_y^{(t)}) + \sum_{y=1}^{G}\beta_y(k)} \tag{23}$$

Using the modified one, we can proceed to the M-step at time k as follows:

$$\begin{aligned}
\alpha_y^{(t+1)}(k) &= \frac{1}{L}\sum_{j=1}^{L} p_{ADT}(y \mid z_j)(k), \\
m_y^{(t+1)}(k) &= \frac{\sum_{j=1}^{L} p_{ADT}(y \mid z_j) z_j}{\sum_{j=1}^{L} p_{ADT}(y \mid z_j)} = \frac{1}{\alpha_y L}\sum_{j=1}^{L} p_{ADT}(y \mid z_j) z_j, \\
\Sigma_y^{(t+1)}(k) &= \frac{1}{\alpha_y L}\sum_{j=1}^{L} p_{ADT}(y \mid z_j)\left[(z_j - m_y^{(t+1)})(z_j - m_y^{(t+1)})^T\right]
\end{aligned} \tag{24}$$

We can estimate the t^{th} iteration result of the adaptive *posterior* probability $p_{ADT}(y|z_j)$ at time k from (23). Based on the modified result we can calculate the *prior* probability (α_y), the mean (m_y), and the covariance (Σ_y) in the $(t+1)^{th}$ iteration for the collaborative

www.manaraa.com

grouping for time sequential data, respectively, using (24). A local maximum at time k can be selected by iterating the above two steps. We can select the collaborative cluster number (G^*_{ADT}) by introducing $\Delta_{ADT}(G)$ that is a log-likelihood function after parameter learning with the following equations:

$$G^*_{ADT} = \arg\min_{G} \Delta_{ADT}(G),$$
$$\Delta_{ADT}(G) = \delta_{ADT}(\Theta^*, G) - \delta_{ADT}(\Theta^*, G-1) \tag{25}$$

where δ_{ADT} is a log-likelihood function with the adaptive *posterior* probability. Note that Eq. (20) is extended into Eq. (25) with the hyper-parameter (β_y). Comparing with the previous algorithm, the collaborative grouping with time sequential data can select local maxima at time k by iterating two steps: E-step and M-step. We can select the global optimal from a series of local maxima of time k, as shown in Fig. 20.

```
set βy;                          // hyper-parameter
set μy;                          // switching probability
calculate βy(k)                  // adaptive hyper-parameter
while ( G ≠ L))
{
        E-Step   : calculate pADT(y|zj);
        M-Step   : calculate αy, my, and Σy;
        calculate ΔADT(G);
        temp = ΔADT(G);
        if  ( minDelta ≥ temp)
                minDelta = temp;
}
```

Figure 20. Collaborative group number selection with the adaptive hyper-parameter

3.3.2 Estimated Parameters used for Interacting Multiple Model Estimator (IMME)

Collaborative grouping with time sequence data can select local maxima at time k using the difference (Δ) of the consecutive log-likelihood functions. We can set the difference

www.manaraa.com

(Δ) of the consecutive log-likelihood functions as Δ(G*, k) with respect to time k. To reduce the notation complexity, Δ(k) is simply used for Δ(G*, k). Now we can use this Δ(k) for IMME to estimate the multi-channel estimates and covariance. As mentioned, the log-likelihood function in each EM step cannot decrease [209]. That means we can minimize the difference (Δ(k)) of the consecutive log-likelihood functions with respect to time k because Δ(k) converges to zero over a period of time. Therefore, we can find out the following relationship: Δ(k–1) ≥ Δ(k). In the standard IMME, it is assumed that the mixing probability density is a normal distribution all the time. Now we derive the switching probability (μ_{ab}) for the estimated parameter from mixing probability (μ_{ij}). Since it is hard to get $\mu_{ab}(k\text{-}1)$ directly, we used a tractable estimation $\mu_{ab}(k\text{-}1)+\Delta(k\text{-}1)$, as follows:

$$\mu_{ab}(k-1) = \mu_{ij}(k-1) + \Delta(k-1), \tag{26}$$

where μ_{ij} is the mixing probability that represents the conditional transition probability from state i to state j, and μ_{ab} is the switching probability that represents the conditional transition probability from channel a to channel b. Note that we define mixing probability (μ_{ij}) as switching probability (μ_{ab}). That means our assumption is still valid in the switching probability, *i.e.*, switching probability density follows a normal distribution (see appendix). The equality of Eq. (26) is true because the value of Δ(k-1) can be zero as k goes to the infinity. We can use the right side of Eq. (26) to dynamically select the switching probability (μ_{ab}) with Δ(k). Eq. (26) above provides us with a method to design the filter for distributed sensors at the second stage, because the switching probability can

www.manaraa.com

be adjusted more dynamically based on $\Delta(k)$ in the second stage filter. We will explain how to estimate the MC estimates and covariance using (26) in the next Chapter.

www.manaraa.com

3.4 SENSORS MULTI-CHANNEL (MC) IMME: PROPOSED SYSTEM DESIGN

The proposed method with collaborative grouping for distributed sensory data can achieve the efficient target estimation by using geometric relationships of target information emerging from distributed measurements. Fig. 21 shows a general block diagram to represent the relationship between the proposed method (MC-IMME) and IMME.

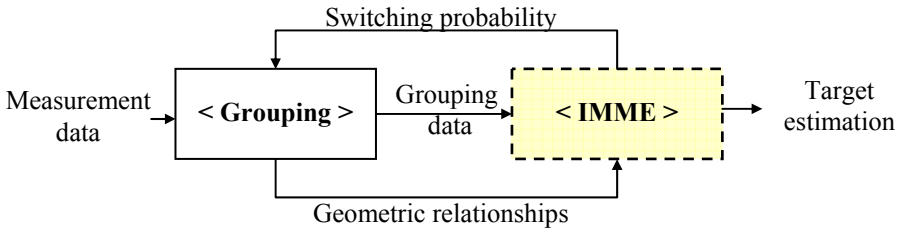

Figure 21. General block diagram for the proposed MC-IMME.

In MC-IMME, grouping data can be used for target-tracking estimation with IMME. Geometric information of distributed measurements is used for the switching probability update in the target estimation. Even though the proposed method needs the initialization process that is the same as in the IMME prediction, the interactive relationship with distributed sensors can compensate for the prediction estimate error. For the interactive tracking estimate, the proposed system design herein can be extended from Fig. 18 to Fig. 22.

www.manaraa.com

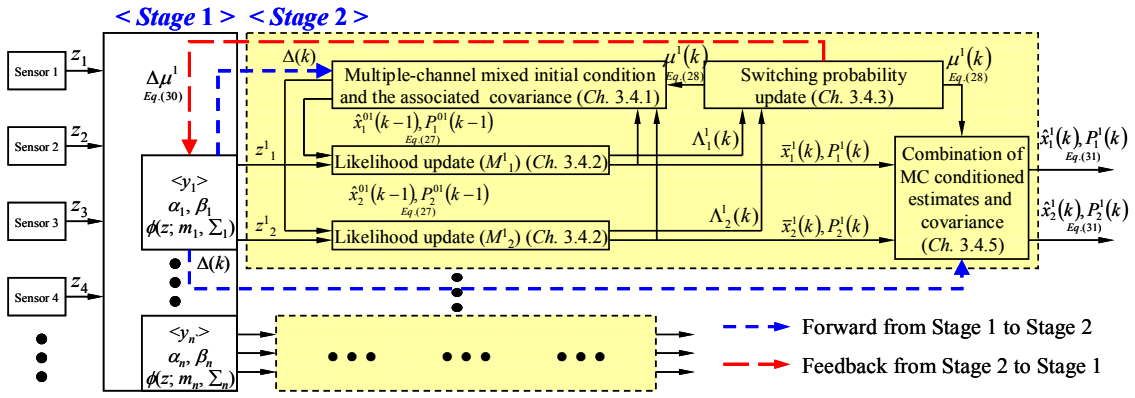

Figure 22. System design for distributed body sensors has two stages.

At the first stage, all the distributed sensors are partitioned into the groups that have a tracking relationship with each other. At the second stage, the interactive tracking estimate is performed for distributed groups.

3.4.1 MC Mixed Initial Condition and the Associated Covariance

Starting with an initial $\bar{x}_a^y(k\text{–}1)$ for each channel a in a group y, new filter state is computed to estimate the mixed initial condition and Kalman filter covariance matrices (27) according to the relationships

$$\hat{x}_a^{0y}(k-1)=\sum_{a=1}^{r}\bar{x}_a^y(k-1)[\mu_{ab}^y(k-1)],$$
$$P_a^{0y}(k-1)=\sum_{a=1}^{r}[\mu_{ab}^y(k-1)][P_a^y(k-1)+DP_{ab}^y(k-1)] \quad (27)$$

where μ_{ab}^y is a switching probability presenting the relationship between channel a and channel b within the same group y. As shown in Fig. 22, we have added the blue line indicating how the difference ($\Delta(k)$) in Stage 1 would be used for Stage 2. We denote r as the channel number of the group and $DP_{ab}^y(k\text{-}1)$ as an increment to the covariance matrix to account for the difference in the state estimates from channels a and b, expressed by $[\bar{x}_a^y(k\text{–}1) - \bar{x}_b^y(k\text{–}1|)]\cdot[\bar{x}_a^y(k\text{–}1) - \bar{x}_b^y(k\text{–}1)]^T$.

Note that the initial states of IMME are extended into Eq. (27) incorporating with the switching probability and $\Delta(k\text{–}1)$. We have adopted the results of Chapter 3.3 on

www.manaraa.com

grouping criteria $\Delta(k)$ of Eq. (26). The difference ($\Delta(k)$) can be minimized with respect to time k, so we can adjust to estimate the filter state from a coarse probability to a dense probability.

3.4.2 MC LIKELIHOOD UPDATE

The above estimates and covariance are used as input to the filter matched to $M_a^y(k)$, which uses $z_a^y(k)$ to yield $\hat{x}_a^y(k)$ and $P_a^y(k)$. The likelihood functions corresponding to each channel are computed using the mixed initial condition and the associated covariance matrix (27) as follows: $\Lambda_a^y(k)=p[z(k)|\ M_a^y(k),\ \hat{x}_a^{0y}(k-1),\ P_a^{0y}(k-1)]$, where y is a group number and $r(y)$ is the number of sensors for each group y. To reduce the notation complexity, r is simply used for $r(y)$.

3.4.3 SWITCHING PROBABILITY UPDATE

Given the likelihood function ($\Lambda_a^y(k)$) corresponding to each channel, the switching probability update is done as follows:

$$\mu^y(k)=\frac{1}{c_y}\Lambda^y(k)\bar{c}_y,\quad y=1,...,G,\quad \Lambda^y=\frac{1}{r}\sum_{a=1}^{r}\Lambda_a^y,\quad c_y=\sum_{y=1}^{G}\Lambda^y\bar{c}_y, \tag{28}$$

where Λ^y is the likelihood function for a group y, c_y is the summarized normalization constant, r is the channel number of a group y, and G is the number of group. Eq. (28) above provides the probability matrices used for combination of MC-conditioned estimates and covariance in the next step. It can also show us how to use these parameter results for collaborative grouping criteria with multiple sensors.

www.manaraa.com

3.4.4 Feedback from Switching Probability Update to Stage 1 for Grouping Criteria with Distributed Sensors

For the collaborative grouping, we introduced the adaptive hyper-parameter ($\beta_y(k)$) in Chapter 3.3.1. The adaptive hyper-parameter $\beta_y(k)$ can be dynamically increased or decreased depending on the weight of the channel. The weight of channel can be represented as the switching probability. That means we can use the switching probability ($\mu^y(k)$) as a reference to adjust the adaptive hyper-parameter as follows:

$$\begin{aligned} &\beta_y(k) > \beta_y(k-1) \quad if \quad \mu^y(k) > \mu^y(k-1) \\ &\beta_y(k) = \beta_y(k-1) \quad if \quad \mu^y(k) = \mu^y(k-1). \\ &\beta_y(k) < \beta_y(k-1) \quad if \quad \mu^y(k) < \mu^y(k-1) \end{aligned} \tag{29}$$

If there is no change of the switching probability, $\beta_y(k)$ is the same as $\beta_y(k–1)$. If the current switching probability is greater than the previous one, $\beta_y(k)$ could be increased; otherwise, $\beta_y(k)$ could be decreased as shown in (29). Therefore, we can calculate the difference ($\Delta\mu^y$) between the current switching probability and the previous one as follows:

$$\Delta\mu^y = \mu^y(k) - \mu^y(k-1) \tag{30}$$

That means the adaptive hyper-parameter $\beta_y(k)$ can be increased or decreased based on the current switching probability compared to the previous one. In Fig. 22, we have added the red line indicating how the difference of the switching probability in Stage 2 would be used for Stage 1.

3.4.5 Combination of MC conditioned estimates and covariance

Combination of the MC conditioned estimates and covariance is done according to the mixture equations

www.manaraa.com

$$\hat{x}_a^y(k) = \sum_{b=1}^{r} \bar{x}_b^y(k)[\mu_{ab}^y(k)], \quad P_a^y(k) = \sum_{b=1}^{r} [\mu_{ab}^y(k)][P_b^y(k) + DP_b^y(k)]. \tag{31}$$

where μ_{ab}^y is a switching probability presenting the relationship between channel a and channel b within the same group y, and $DP_b^y(k)$ as an increment to the covariance matrix to account for the difference between the intermediate state and the state estimates from model b, expressed by $[\bar{x}_b^y(k) - \hat{x}_b^y(k)]\cdot[\bar{x}_b^y(k) - (\hat{x}_b^y(\underline{k})]^T$.

Note that the combination of the model-conditioned estimates and covariance matrices in Fig. 18 is extended into Eq. (31) incorporating with the switching probability and $\Delta(k)$. As can be seen in Chapter 3.4.1, we also have adopted the results of Chapter 3.3 on grouping criteria $\Delta(k)$ of Eq. (26). In Fig. 22, the entire flow chart illustrates the idea of MC-IMME proposed in this study. We have added the blue line indicating how the difference ($\Delta(k)$) in Stage 1 would be used for the IMME outcomes of Stage 2, corresponding to (31). This combination is only for output purposes.

3.4.6 Computational Time

We have evaluated how much additional computational time is required when we implement the proposed method by comparing it to KF and the IMME method in Table 5.

Table 5. Comparison of the Computational Complexity (KF vs IMME vs MC-IMME)

Methods	KF	IMME	MC-IMME
Complexity	$O(L\times k\times N^3)$	$O(L\times k\times T(N)$	$O(L\times k\times T(N)$

The computational complexity of KF for the upper bound is orders of growth N^3, where N represents the states estimated using the KF derived by Karlsson *et al*. [202]. The

www.manaraa.com

computational complexity can be increased as a linear function of the sensor number (L) and time k. Accordingly, the asymptotic upper bound of KF is orders of growth $L \times k \times N^3$. IMME extends the complexity by defining $T(N)$ as the asymptotic upper bound of recursive computation based on the states estimated using IMME. In the IMME the computational complexity is increased as a linear function of the independent sensor number (L). In addition, IMME needs recursive computation based on time k. Therefore, the asymptotic upper bound for the worst-case running time of IMME is orders of growth $L \times k \times T(N)$ [210].

Let us define $T(L)$ as a upper bound of iteration execution time for k-means clustering based on L points. Har-Peled *et al.* showed that the k-means heuristic needs orders of growth L iterations for L points in the worst case [193]. In addition, the adaptive grouping method needs to calculate the difference ($\Delta_{ADT}(G)$) of the consecutive log-likelihood functions based on time sequential data (k) for the appropriate group number selection. Therefore, the upper bound for the worst-case running time of the adaptive grouping method is orders of growth $L \times k \times T(L)$.

MC-IMME uses the same recursive computation as IMME with respect to the estimated states. That means the running time of stage 2 is the same as simple IMME. MC-IMME also needs additional computation for the first stage to make grouping. Suppose that asymptotic upper bound of recursive computation ($T(N)$) is equal to a upper bound of iteration execution time for k-means clustering ($T(L)$). Then, the asymptotic upper bound for the computational complexity of Multiple-channel is orders of growth $L \times k \times T(N)$, because both stage 1 and stage 2 have the same orders of growth $L \times k$ [52]. Please note

www.manaraa.com

that IMME and MC-IMME have the identical computational complexity since distributed sensory systems both have the same channel number L, and the same data length k.

www.manaraa.com

3.5 Experimental Results

The motivation of this Chapter is to validate the proposed MC-IMME with comprehensive experimental results. In Chapter 3.5.1, we will describe the target motions for the experimental tests (chest, head, and upper body). For each target motion, the optimal cluster number based on the proposed grouping method is selected in Chapter 3.5.2, and this selection number is further investigated in comparison to grouping number methods using other clustering techniques in Chapter 3.5.3. The prediction accuracy of the proposed MC-IMME is evaluated with the normalized root mean squared error (NRMSE) and the prediction overshoots, in Chapters 3.5.4 and 3.5.5, respectively. We also show CPU time used for the computational time in Chapter 3.5.6.

3.5.1 Motion Data

We have used three kinds of motion data, *i.e.*, chest motion, head motion, and upper body motion. Motion data was collected using a Polhemus Liberty AC magnetic tracker in Fig. 23, operating at 240*Hz* for approximately 20 seconds (4,800 sample dataset) [213]. Eight sensors were attached on the target motion surface with the magnetic source rigidly mounted approximately 25.4 cm from the sensors.

www.manaraa.com

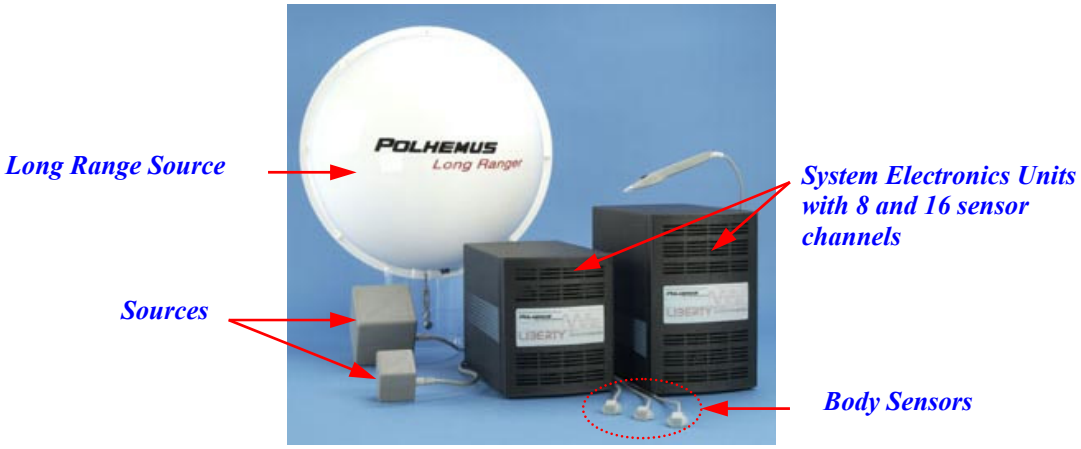

Figure 23. Polhemus Liberty AC magnetic tracker.

Each motion data was randomly selected based on the motion speed for Monte Carlo analysis with three sets of motion data—the first datasets for slow motion, the second datasets for moderate, and the rest for the violent motion. For the target estimation, the experimental tests have been conducted based on repeated random sampling to compute their results for Monte Carlo analysis. Each of the datasets was taken with great care to limit target movement to the type based on Table 6.

Table 6. Characteristics of the Motion Data

Motion Data	Motion Type	Speed (cm/sec)	Recording Time (sec)
Chest_1	Slow motion	0.64 – 0.87	20.72
Chest_2	Moderate motion	6.7-0 – 7.91	20.40
Chest_3	Violent motion	24.84 – 32.63	22.18
Head_1	Slow motion	0.63 – 1.08	20.36
Head_2	Moderate motion	6.62 – 8.37	20.40
Head_3	Violent motion	16.07 – 67.70	21.43
Upper Body_1	Slow motion	0.68 – 1.48	20.83
Upper Body_2	Moderate motion	3.64 – 28.08	20.64
Upper Body_3	Violent motion	38.48 – 118.18	21.03

www.manaraa.com

3.5.2 COLLABORATIVE GROUPING INITIALIZATION

Before the efficient target tracking, the proposed collaborative method needs to make the grouping for distributed sensory data. The objective of this Chapter is to find out the optimal group number with an adaptive hyper-parameter. First, we need to find out the initial hyper-parameter (β_y) in Subchapter 1) and then calculate the group number (G) based on the adaptive (ADT) *posterior* probability $p_{ADT}(y \mid z_j)$. Subchapter 2) and 3) compared the difference (Δ) of the consecutive log-likelihood functions between non-collaborative grouping method described in Chapter 3.2.2 and collaborative grouping method described in Chapter 3.3.1.

1) Calculation of Hyper-parameter (β_y)

The objective of this Chapter is to calculate the initial hyper-parameter (β_y) with potential group numbers. To find out the hyper-parameter, we iterate expectation and maximization steps with sample training data (approximately 2,400 sample dataset) for each motion. We increased the group number (G) of EM process from two to seven to show all the potential hyper-parameters. Please note that we have eight sensory channels, so that we can show all available group numbers in Fig. 24.

www.manaraa.com

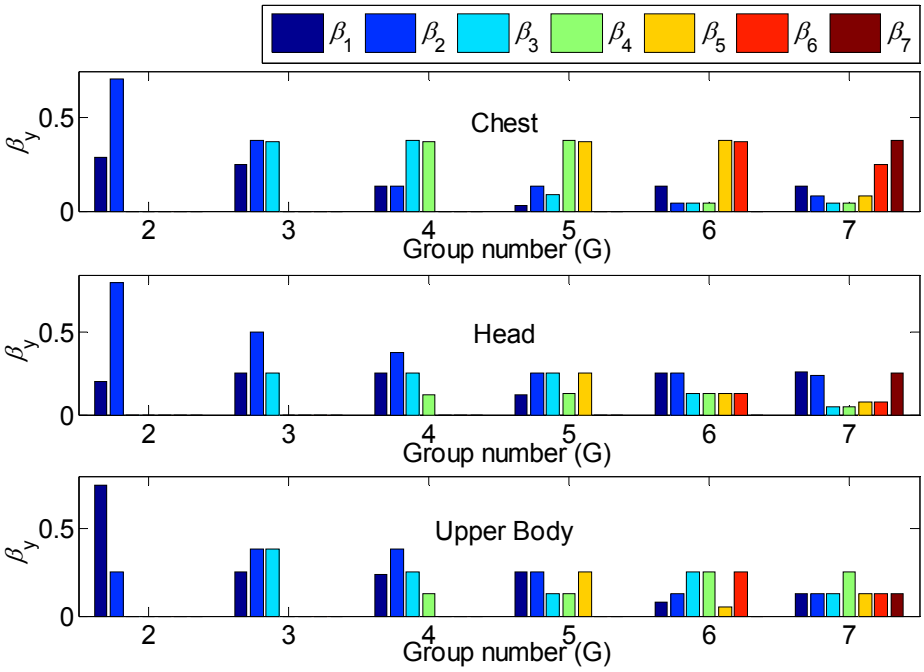

Figure 24. Hyper-parameter values based on the target motion data and group number.

Fig. 24 shows the hyper-parameters (β_y, where y is a group number) described in Eq. (21), based on the target motion data and group number (G). Given in Fig. 24, we can notice that the higher the group number, the bigger the iteration number; and the more even the group distribution probabilities of a sample training data, the smaller the iteration number.

www.manaraa.com

2) *Calculation of the difference (Δ) of the consecutive log-likelihood with non-collaborative grouping*

The objective of this Chapter is to find an optimal cluster number (G^*) with the consecutive log-likelihood functions (20) based on EM process. Fig. 25 below shows the difference ($\Delta(G)$) of the consecutive log-likelihood functions, described in Eq. (20). For example, when G=2, we calculate all the log-likelihood functions of EM operations, and then select the minimum as a representing value in Fig. 25. We iterate the same procedure with different group number (G =2,..., 7) in the three kinds of the motion data. We expect to find out, as described in Chapter 3.2.2, the minimum of $\Delta(G)$.

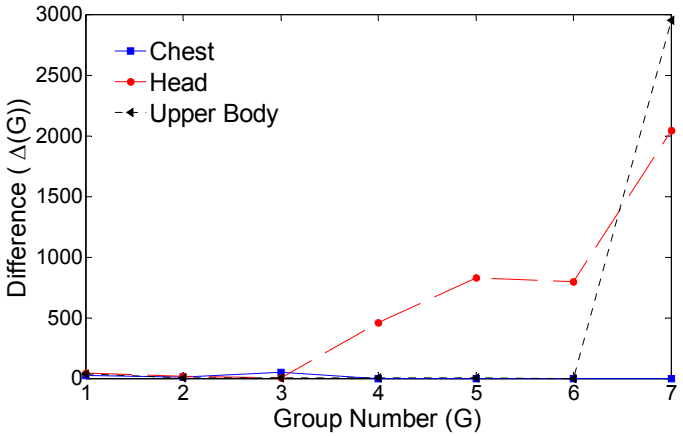

Figure 25. The difference ($\Delta(G)$) with non-collaborative grouping.

Given the results in Fig. 25, we may select the group number G^* for the three datasets: 2, 4, or 6 for Chest; 2 or 3 for Head; and 2, 4, 5, or 6 for Upper Body. As the group numbers are increased, the differences start to become drastically greater.

However, we cannot identify the least minimum number; for example, it is hard to choose among 2, 4, 5, or 6 for Upper Body. Therefore, in the next experiment, we will recalculate the difference ($\Delta_{ADT}(G)$) of the consecutive log-likelihood with collaborative grouping.

www.manaraa.com

3) Calculation of the difference (Δ) of the consecutive log-likelihood with collaborative grouping

The objective of this Chapter is to find an optimal cluster number (G*) using log-likelihood function with the adaptive *posterior* probability (25). Based on the initial hyper-parameter (β_y) (21), we can calculate the adaptive (ADT) *posterior* probability $p_{ADT}(y \mid z_j)$ and iterate E-step (23) and M-step (24) with a specific group number (G).

Now we can show the difference ($\Delta_{ADT}(G)$) of the consecutive log-likelihood functions, described in Eq. (25) of Chapter III.A, with the adaptive *posterior* probability in the three kinds of the motion data. We applied Eq. (25) for the minimum value of $\Delta_{ADT}(G)$. We iterate the same procedure with a different group number (G =2,..., 7), as shown in Fig. 26.

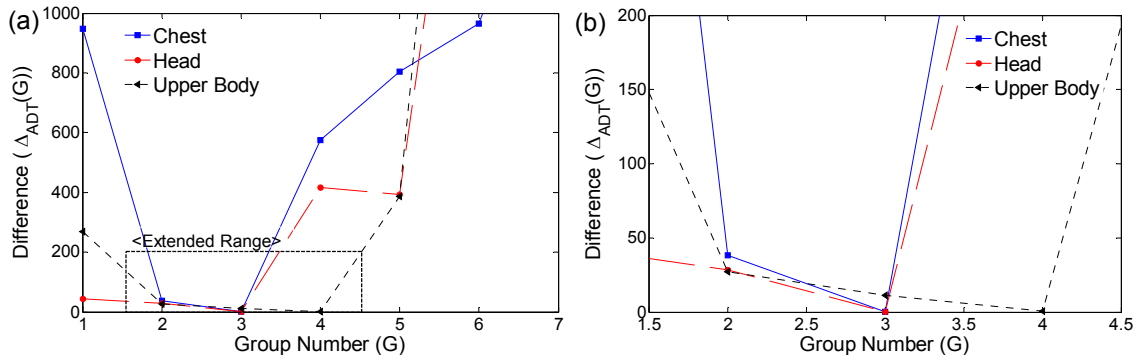

Figure 26. The difference ($\Delta_{ADT}(G)$) with collaborative grouping
(a) whole range, (b) extended range.

In Fig. 26 we can select the group number G* for the three datasets: 3 for Chest; 3 for Head; and 4 for Upper Body. Compared to Fig. 25 and Fig. 26, it is clear that the collaborative grouping provides more distinct difference $\Delta_{ADT}(G)$ of grouping numbers; for example, while Fig. 25 had the candidates of the group numbers 2, 4, 5, or 6 for

www.manaraa.com

Upper Body, Fig. 26 now identifies the minimum number 4 for Upper Body by introducing the adaptive *posterior* probability.

4) Sensor Placement Results of Collaborative Grouping

In the previous Chapter, we have performed the collaborative grouping given the sample training data. The goal of the first stage is to partition all the measurements into the grouping for a tracking relationship.

Now we can show the sensor placement results of each motion based on the given data in Fig. 27. In this figure, we denote symbols (+) as the sensor placement for each motion, the ellipse as each group, and the number of ellipse in a figure as the group number. As can be seen in the following figure, the group numbers for each motion data are the same as 3 for Chest, 3 for Head, and 4 for Upper Body.

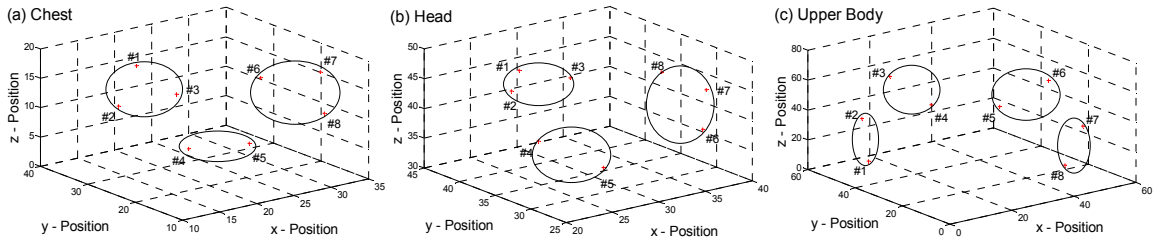

Figure 27. Sensory Position and Grouping of Motion Data.
(a) Chest, (b) Head, and (c) Upper body

3.5.3 Comparison of Grouping Methods with Other Techniques

To find out the best grouping numbers, we have evaluated several clustering algorithms: *k*-means [191], spectral clustering [195] [196], nonparametric Bayesian inference [214], and EM algorithm [189]. To determine the quality of group number hypothesis, we would like to show established metrics, *i.e.*, Akaike's Information Criterion (AIC) that provides a measure of model quality by simulating a statistical model for model selection

www.manaraa.com

[215]. For this selection, we assume that the model errors are normally and independently distributed and that the variance of the model errors is unknown but equal for them all.

Let n be the number of training observations. The formula AIC can be expressed as a simple function of the residual sum of squares (RSS), *i.e.*, AIC = $2k + n[\ln(\mathrm{RSS}/n)]$, where k and RSS are the number of parameters in the statistical model and the residual sum of squares ($\sum_{i=1}^{n} \varepsilon_i^2$, ε_i : estimated residuals for a candidate model), respectively (see Chapter 2.2 in [215]). Given any estimated models, the model with the minimum value of AIC is the one to be preferred.

Table 7. Comparison of grouping number methods with AIC values

		k-means	Spectral clustering	Nonparametric Bayesian	EM algorithm
Chest	G=2	7444	7411	7346	7404
	G=3	**7393**	**6328**	**6942**	**7379**
	G=4	7608	6356	7523	7603
	G=5	7824	6977	7383	7550
	G=6	7674	7365	7662	7680
	G=7	7761	7177	7514	7497
Head	G=2	6272	**6272**	6284	6256
	G=3	**6222**	6314	**5847**	**6220**
	G=4	6783	6509	6500	6770
	G=5	6677	6455	6337	6305
	G=6	6427	6512	6325	6529
	G=7	6711	6471	6402	6530
Upper Body	G=2	10874	10885	10760	10827
	G=3	11043	10967	10645	10780
	G=4	**10809**	**10874**	**10617**	**10448**
	G=5	10962	10928	10757	10928
	G=6	10941	10987	10938	10987
	G=7	11127	10901	10876	10861

We set the number of training observations to n = 1000 for all the datasets. Table 7 shows the comparison of grouping number methods with AIC values. We can notice that all of the

www.manaraa.com

methods except the spectral clustering method have selected the identical grouping numbers: G=3 for Chest datasets, G=3 for Head datasets, and G=4 for Upper Body. Please note that all the grouping number methods have the minimum AIC values for Chest (G=3) and Upper Body (G=4) datasets. In Head datasets, there exists inconsistency among the methods. That means head motion can be classified into different groups in the given datasets. For our tracking estimation, we use grouping number G=3 for Head datasets because of the minimum AIC value in the given results.

3.5.4 MULTI-CHANNEL (MC) IMME

Based on the group number (G^*) chosen in the experiment Chapter 3.5.2 of the first stage, we can perform the target estimation using Multi-channel (MC) IMME of the second stage with respect to each group.

1) Position Estimation

We compare the performance of motion tracking estimation among KF, IMME, and MC-IMME. Fig. 28 shows that MC-IMME can estimate the target motion more accurately than other tracking methods, KF and IMME at the initial stage.

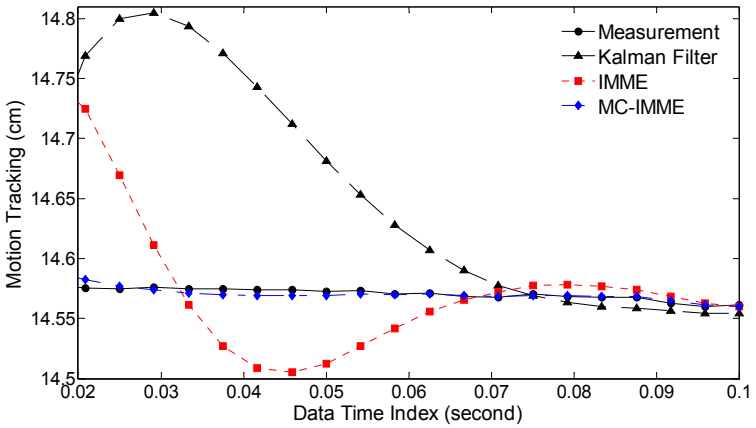

Figure 28. Comparison of motion tracking estimation for Head_1 dataset.

www.manaraa.com

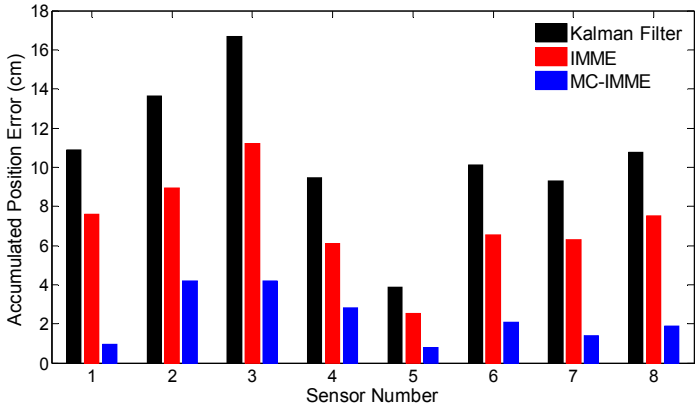

Figure 29. Comparison of accumulated position error of each channel for Head_1.

In addition, we compare the accumulated position errors for each channel across the entire measurement period among KF, IMME, and MC-IMME. Fig. 29 shows that the accumulated position errors of KF and IMME are greater that those of MC-IMME for Head_1 dataset for each sensor channel. We can notice that MC-IMME outperforms IMME by 38.33% in the benign Head motion.

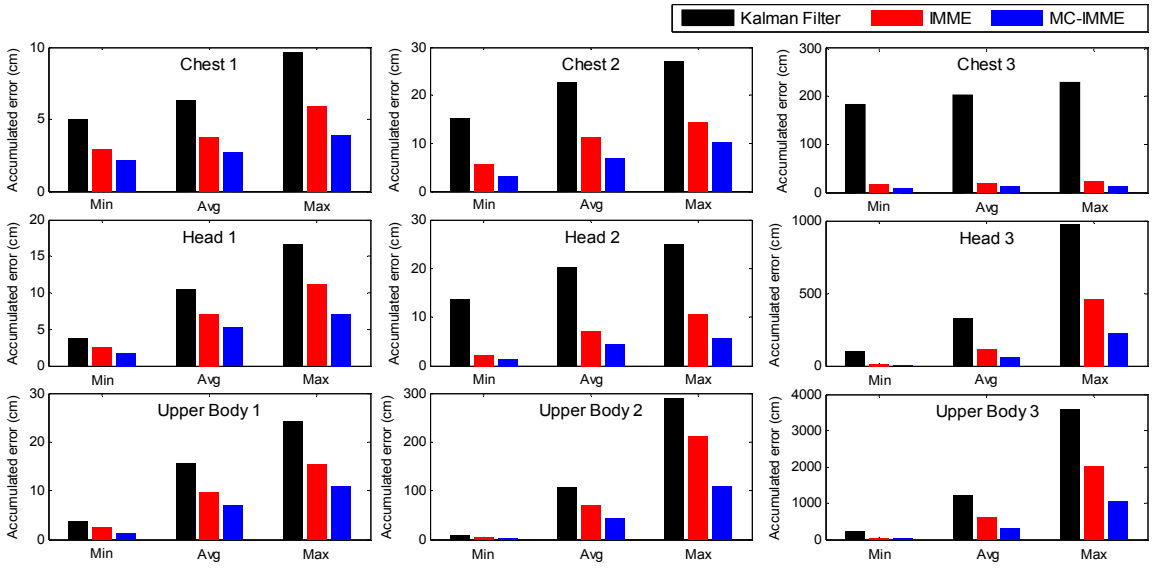

Figure 30. Overall performance of accumulated error among the datasets

Fig. 30 shows the overall performance of accumulated error among the datasets listed in Table 6. As shown in Fig. 30, MC-IMME can show 68.84% of the average improvement

www.manaraa.com

with comparison to KF. In addition, the proposed method outperforms IMME around by 25.38~27.66% in the benign motion, 38.33~39.14% in the moderate motion, and 42.94~48.75% in the aggressive motion. Please note that the proposed method can achieve 48.75% improvement over IMME in Upper_Body_3 dataset.

2) Prediction Time Horizon

For the prediction accuracy, we changed the prediction time horizon. Here, prediction time horizon is the term to represent the time interval window to predict the future sensory signal. We would like to compare the error performance among the various prediction time horizons between IMME and MC-IMME in Fig. 31. For the comparison, we used a normalization that is the normalized root mean squared error (NRMSE) between the predicted and actual signal over all the samples in the test datasets, as follows: $NRMSE = \sqrt{\sum_i (z_i - \hat{z}_i)^2 / \sum_i (z_i - m_z)^2}$.

where z_i is the i^{th} measurement, $\hat{z}_i$ is the estimation of the i^{th} measurement, and m_z is the mean of all the measurements. This metric is dimensionless and allows us to compare prediction accuracy for different signals of widely varying amplitude.

www.manaraa.com

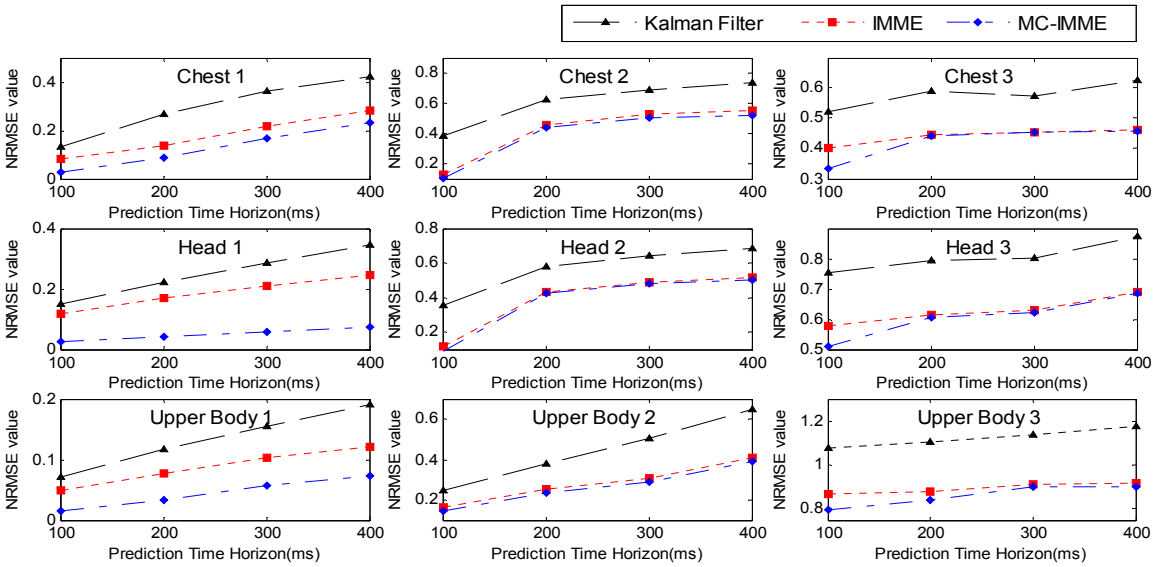

Figure 31. Error performance among prediction time horizon.

In Fig. 31, the error performance of Chest_1 dataset in the proposed MC-IMME was improved by 61.62% for KF and 36.02% for IMME of the average prediction time horizon. We can notice that the proposed method outperforms KF and IMME in the other Chest motion datasets as well, even though the average improvements were less than 7% with comparison to IMME. The average improvements were 42.77% for KF and 16.35% for IMME.

In the Head_1 dataset, the error performance was significantly improved by 80.24% for KF and 73.40% for IMME of the average prediction time horizon. Notice that the improvement of error performance for the proposed method maintained around 65% across the prediction time horizons. In the other Head motion datasets, the proposed method can improve other methods, even though the average improvements were less than 5% in comparison to IMME. The average improvements were 47.71% for KF and 27.92% for IMME.

In the Upper_Body_1 dataset, the proposed method was improved by 68.79% for KF and 52.52% for IMME of the average prediction time horizon. We can notice that the

www.manaraa.com

improvement of MC-IMME maintained around 40% across the prediction time horizons. We can also notice that the proposed method outperforms KF for 44.10% and IMME for 20.91% of the average prediction time horizon over all the datasets, even though the average improvements were less than 6.3% for Upper_Body_2 and 3.91 for Upper_Body_3 in comparison to IMME.

3) Velocity estimation

Fig. 32 shows the average velocity of group number 1 for Head_1 dataset. The velocity estimations of MC-IMME align more closely to the measurements, than KF and IMME values. The overall improvements for the group number 1 of Head_1 dataset are 50.76% for KF and 49.40% for IMME.

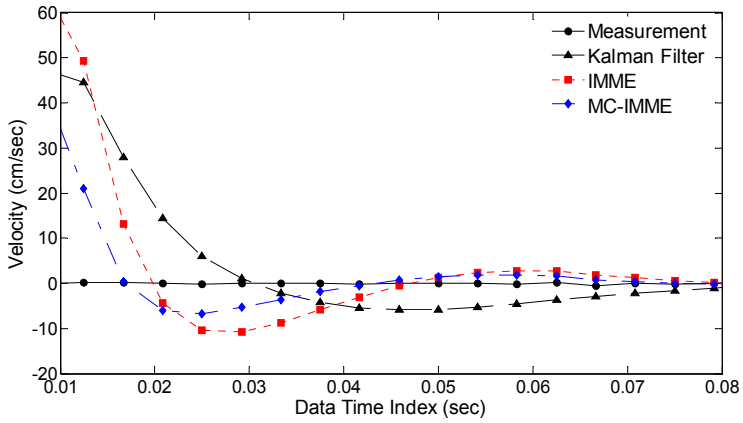

Figure 32. Comparison of average velocity estimation of group number 1 for Head_1.

4) Effect of the feedback/forward method

We would like to show the advantage of the proposed feedback/forward method by comparing the performance of velocity estimation of MC-IMME with no feedback/forward vs. feedback/forward. We have evaluated the tracking performance of the average velocity for the Chest_3 dataset in Fig. 33. We have observed in Fig. 33 that

www.manaraa.com

the feedback/forward method slightly increases the performance of pure MC-IMME in 14%.

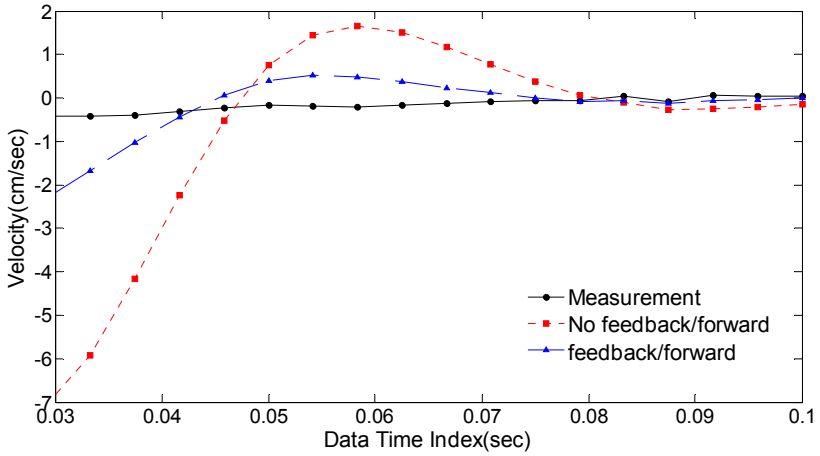

Figure 33. Comparison of the velocity estimations with no feedback/forward vs. feedback/forward.

Table 8. Comparison of Overall Velocity error averaged among 8 channels

Datasets	No feedback/forward (cm/sec)	Feedback/forward (cm/sec)
Chest_1	0.415	0.292
Chest_2	0.335	0.211
Chest_3	0.605	0.514
Head_1	1.527	1.168
Head_2	1.386	1.014
Head_3	1.517	1.201
Upper Body_1	2.012	1.550
Upper Body_2	3.162	2.572
Upper Body_3	3.999	3.404

We show all nine datasets to compare the overall performance of velocity error averaged among eight channels between no feedback/forward vs. feedback/forward. Table 8 shows the overall performance of velocity error among the datasets listed in Table 6. Given in Table 8, feedback/forward method outperforms no feedback/forward method around 15~37% for Chest dataset, 20~26% for Head dataset, and 14~22% for Upper Body dataset.

www.manaraa.com

3.5.5 PREDICTION OVERSHOOT

We define overshoot for cases in which the predicted output exceeds a certain marginal value with confidence levels corresponding to the tolerances [184]. The initialization process is an essential step of Kalman filter-based target tracking. Unfortunately, this process produces an unexpected prediction estimate error. To compensate for the prediction estimate error, we used a marginal value to generate a 95% prediction interval for the measurement prediction, so that we can define the upper bound and the lower bound by adding the marginal value to the measurement and subtracting the marginal value from the measurement, respectively [184].

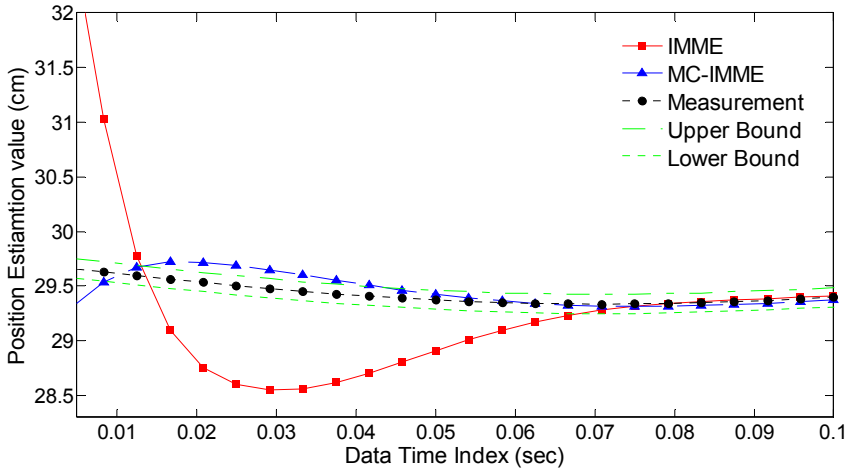

Figure 34. Prediction overshoot comparison between IMME and MC-IMME.
The prediction overshoot error can be improved with MC-IMME.

Fig. 34 shows the prediction overshoot comparison between IMME and MC-IMME. We can notice that the prediction overshoot error with distributed sensory data was improved in the average of 10.84% with slow motion, 12.43% with moderate motion, and 34.66% with violent motion. Moreover, the total error of MC-IMME was decreased by 23.63% in comparison with that of IMME.

www.manaraa.com

Table 9. Prediction Overshoot Comparison listed in Table 6

Datasets	Average number of overshoot dataset (IMME/MC-IMME) (Unit: overshoot dataset #)	Improvement (%)
Chest_1	15.00/13.37	10.83
Chest_2	15.25/13.12	13.93
Chest_3	26.00/12.75	**50.96**
Head_1	15.00/13.75	8.33
Head_2	14.75/13.62	7.62
Head_3	62.12/42.50	**31.58**
Upper Body_1	15.00/13.37	13.37
Upper Body_2	27.75/23.37	15.76
Upper Body_3	99.62/78.25	**21.45**

Table 9 shows the comparison of overshoot dataset sample numbers between IMME and MC-IMME, where the second column represents the average number of overshoot dataset samples listed in Table 6. The overall improvement in the benign motion is around 10%, whereas the overall improvement in the aggressive motion is over 20%. That means distributed sensory data can reduce the prediction estimate error at the beginning of target tracking. We may expect this prediction accuracy to decrease for different datasets (*e.g.*, including head and chest motions) due to the lack of interactive relationships. For our experimental tests, however, we have focused on the human body motion including head and chest. That means our experimental results can be generalized to the upper body.

3.5.6 Computational Time

Regarding CPU experimental time, we have evaluated the overall performance of average CPU time used for the datasets listed in Table 6. We have collected the motion data using a Polhemus Liberty AC magnetic tracker with eight sensors, and then conducted the experimental test for the computational complexity with offline. We have implemented

www.manaraa.com

the proposed method with Matlab language using a PC of Pentium core 2.4 GHz with RAM 3.25 GB.

Table 10. CPU Time Used among the Datasets

Datasets	KF	IMME	MC-IMME
Chest	0.244	0.957	0.802
Head	0.246	0.966	0.804
Upper Body	0.249	0.974	0.829

(Unit: ms/sample numbers)

In Table 10, we evaluated the individual dataset to compare KF and IMME with MC-IMME. Table 10 shows the overall performance of CPU time used among the datasets. Here, we used the period of the first 20 seconds for all nine datasets to calculate CPU time used for KF, IMME, and MC-IMME. For the comparison of the different target-tracking methods, we evaluated the computational time calculating target-tracking estimate filters. That means we only counted the calculation time for KF and IMME operations with all the methods. Note that MC-IMME can improve approximately 16% of the average computational time with comparison to IMME, even though it requires more than twice the computational time of KF, as shown in Table 10. An interesting result is that the proposed method can improve the computational time over IMME. We think that the actual difference for CPU time used in Table 10 mainly comes from the simultaneous calculation of distributed sensory data in MC-IMME. In IMME, it needs to calculate target-tracking estimation individually, whereas MC-IMME can evaluate a couple sets of target estimation simultaneously.

www.manaraa.com

3.6 SUMMARY

In this Chapter we have presented a new MC-IMME and grouping criteria with distributed sensors placement. Our new method has two main contributions to improve the traditional IMME-based target tracking. The first contribution is to comprehensively organize the distributed channel sensory process by providing a collaborative grouping number with the given datasets to achieve the efficient target estimation. The second contribution is to add feedback/forward modules to import the results from the first multiple channels grouping for interactive tracking estimation to employ a tracking relationship with each other.

The experiment results validated that we can identify a proper group number with the collaborative grouping method using hyper-parameter and the collaborative grouping method can outperform the conventional target-tracking methods, *e.g.*, KF and IMME, by comparing the prediction overshoot and the performance of tracking errors with respect to the accumulated position error. We have also evaluated that MC-IMME with feedback/forward method can increase the performance of pure MC-IMME throughout the experiment results. The prediction overshoot error at the beginning of target tracking can be improved in the average of 19.31% with employing a tracking relationship in this specific datasets. For the generalized extent of motion tracking, more complicated motions and different sensory positions are required. This will be our future works.

www.manaraa.com

The extended Kalman filter (EKF) can be used for the purpose of training nonlinear neural networks to perform desired input-output mappings. To improve the computational requirements of the EKF, Puskorius *et al*. proposed the decoupled extended Kalman filter (DEKF) as a practical remedy for the proper management of computational resources. This approach, however, sacrifices computational accuracy of estimates because it ignores the interactions between the estimates of mutually exclusive weights. To overcome such a limitation, therefore, we proposed hybrid implementation based on EKF (HEKF) for respiratory motion estimate, which uses the channel number for the mutually exclusive groups and the coupling technique to compensate the computational accuracy. Moreover, the authors restricted to a DEKF algorithm for which the weights connecting inputs to a node are grouped together. If there are multiple input training sequences with respect to time stamp, the complexity can increase by the power of input channel number. To improve the computational complexity, we split the complicated neural network into a couple of the simple neural networks to adjust separate input channels. The experiment results validated that the prediction overshoot of the proposed HEKF was improved by 62.95% in the average prediction overshoot values. The proposed HEKF showed the better performance by 52.40% improvement in the average of the prediction time horizon. We have evaluated that a proposed HEKF can outperform DEKF by comparing the prediction overshoot values, the performance of tracking estimation value and the normalized root mean squared error (NRMSE).

www.manaraa.com

4.1 INTRODUCTION

The problem of predicting the moving objects with a given reference trajectory is a common estimate problem [216] [217] [218] [219] [220]. Kalman filters can be widely used in many industrial electronics for the state estimation and prediction [221] [222] [223] [224] [225] [226] [227] [228] [229]. Due to increasingly complex dynamical systems, a variety of methodologies has been proposed based on the Kalman filter and its hybrid approach [145] [229] [230] [231] [232] [233]. The recurrent neural network (RNN) can also be one of the estimation methods for the predictive control in many application systems [234] [235] [236] [237] [238] [239] [240] [241] [242] [243] [244] [245] [246]. Here, RNN is a class of neural network where connections between units exhibit dynamic temporal behavior with their synaptic weights. Owing to this dynamic behavior, RNN can implement dynamical nonlinear multivariable discrete-time systems of arbitrary complexity [247] [248] [249] [250].

A target-tracking estimation can be one of the applications for RNN because of its adaptive learning, an ability to learn how to do tasks based on the data given for training or initial experience [234] [235] [239] [240]. For example, RNN can be used for the respiratory motion prediction for real-time motion adaptation in the medical application [36] [37] [40] [47] [46] [87]. Because of the self-organized characteristic of neural networks, it can have a built-in capability to adapt their synaptic weights to change based on the given samples in the specific circumstance; thus, it can provide the better performance in comparison to the conventional methods of the respiratory motion prediction [4] [251] [252] [253] [254]. Intrinsically, training algorithm for RNN became

www.manaraa.com

an issue to improve the performance of dynamical systems with respect to the specific environment [255].

There are several algorithms available for training the weights of recurrent networks based on streams of input-output data. Basically, the most widely used are the back-propagation-through-time (BPTT) algorithm [256] [257] [258] and the real-time recurrent learning (RTRL) algorithm [258] [259] [260] [261], which are both based on computation of the gradient of an output error measure with respect to network weights. However, the calculation of dynamic derivatives of a recurrent network's outputs with respect to its weights by RTRL is computationally expensive, since these derivatives cannot be computed by the same back-propagation mechanism that was employed in the training of multilayer perceptron (MLP) networks [146].

As an alternative or improvement of the gradient descent-based methodology, several authors have noted that the extended Kalman filter (EKF) can also be used for the purpose of training networks to perform desired input-output mappings [145] [230] [231] 232] [233]. Note that the predictor-corrector property is an intrinsic property of the Kalman filter, its variants, and extensions. Thus, whereas in traditional applications of the Kalman filter for sequential state estimation, the roles of predictor and corrector are embodied in the Kalman filter itself; in supervised-training applications these two roles are split between the RNN and the EKF. Here, the RRN in which the input training samples are applied to the recurrent multilayer perceptron (RMLP) as the excitation, performs the role of the predictor, and the EKF, in which the training samples of desired response are applied to the EKF as the observable to provide the supervision, performs the role of the corrector [146].

www.manaraa.com

With comparison to the gradient descent algorithms, EKF-based algorithms for recurrent networks do not require batch processing, making them more suitable for on-line use. To improve the computational requirements of the EKF, Puskorius *et al.* proposed decoupled extended Kalman filter (DEKF) as a practical remedy for the proper management of computational resources [145]. The author in [145] restricted to a DEKF algorithm for which the weights connecting inputs to a node are grouped together. This approach, however, sacrifices computational complexity and estimation accuracy since DEKF defines a node as the mutually exclusive weight group. If there are multiple input training sequences with respect to time stamp, the complexity can increase by the power of input channel number. To overcome these limitations, we do not adopt the mutually exclusive weight groups. Instead, we adopt the channel number for the mutually exclusive groups to propose the coupling technique to compensate the computational accuracy using multiple sensory channel inputs. We call this new proposed method Hybrid motion estimation based on EKF (HEKF).

The contribution of this study is twofold: First, we propose a new approach to split the whole RMLP with the complicated neuron number into a couple of RMLPs with the simple neuron number to adjust separate input channels. Second, we present a new method for the respiratory motion estimation using EKF which adapts the coupling technique using multiple channel inputs for the mutually exclusive groups to compensate the computational accuracy, instead of mutually exclusive weight groups.

This Chapter is organized as follows. In Chapter 4.2, the theoretical background for the proposed algorithm is briefly discussed. In Chapter 4.3, the proposed hybrid implementation based on EKF for RNN with multiple sensory channel inputs are

www.manaraa.com

presented in detail. Chapter 4.4 presents and discusses experimental results of proposed filter design method— efficient estimation of the measurements, optimized group number for RMLP, prediction overshoot analysis, prediction time horizon, and computational complexity of HEKF and DEKF. A summary of the performance of the proposed method is presented in Chapter 4.5.

www.manaraa.com

4.2 Related Work

4.2.1 Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. A network with a rich representation of past outputs is a fully connected recurrent neural network, known as the Williams-Zipser network, as shown in Fig. 35 [255]. This network consists of three layers: the input layer, the processing layer and the output layer. For each neuron i (i = 1, 2,…, N), the elements u_j of the input vector (j = 1, 2,…, $M + N + 1$) to a neuron u are as follows:

$$u_j^T(k) = [x(k-1),...,x(k-M),1,y_1(k-1),...,y_N(k-1)], \tag{32}$$

where M is the number of external inputs, N is the number of feedback connections, $(\cdot)^T$ denotes the vector transpose operation, and the $(M + N + 1) \times 1$ dimensional vector u comprises both the external and feedback inputs to a neuron, as well as the unity valued constant bias input. Eq. (32) is weighted, and then summed to produce an internal activation function of a neuron v as follows:

$$v_i(k) = \sum_{l=1}^{M+N+1} w_{i,l}(k)u_l(k), \tag{33}$$

where w are weights. Finally Eq. (33) is fed through a nonlinear activation function Φ, to form the output of the ith neuron y_i. Here, the function Φ is a monotonically increasing sigmoid function with slope β, as for instance the logistic function,

$$\Phi(v) = \frac{1}{1+e^{-\beta v}}. \tag{34}$$

www.manaraa.com

At the time instant k, for the ith neuron, its weights form a $(M + N + 1) \times 1$ dimensional weight vector $w_i^T(\mathrm{k}) = [w_{i,1}(k),\ldots, w_{i,\,M+N+1}(k)]$. One additional element of the weight vector w is the bias input weight. After feeding (33) into (34) using the function Φ, the output of the ith neuron y_i can be formed as follows:

$$y_i(k) = \Phi(v_i(k)), \quad i = 1,2,...,N\,. \tag{35}$$

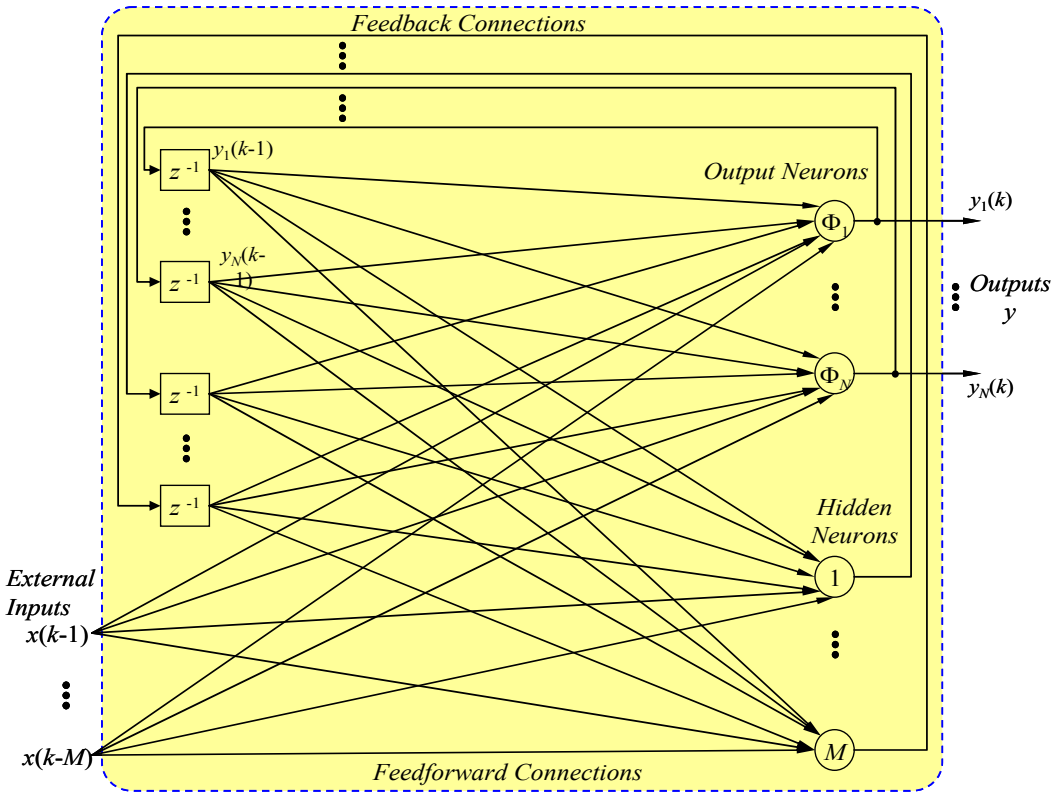

Figure 35. A fully connected recurrent neural network with external inputs.

In a recurrent neural network architecture, the feedback brings the delayed outputs from hidden and output neurons back into the network input vector $u(k)$, as shown in Fig. 35. Due to the recursive function at each time instant, the network is presented with the raw, possibly noisy, external input data $x(k)$, $x(k\text{-}1)$,…, $x(k\text{-}M)$ from Fig. 35 and Eq. (32), and filtered data $y_1(k\text{-}1)$,…, $y_N(k\text{-}1)$ from the network output. Intuitively, this filtered input history helps to improve the processing performance of recurrent neural networks, as

www.manaraa.com

compared with feedforward networks. Therefore, a recurrent neural network should be able to process signals corrupted by additive noise even in the case when the noise distribution is varying over time.

4.2.2 Extended Kalman Filter for Recurrent Neural Networks

As mentioned in the previous Chapter, the learning algorithm based on gradient descent, exemplified by the real-time recurrent learning algorithm, is typically slow due to reliance on instantaneous estimates of gradients [145]. We can overcome this serious limitation by using the supervised training of a recurrent network which recursively utilizes information contained in the training data in a manner going back to the first iteration of the learning process. That is based on Kalman filter theory [146].

Consider a recurrent network built around a static multilayer perceptron with s weights and p output nodes. Let the vectors $w(k)$, $v(k)$ and $u(k)$ denote the weights of the entire network, the recurrent activities inside the network and the input signal applied to the network at time k, respectively. With adaptive filtering in mind, the system state model and measurement model equations for the network may be modeled as follows:

$$w(k+1) = w(k) + q(k), \tag{36}$$

$$d(k) = b(w(k), v(k), u(k)) + r(k), \tag{37}$$

where $q(k)$ and $r(k)$ are the process and measurement noise with the property of a multivariate zero-mean white noise with covariance matrix, Q and R, respectively. $d(k)$ is the observable and $b(\cdot,\cdot,\cdot)$ is measurement function that accounts for the overall nonlinearity of the multilayer perceptron from the input to the output layer.

For us to be able to apply the EKF algorithms as the facilitator of the supervised-learning task, we have to linearize the measurement equation (37) by retaining first-order terms in

www.manaraa.com

the Taylor-series expansion of the nonlinear part of the equation. With $b(w(k),v(k),u(k))$ as the only source of nonlinearity, we may approximate Eq. (37) as follows:

$$d(k) = B(k)w(k) + r(k), \tag{38}$$

where $B(k)$ is the $p \times s$ measurement matrix of the linearized model. The linearization consists of the partial derivatives of the p outputs of the whole network with respect to the s weights of the model as shown

$$B(k) = \begin{bmatrix} \frac{\partial b_1}{\partial w_1} & \frac{\partial b_1}{\partial w_2} & \cdots & \frac{\partial b_1}{\partial w_s} \\ \frac{\partial b_2}{\partial w_1} & \frac{\partial b_2}{\partial w_2} & \cdots & \frac{\partial b_2}{\partial w_s} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial b_p}{\partial w_1} & \frac{\partial b_p}{\partial w_2} & \cdots & \frac{\partial b_p}{\partial w_s} \end{bmatrix}. \tag{39}$$

The partial derivatives in Eq. (39) are evaluated at $w(k)= \hat{w}(k|k-1)$, where $\hat{w}(k|k-1)$ is the prediction of the weight vector $w(k)$ computed by extended Kalman filter at time k, given the observed data up to time $k-1$.

For the purpose of our present discussion, the relevant equations in the EKF algorithm are the innovations process and the weight update equations as follows:

$$\alpha(k) = d(k) - b(k)(\hat{w}(k \mid k-1), v(k), u(k)), \tag{40}$$

$$\hat{w}(k+1 \mid k) = \hat{w}(k \mid k-1) + G(k)\alpha(k), \tag{41}$$

where $\alpha(k)$ is $p \times 1$ matrix denoting the innovations defined as the difference between the desired response $d(k)$ for the linearized system and its estimation, $\hat{w}(k|k-1)$ is $s \times 1$ vector denoting the estimate of the weight vector $w(k)$ at time k given the observed data up to time $k-1$, $\hat{w}(k|k)$ (= $\hat{w}(k+1|k)$) is the filtered updated estimate of $w(k)$ on receipt of the observable $d(k)$. $G(k)$ is $s \times p$ matrix denoting the Kalman gain that is an integral part of the EKF algorithm.

www.manaraa.com

Let $\Gamma(k)$, $P(k|k\text{-}1)$ and $P(k|k)$ be defined as $p \times p$ matrix denoting the global conversion factor for the entire network, $s \times s$ prediction-error covariance matrix and $s \times s$ filtering-error covariance matrix, respectively. In light of these new notations, we can write the EKF algorithms as follows:

$$\Gamma(k) = \left[B(k)P(k \mid k-1)B^T(k) + R(k)\right]^{-1}, \tag{42}$$

$$G(k) = P(k \mid k-1)B^T(k)\Gamma(k), \tag{43}$$

$$P(k \mid k) = P(k \mid k-1) - G(k)B(k)P(k \mid k-1), \tag{44}$$

$$P(k+1 \mid k) = P(k \mid k) + Q(k), \tag{45}$$

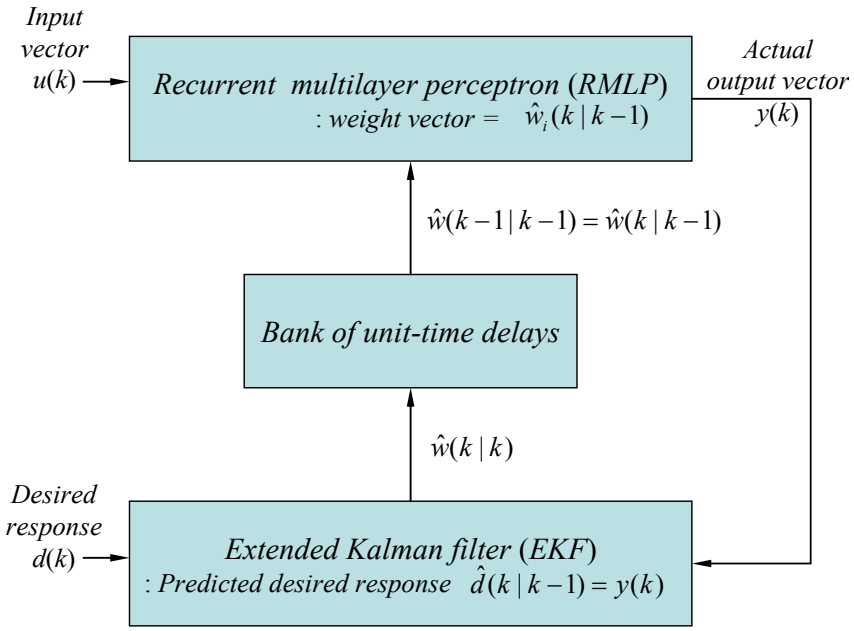

Figure 36. Closed-loop feedback system embodying the RMLP and the EKF

As can be seen in Fig. 36, with the weight vector set at its old predicted value $\hat{w}(k|k-1)$, the RMLP computes the actual output vector $y(k)$ in response to the input vector $u(k)$. After updating the old estimate of the weight vector by operating on the current desired response $d(k)$, the filtered estimate of the weight vector $\hat{w}(k|k)$ is computed in accordance with Eq.

www.manaraa.com

(41). Note that in EKF-RNN of Fig. 36, the recurrent neural network performs the role of the predictor and the extended Kalman filter performs the role of the corrector.

www.manaraa.com

4.3 MULTI-CHANNEL COUPLED EKF-RNN

4.3.1 DECOUPLED EXTENDED KALMAN FILTER (DEKF)

The computational requirement of the EKF is dominated by the need to store and update the filtering-error covariance matrix $P(k|k)$ at time-step k. For a recurrent neural network containing p output nodes and s weights, the computational complexity of the EKF is $O(ps^2)$ and its storage requirement is $O(s^2)$. For large s, these requirements may be highly demanding. In such situations, we need to look for a practical remedy for the proper management of computational resources, *i.e.* Decoupled Extended Kalman Filter (DEKF) [145] [146].

The basic idea behind the DEKF is to ignore the interactions between the estimates of certain weights in the recurrent neural network. If the weights in the network are decoupled in such a way that we can create *mutually exclusive weight groups*, then the covariance matrix $P(k|k)$ is structured into a block-diagonal form as shown in the bottom left of Fig. 37.

Let g denote the designated number of mutually exclusive disjoint weight groups. Also, for $i = 1, 2,\ldots, g$, let $\hat{w}_i(k|k)$, $P_i(k|k)$ and $G_i(k)$ be defined as filtered weight vector, subset of the filtering-error covariance matrix and Kalman gain matrix for the group i, respectively. The concatenation of the filtered weight vectors $\hat{w}_i(k|k)$ forms the overall filtered weight vector $\hat{w}(k|k)$. In light of these new notations, we can now rewrite the DEKF algorithm for the *i-th* weight group as follows:

$$\alpha_i(k) = d_i(k) - b_i(k)(\hat{w}_i(k \mid k-1), v_i(k), u_i(k)), \tag{46}$$

$$\Gamma(k) = [\sum_{i=1}^{g} B_i(k) P_i(k \mid k-1)(B_i(k))^T + R(k)]^{-1}, \tag{47}$$

www.manaraa.com

$$G_i(k) = P_i(k \mid k-1)(B_i(k))^T \Gamma(k), \tag{48}$$

$$\hat{w}_i(k+1 \mid k) = \hat{w}_i(k \mid k-1) + G_i(k)\alpha_i(k), \tag{49}$$

$$P_i(k+1 \mid k) = P_i(k \mid k) + Q_i(k), \tag{50}$$

$$P_i(k \mid k) = P_i(k \mid k-1) - G_i(k)B_i(k)P_i(k \mid k-1), \tag{51}$$

where, $\alpha_i(k)$, $\Gamma(k)$, and $P_i(k+1|k)$ denote the difference between the desired response $d_i(k)$ for the linearized system and its estimation for the *i-th* weight group, the global conversion factor for the entire network, and the prediction-error covariance matrix, respectively.

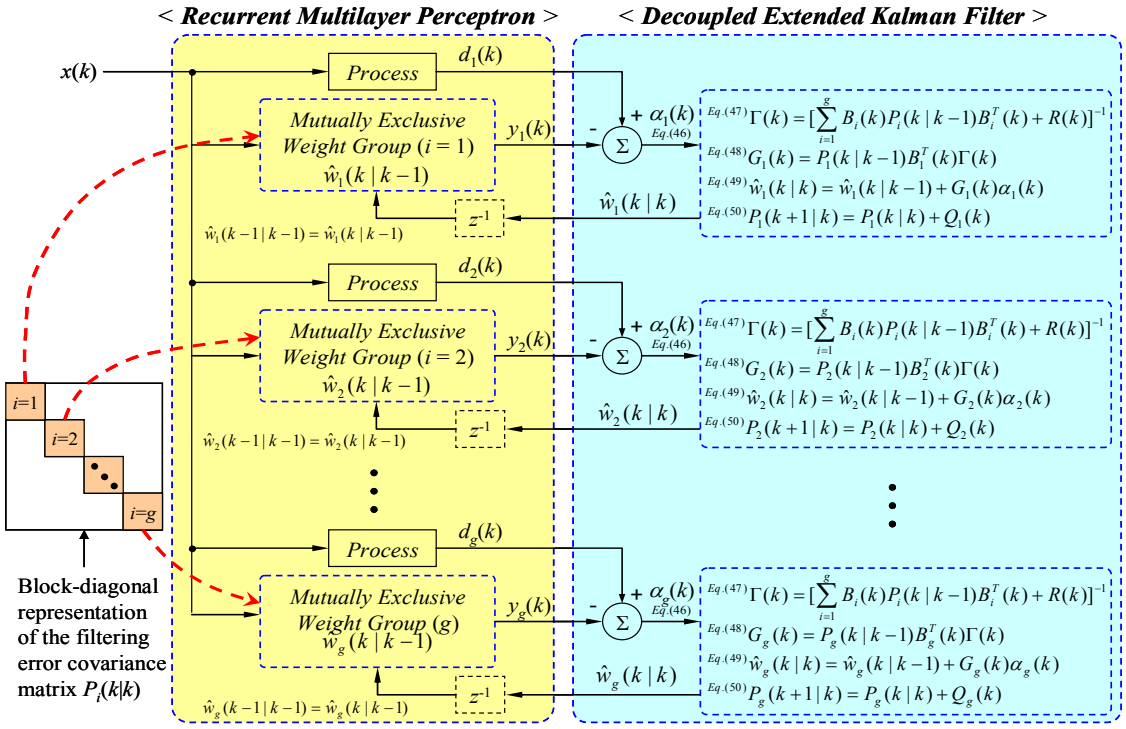

Figure 37. Decoupled Extended Kalman Filter (DEKF) for RNN.
Each group is corresponding to mutually exclusive weight group. The concatenation of the filtered weight vector $\hat{w}_i(k|k)$ forms the overall filtered weight vector $\hat{w}(k|k)$.

DEKF can reduce the computational complexity and its storage requirement of the EKF, but [145] restricts to a DEKF algorithm for which the weights are grouped by node. That sacrifices the computational accuracy because of omitting the interactions between the estimates of certain weights.

www.manaraa.com

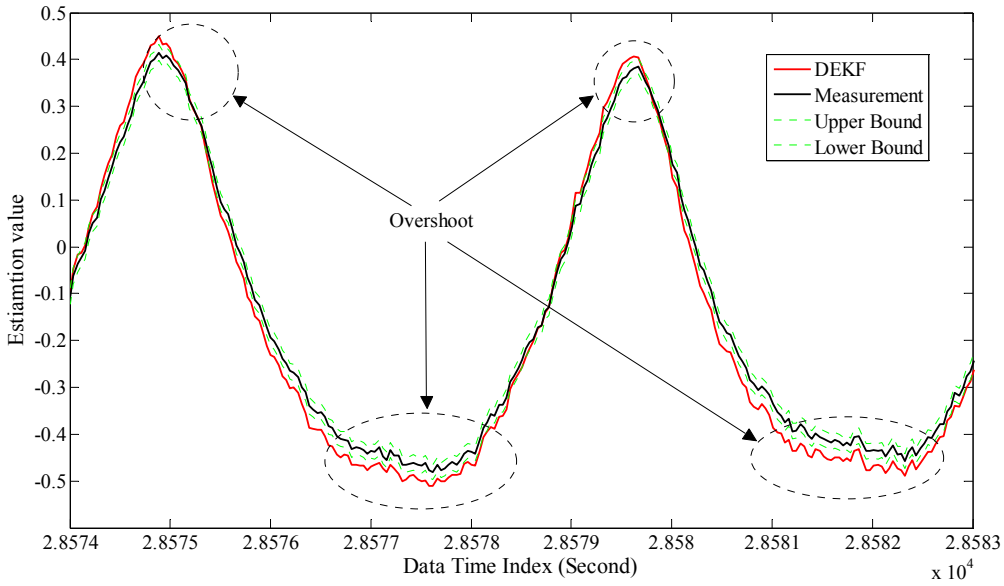

Figure 38. Prediction overshoots with DEKF.

To verify the prediction accuracy, we used a certain marginal value that can be explained in detail in Chapter 4.3.4. Fig. 38 shows the estimation of the respiratory motion with DEKF. As you can see in Fig. 38, we can notice that the percentage of prediction overshoot based on the marginal value is over 35%. That means we need a new approach to compensate the prediction accuracy with multiple input sequences. Therefore, we will show a hybrid motion estimation based on EKF (HEKF) in the next Chapter, which uses the channel number for the mutually exclusive groups and the coupling technique to compensate the computational accuracy.

4.3.2 Hybrid Estimation based on EKF for Neural Network (HEKF)

We have extended the DEKF into hybrid motion estimation based on EKF (HEKF). The author in [145] restricted to a DEKF algorithm for which the weights connecting inputs to a node are grouped together. If there are multiple input sequences with respect to time k, the complexity can increase by the power of the input number. To overcome

www.manaraa.com

computational complexity and estimation accuracy, we propose the coupling technique to compensate the computational accuracy using multiple sensory channel inputs. We refer to this newly proposed method as hybrid motion estimation based on EKF (HEKF).

There are two significant innovations for the proposed HEKF. The first innovation is to comprehensively organize the multiple channel sensory process by adapting the coupling technique. The second innovation is the multiple RMLPs with the simple neuron number for separate input channels. We first introduce the coupling matrix in Eq. (52), and then show the separate EKF process for each RMLP in Eq. (53) – (58).

Let c denote the designated channel number for the mutually exclusive groups. Here, each group is corresponding to an individual channel that is composed of position vector sequence with respect to time k. Also, for $i = 1, 2,\ldots, c$, let $\hat{w}_i^{CP}(k|k)$ be defined as filtered weight vector, $P_i^{CP}(k|k)$ and $G_i^{CP}(k)$ are subsets of the filtering-error covariance matrix and Kalman gain matrix for the channel i coupled with other channels, respectively.

Let $\Gamma^{CP}(k)$, $P_i^{CP}(k|k\text{-}1)$ be defined as $p\times p$ matrix denoting the global conversion factor for the coupled entire network, $s\times s$ prediction-error covariance matrix for the coupled EKF, respectively. Here, we also need to define the degree of coupling, μ_{ij} representing the degree to which component (i) depend on one another (j). Coupling matrix Π is $p\times p$ matrix containing all components of coupling degree. We can represent coupling matrix (Π) and coupling degree (μ_{ij}) as follows:

$$\Pi = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1p} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{p1} & \mu_{p2} & \cdots & \mu_{pp} \end{bmatrix}, \quad \sum_{j=1}^{p} \mu_{ij} = 1, \text{ for all } i. \tag{52}$$

www.manaraa.com

The closer to one the coupling degree is, the more tightly the channel i and j are coupled, *i.e. tight coupling*. If the coupling degree is close to zero, we can expect *loose coupling*. If μ_{ij} is corresponding to zero, there is no coupling with one another. For i = 1, 2,…, c, let define $\hat{w}_i^{CP}(k|k)$ as filtered weight vector, $P_i^{CP}(k|k)$ and $G_i^{CP}(k)$ are subset of the filtering-error covariance matrix and Kalman gain matrix for the channel number i, respectively. In light of these new notations, we can write the hybrid motion estimation based on EKF (HEKF) as follows:

$$\alpha_i^{CP}(k) = d_i(k) - [\sum_{j=1}^{p} \mu_{ij} \times y_j], \tag{53}$$

$$\Gamma^{CP}(k) = [\sum_{i=1}^{p}\sum_{j=1}^{p} \mu_{ij} B_i(k) P_i^{CP}(k \mid k-1)(B_i(k))^T + R(k)]^{-1}, \tag{54}$$

$$G_i^{CP}(k) = P_i^{CP}(k \mid k-1)(B_i(k))^T \Gamma^{CP}(k), \tag{55}$$

$$\hat{w}_i^{CP}(k+1 \mid k) = \hat{w}_i^{CP}(k \mid k-1) + G_i^{CP}(k)\alpha_i^{CP}(k), \tag{56}$$

$$P_i^{CP}(k+1 \mid k) = P_i^{CP}(k \mid k) + Q_i(k), \tag{57}$$

$$P_i^{CP}(k \mid k) = P_i^{CP}(k \mid k-1) - G_i^{CP}(k) B_i(k) P_i^{CP}(k \mid k-1), \tag{58}$$

where $\alpha_i^{CP}(k)$, $\Gamma^{CP}(k)$ and $P_i^{CP}(k+1|k)$ denote the difference between the desired response $d_i(k)$ for the linearized system and coupled estimations for the channel number i, the global conversion factor for the entire-coupled network, and the prediction-error covariance matrix for the coupled, respectively. In the case of HEKF, we have c identical networks for c input channels. Each input sequence is inserted into individual neural network process for each channel prediction, as shown in Fig. 39.

www.manaraa.com

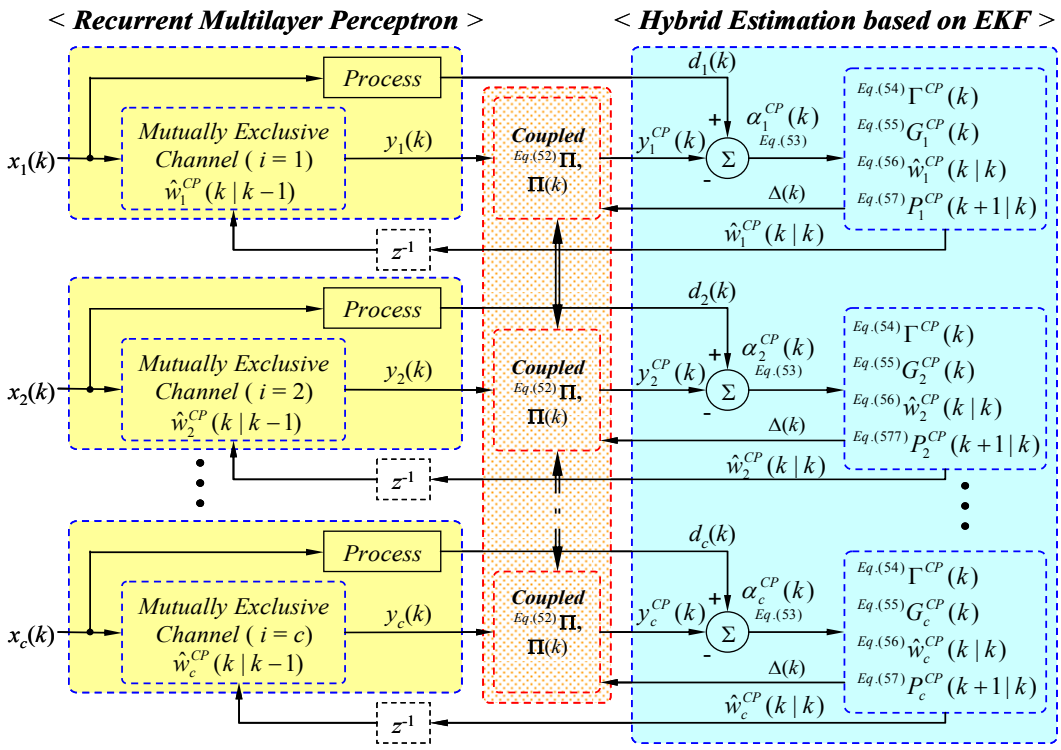

Figure 39. Hybrid motion estimation based on EKF (HEKF) for RNN.
Each group is corresponding to an individual channel that is composed of (x, y, z) position sequence with respect to the time step k.

4.3.3 Optimized Group Number for Recurrent Multilayer Perceptron (RMLP)

In DEKF algorithm the weights connecting inputs to a node are grouped together, whereas each group in HEKF algorithm corresponds to the individual channel that is composed of position vector sequence with respect to time k. In order to analyze the group number, we can incorporate Fisher Linear Discriminant on the discriminant analysis, which employs the *within-class scatter* value (S_W) and the *between-class scatter* value (S_B) in the given samples [206].

We have a set of n D-dimensional samples, which correspond to the filtering-error covariance matrices (P_i & P_i^{CP}) defined in Eq. (51) and (58) for each group i. Let m_i denote the D-dimensional sample mean for group i, and then define m_i as follows:

www.manaraa.com

$$m_i = \frac{1}{n_i}\sum_{j=1}^{n_i} P_i(j), \tag{59}$$

where n_i is the component number of group i. To obtain the optimization objective function, we define the scatter values S_i and S_W by

$$S_i = \sum_{P \in P_i} (P - m_i)^2, \tag{60}$$

and

$$S_W = \sum_{k=1}^{g} S_k, \tag{61}$$

where g is the number of group in the given samples. We define the *between-class scatter* value S_B as follows:

$$S_B = \sum_{i=1}^{g}\sum_{j=1}^{g} \left| m_i - m_j \right|^2 (i \neq j), \tag{62}$$

where g is the number of group in the given samples and m_i is not identical to m_j. In terms of S_B and S_W, the objective function $J(\cdot)$, called discriminant criterion, can be written by

$$J(g) = \arg\min_{g} \frac{S_W}{S_B}. \tag{63}$$

This criterion introduced expects that *within-class scatter* value should be minimized and the *between-class scatter* value should be maximized in the given number. Under the minimizing Eq. (63), we can get the optimized number of group (g) for RMLP by choosing the smallest $J(\cdot)$ with optimized group number (g). This value can be used to test the optimized number of RMLP between HEKF and DEKF. We can evaluate whether HEKF or DEKF could be more discriminated by comparing the objective function values $J(\cdot)$ as the discriminant degree at the selected (g).

www.manaraa.com

4.3.4 PREDICTION OVERSHOOT ANALYSIS

Here, we evaluate the performance of overshoot for the prediction values. We define overshoot for cases in which the predicted output exceeds a certain marginal value with confidence levels corresponding to the tolerances. We would like to derive such marginal value based on the estimate process of the uncertainty point estimators or predictors [184] [262] [263] [264] [265] [266] [267].

We noted in Eq. (35) in the previous Chapter that generally a neural network model can be represented as a nonlinear regressive function as follows:

$$y_i(k) = \Phi(x_i, \theta) + \varepsilon_i, \quad i = 1,2,...,n, \tag{64}$$

where x_i (with dimension $M\times1$) is input vector, and θ (with dimension $s\times1$) is a set of neural network true weights. It is assumed that ε_i are independent and identically-distributed with a normal distribution $N(0, \sigma^2)$. Let define $\hat{\theta}$ as the least square estimation of θ. In a small neighborhood θ the linear Taylor series expansion for the model (64) can be shown as follows [267]:

$$\hat{y}_i(k) = \Phi(x_i, \theta) + \Phi_0^T(\hat{\theta} - \theta), \quad i = 1,2,...,n, \tag{65}$$

where

$$\Phi_0^T = \left[\frac{\partial \Phi(x_i, \theta)}{\partial \theta_1} \quad \frac{\partial \Phi(x_i, \theta)}{\partial \theta_2} \quad \cdots \quad \frac{\partial \Phi(x_i, \theta)}{\partial \theta_s} \right]. \tag{66}$$

To construct marginal values for nonlinear regressive models in neural networks, the standard asymptotic theory should be applied. For the linear model in (65), an approximate marginal value (γ) with 100(1−α) confidence can be obtained [263] [267]:

$$\gamma = \pm t_{1-\alpha/2,n} \hat{\sigma} \sqrt{1 + \sum_{i=1}^{n} (F_i \cdot F_i^T)}, \tag{67}$$

www.manaraa.com

where $t_{1-\alpha/2,n}$ is the $1-\alpha/2$ quantile of a t–distribution function with n degrees of freedom, $\hat{\sigma}$ is the standard deviation estimator, and F_i is the Jacobian matrix of neural network outputs with respect to weights, respectively. $\hat{\sigma}$ and F_i are calculated as follows:

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \Phi(x_i, \hat{\theta})\right)^2}, \tag{68}$$

$$F_i = \left[\frac{\partial \Phi(x_i, \hat{\theta})}{\partial \theta_1} \quad \cdots \quad \frac{\partial \Phi(x_i, \hat{\theta})}{\partial \theta_s}\right], \tag{69}$$

In the experimental Chapter 4.4.5, we use this marginal value to judge whether the predicted outcomes exceed or not, and how many overshoots occur.

4.3.5 Comparisons on Computational Complexity and Storage Requirement

The computational requirements of the DEKF are dominated by the need to store and update the filtering-error covariance matrix $P(k|k)$ at each time step n. For a recurrent neural network containing p output nodes and s weights, the computational complexity of the DEKF assumes the following orders:

$$\text{Computational complexity: } O\left(p^2 s + p\sum_{i=1}^{g} s_i^2\right), \tag{70}$$

$$\text{Storage requirement: } O\left(\sum_{i=1}^{g} s_i^2\right), \tag{71}$$

where s_i is the size of the state in group i, s is the total state size, and p is the number of output nodes [231].

The computational requirements of the HEKF are also determined by the need to store and update the filtering-error covariance matrix P^{CP} at each time step n. In the HEKF, it needs to calculate the coupling matrix that contains all components of coupling degree as

www.manaraa.com

well. That means we need additional p^2 computation at each time step n. Therefore, the computational complexity of the HEKF assumes the following orders:

$$\text{Computational complexity: } O\left(p^2(s+1) + p\sum_{i=1}^{c} s_i^2 \right), \tag{72}$$

$$\text{Storage requirement: } O\left(p^2 + \sum_{i=1}^{c} s_i^2 \right), \tag{73}$$

where s_i is the size of the state in channel i.

Note that HEKF algorithm needs additional computation to calculate the coupling matrix, whereas the total computational complexity depends on the channel number c. The total computational complexity of the DEKF algorithm can be determined by the group number g. Here, we need to consider the group number and the channel number. If the group number is greater than the channel number (g > c) and the output node number is smaller than the size of the state in group i, ($p < s_i$), the HEKF algorithm can improve computational complexity with comparison to the DEKF algorithm. Note that this complexity analysis does not include the computational requirements for the matrix of dynamic derivatives.

When we compare HEKF and DEKF, the computational complexity of DEKF is recalculated as multiple channel numbers. When we use multiple channel numbers, the computational complexity of the DEKF assumes the following orders:

$$\text{Computational complexity: } O\left(p^2 sc + cp\sum_{i=1}^{g} s_i^2 \right), \tag{74}$$

$$\text{Storage requirement: } O\left(c\sum_{i=1}^{g} s_i^2 \right), \tag{75}$$

where c is the channel number. The computational complexities of the DEKF shown in Eq. (74) and (75) are larger than HEKF shown in Eq. (72) and (73).

www.manaraa.com

When we implement the proposed HEKF method by comparing it to the DEKF method, it is required to evaluate how much additional computational time. For the comparison on computational complexity, we have evaluated the performance of average CPU time in Experimental Chapter *F*. Here, we used three RMLPs for each channel in HEKF, whereas we used one RMLP in DEKF.

www.manaraa.com

4.4 Experimental Results

4.4.1 Motion Data Captured

We used three channel sets of patient breathing data to evaluate the filter performance. Each set of data consisted of chest displacements recorded continuously in three dimensions at a sampling frequency of 26*Hz*. The recordings lasted anywhere from 5 minutes to 1.5 hours of the average time at the Georgetown University Cyberknife treatment facility. These records were arbitrarily selected to represent a wide variety of breathing patterns, including highly unstable and irregular examples. Each patient's breathing record was used to independently train and test the predictive accuracy of the filter.

4.4.2 Optimized Group Number for RMLP

1) Optimized Group Number

With the respect to the selected group number (g) to implement the RMLP, we used a multilayer perceptron with two hidden layers, where the first hidden layer is recurrent and the second one is not. We increased the number of hidden units for the first and the second hidden layer according to the group number to calculate the objective function value for comparing two different methods. In order to analyze the group number for RMLP, we incorporated objective function (63) in Chapter 4.3.3.

As shown in Fig. 40, HEKF is optimized when the group number is 2, whereas DEKF is optimized when the group number is 6. Therefore, we choose the neuron number 2 for HEKF and 6 for DEKF.

www.manaraa.com

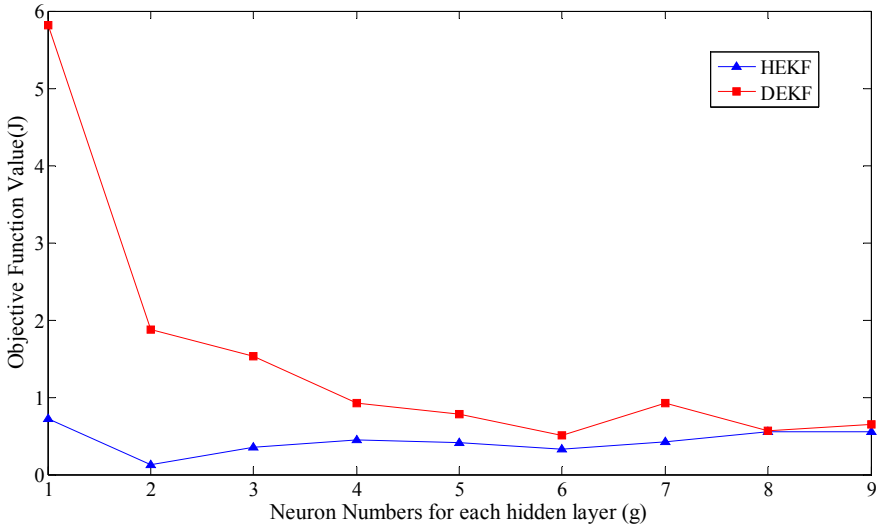

Figure 40. Comparison of objective function values between HEKF and DEKF.
With this figure, we can expect to choose the selected neuron number for HEKF or DEKF to be more optimized. Also, the discriminant criterion itself tests whether HEKF or DEKF is less enormous.

2) Discriminant Criterion to compare HEKF and DEKF

Using Fisher Linear Discriminant on the discriminant analysis in Chapter 4.3.3, we can expect that HEKF or DEKF could be more optimized by comparison with the objective function values $J(\cdot)$. Fig. 40 shows the objective function values $J(\cdot)$ defined in Eq. (63). HEKF has fewer values themselves than DEKF has, thus HEKF has more discriminated or further discriminant degree with comparison to DEKF across any group numbers selected, which means HEKF has less error than DEKF.

4.4.3 Prediction Overshoot Analysis

To evaluate the performance of overshoot for the prediction values, we derived the marginal value (γ) using Eq. (67) in Chapter 4.3.4. In this Chapter, we would like to use this marginal value to judge whether the predicted outcomes exceed or not, and how many overshoots occur. With the marginal value (γ), we can define the upper bound and

www.manaraa.com

the lower bound by adding the marginal value to the measurement value and subtracting the marginal value from the measurement value, respectively.

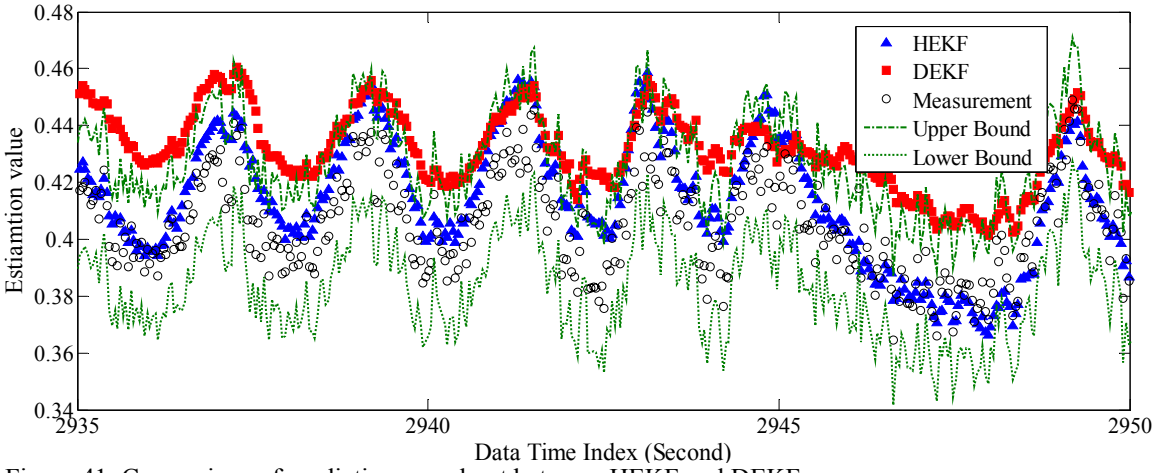

Figure 41. Comparison of prediction overshoot between HEKF and DEKF.
With this figure, we can notice that most of the estimation values of HEKF align between the upper bound and the lower bound, whereas the values of DEKF do not. After the transient state, we can evaluate that the average percentage of prediction overshoot for HEKF is 3.72%, whereas the average percentage of prediction overshoot for DEKF is 18.61%, thus HEKF has less prediction overshoot value than DEKF has.

Fig. 41 shows the comparison of prediction overshoots between HEKF and DEKF. As can be seen in Fig. 41, most of the estimation values of HEKF align between the upper bound and the lower bound. After the transient state, we can notice that the average percentage of prediction overshoot for HEKF is 3.72%, whereas the average percentage of prediction overshoot for DEKF is 18.61%. As can be seen in Table 11, most of the prediction overshoot of HEKF are within 5% except the datasets DB00, DB02, and DB03. We have also noticed that DEKF is slightly better than HEKF in the case of datasets DB13 and DB14, which include some discontinuities as well as the system noise because of the irregular patient breathing and the system latency during the breathing record [40]. We think these lacks of continuity could decrease the Kalman filter gain during the target prediction. In spite of these defect, however, the proposed HEKF can improve the average prediction overshoot by 62.95% with comparison to DEKF.

www.manaraa.com

Table 11. Prediction Overshoot Analysis (HEKF versus DEKF)

Datasets	# Total frames	HEKF (#Overshoot Frame / #Total Frame: %)	DEKF (#Overshoot Frame / #Total Frame: %)	Improvement (%)
DB00	79078	17.09	31.10	45.06
DB01	145336	3.23	7.72	58.13
DB02	140704	8.65	39.03	77.84
DB03	175896	5.24	6.35	17.52
DB04	93653	4.44	27.78	84.03
DB05	100739	4.14	26.55	84.41
DB06	159855	1.66	32.51	94.89
DB07	110417	0.12	41.50	99.70
DB08	225785	1.49	32.67	95.44
DB09	144149	0.27	0.40	31.75
DB10	185697	4.29	15.31	71.98
DB11	108327	0.95	12.01	92.11
DB12	129503	0.03	2.16	98.46
DB13	146145	0.53	0.51	-4.15
DB14	134683	3.59	3.49	-2.94

4.4.4 Comparison on Estimation Performance

We evaluate the target estimation by comparing the proposed HEKF described in Chapter 4.3.2, with the alternative DEKF described in Chapter 4.3.1.

1) Tracking Position Estimation

Fig. 42 shows the average target position estimation of the 3D Euclidian distance between the predicted value and the measurement values with respect to the data time index given by the original Cyberknife dataset. The unit of vertical axis in Fig. 42 and Fig. 43 is dimensionless for the amplitude, *i.e.* the target estimation corresponds to the 3D position has the range of [−1, +1], corresponding to the real measurement dataset range $[-1.4735\times10^{3}, -1.5130\times10^{3}]$. As you can see, the position estimation values of HEKF align closer to the measurement values than DEKF values.

www.manaraa.com

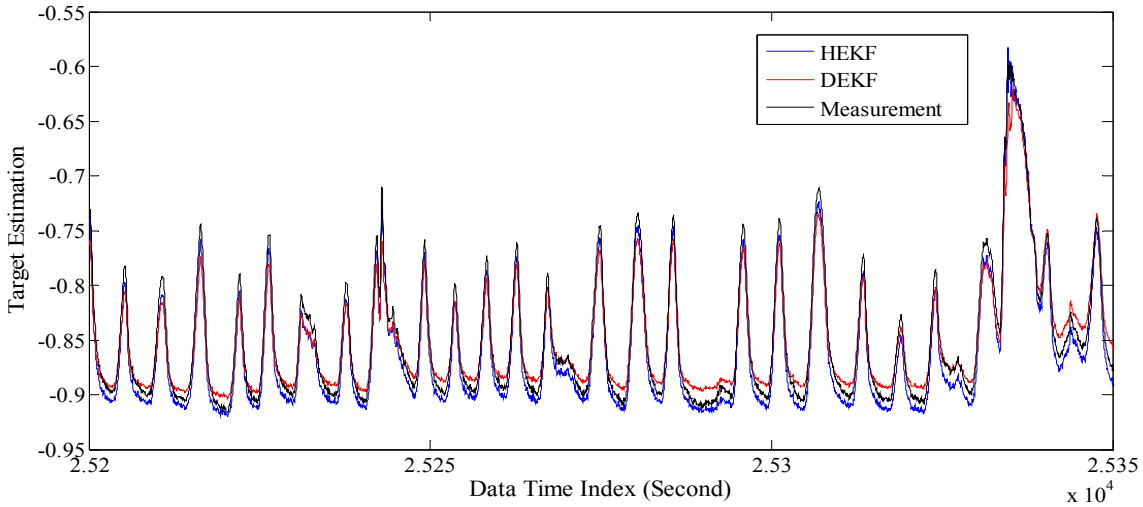

Figure 42. Target estimation between HEKF and DEKF.
This figure shows that the position estimation values of HEKF align closer to the measurement values than DEKF values.

2) Position Error Value

We would like to compare the performance of tracking errors with respect to the data time index across the entire measurement period between HEKF and DEKF. The error value in Fig. 43 was calculated by the subtraction of the 3D Euclidian distance between the predicted values and the measurement values in the data time index.

Fig. 43 shows that the error value of HEKF is smaller than that of DEKF across the data time index 25200 ~ 25350 sec. At the beginning of tracking estimation, we notice that both approaches have several overshoot across the data time because of the unstable initialization of the original dataset. After the steady state, the error value of HEKF aligns more close to zero point. Two significant position errors are shown in DEKF, whereas the position error is negligible in HEKF.

www.manaraa.com

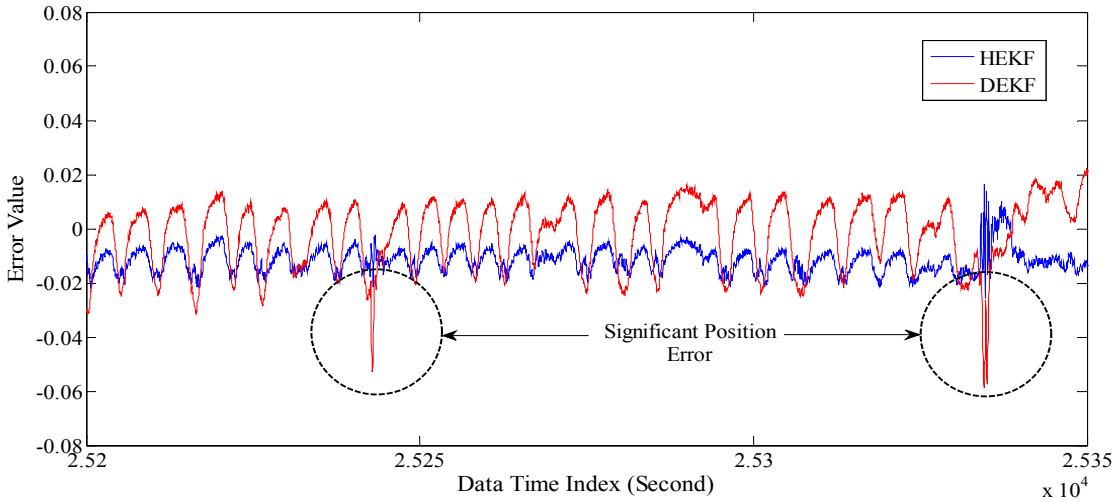

Figure 43. Comparison of position error between HEKF and DEKF.
This figure shows two significant position errors in DEKF, whereas the position error is negligible in HEKF.

4.4.5 Error Performance over Prediction Time Horizon

Prediction Time Horizon is the term to represent the time interval window to predict the future sensory signal. We would like to compare the error performance among the various prediction time horizon between HEKF and DEKF in Table 12. For the comparison, we used a normalization that is the normalized root mean squared error (NRMSE) between the predicted and actual signal over all the samples in the test dataset, as follows [36]:

$$NRMSE = \sqrt{\sum_i (y_i - \hat{y}_i)^2 \Big/ \sum_i (y_i - m_y)^2}\,, \tag{76}$$

where y_i is the i^{th} measurement, $\hat{y}_i$ is the estimation of the i^{th} measurement, and m_y is the mean of all the measurements. This metric is dimensionless and allows us to compare prediction accuracy for different signals of widely varying amplitude.

As can be seen in Table 12, the error performance in the proposed HEKF has improved for all the datasets by 26.65% in the average of the prediction time horizon for 38.46*ms*. The prediction interval time has increased and the calculated NRMSE has increased.

www.manaraa.com

Notice that the 7 datasets are shown in the bold fond since the improvement of error performance for the proposed method maintained over 25 %, with 50% across the prediction time horizons in datasets DB01, DB03, DB07, and DB12. Compared to the patient of the Cyberknife dataset in the latest research [254], the proposed HEKF showed the better NRMSE performance across all variable prediction interval times; for example at the prediction time horizon of 500 *ms*, a 422% NRMSE improvement.

Table 12. Error Performance among Prediction Time Horizon (HEKF versus DEKF)

	Prediction Time Horizon						
	38.46*ms*	115.38 *ms*	192.3 *ms*	269.23 *ms*	346.15 *ms*	423.07 *ms*	500 *ms*
DB00	0.0666/0.0706	0.0714/0.0768	0.0740/0.0782	0.0752/0.0847	0.0790/0.0875	0.0812/0.0850	0.0848/0.0890
DB01	**0.0326/0.0739**	**0.0365/0.0771**	**0.0420/0.0876**	**0.0463/0.1015**	**0.0466/0.1085**	**0.0504/0.1087**	**0.0820/0.1134**
DB02	0.0961/0.1347	0.1128/0.1395	0.1306/0.1419	0.1331/0.1450	0.1333/0.1540	0.1349/0.1637	0.1458/0.1821
DB03	**0.0535/0.0896**	**0.0545/0.0917**	**0.0560/0.1122**	**0.0576/0.1260**	**0.0593/0.1342**	**0.0616/0.1348**	**0.0796/0.1519**
DB04	**0.0440/0.0661**	**0.0503/0.0674**	**0.0589/0.0719**	**0.0613/0.0724**	**0.0638/0.0775**	**0.0668/0.0991**	**0.0672/0.1138**
DB05	**0.0468/0.0789**	**0.0546/0.0830**	**0.0563/0.0863**	**0.0574/0.0907**	**0.0583/0.0947**	**0.0675/0.0966**	**0.0696/0.0987**
DB06	0.0265/0.0304	0.0279/0.0338	0.0304/0.0338	0.0340/0.0366	0.0361/0.0382	0.0388/0.0399	0.0409/0.0449
DB07	**0.0311/0.0864**	**0.0423/0.0941**	**0.0442/0.0957**	**0.0501/0.0959**	**0.0555/0.0993**	**0.0608/0.1333**	**0.0755/0.1444**
DB08	0.0555/0.0606	0.0590/0.0636	0.0621/0.0691	0.0737/0.0783	0.0763/0.0792	0.0816/0.0880	0.0866/0.0921
DB09	**0.1018/0.1123**	**0.1104/0.1305**	**0.1460/0.1712**	**0.1825/0.2283**	**0.1875/0.2928**	**0.1886/0.3428**	**0.1926/0.3556**
DB10	0.1010/0.1064	0.1078/0.1146	0.1133/0.1238	0.1251/0.1344	0.1342/0.1474	0.1495/0.1638	0.1786/0.1906
DB11	0.0731/0.0987	0.1106/0.1237	0.1209/0.1293	0.1358/0.1377	0.1574/0.1586	0.1588/0.1638	0.1770/0.1797
DB12	**0.0424/0.0916**	**0.0459/0.0958**	**0.0470/0.0977**	**0.0474/0.0998**	**0.0504/0.1021**	**0.0505/0.1034**	**0.0630/0.1045**
DB13	0.0651/0.0788	0.0668/0.0811	0.0678/0.0815	0.0687/0.0839	0.0691/0.0908	0.0710/0.0948	0.0732/0.1036
DB14	0.0440/0.0455	0.0456/0.0509	0.0482/0.0511	0.0490/0.0519	0.0505/0.0520	0.0515/0.0538	0.0541/0.0575

www.manaraa.com

4.4.6 COMPARISONS ON COMPUTATIONAL COMPLEXITY

We would like to evaluate how much additional computational time is required when we implement the proposed HEKF method by comparing to DEKF method. For HEKF, we used three RMLPs for each channel, whereas we used one RMLP for DEKF, where the neuron number for the first and the second hidden layer is 2 for HEKF and 6 for DEKF, respectively. Regarding CPU experimental time, we have evaluated the overall performance of average CPU time, using a PC of Pentium core 2.4 GHz with RAM 3.25 GB.

Table 13. CPU Time Used in the Target Estimation

Datasets	Recording time (minutes)	CPU Time used (Millisecond / #Total Frame)	
		HEKF	DEKF
DB00	50.80	9.4306	7.1737
DB01	93.36	9.7759	7.2836
DB02	90.39	10.8872	7.1532
DB03	113.00	10.8578	6.9824
DB04	60.16	10.0511	7.1556
DB05	64.85	10.3541	7.3941
DB06	102.83	10.5332	7.1505
DB07	70.93	9.4372	6.7484
DB08	145.21	11.2489	7.1755
DB09	92.67	10.3379	7.0038
DB10	119.55	11.3506	7.2783
DB11	69.72	9.5831	7.0640
DB12	85.34	9.6143	6.8265
DB13	93.88	11.2510	7.4613
DB14	86.52	9.5256	7.5890

Table 13 shows the performance of CPU time used. As you can see in Table 13, HEKF method needs more time comparing to DEKF. We think that the actual difference for CPU time used in Table 13 mainly comes from the calculation of the coupling matrix and the separate neural network for channel number. Although 30.07% more time is required

www.manaraa.com

to implement the proposed HEKF, it is a modest tradeoff to consider the better performance than better computational time under the condition that PC speed is improving these days.

www.manaraa.com

4.5 SUMMARY

In this Chapter we have presented respiratory motion estimation with hybrid implementation of EKF, called HEKF. Our new method has two main contributions to improve the traditional EKF-based recurrent neural network target tracking. The first contribution is to present a new approach to split the whole RMLP with the complicated neuron number into a couple of RMLPs with the simple neuron number to adjust separate input channels. The second contribution is to comprehensively organize the multiple channel sensory process by adapting the coupling technique using multiple channel inputs for the mutually exclusive groups to compensate the computational accuracy.

The experiment results validated that the prediction overshoot of the proposed HEKF was improved for 13 datasets among 15 datasets by 62.95%. The proposed HEKF showed the better performance by 52.40% NRMSE improvement in the average of the prediction time horizon. We have evaluated that a proposed HEKF can outperform DEKF by comparing the performance of tracking estimation value, NRMSE and prediction overshoot analysis. Moreover, HEKF has more discriminated degree with comparison to DEKF across any group numbers selected, which means HEKF has less error than DEKF. Even though the provided method needed more computational time comparing to the previous method, the experiment results showed that it improved NRMSE around 24.72% across the overall prediction time horizon.

www.manaraa.com

CHAPTER 5 CUSTOMIZED PREDICTION OF RESPIRATORY MOTION

Accurate prediction of the respiratory motion would be beneficial to the treatment of thoracic and abdominal tumors. However, a wide variety of breathing patterns can make it difficult to predict the breathing motion with explicit models. We proposed a respiratory motion predictor, *i.e.*, customized prediction with multiple patient interactions using neural network (CNN). For the preprocedure of prediction for individual patient, we construct the clustering based on breathing patterns of multiple patients using the feature selection metrics that are composed of a variety of breathing features. In the intraprocedure, the proposed CNN used neural networks (NN) for a part of the prediction and the extended Kalman filter (EKF) for a part of the correction. The prediction accuracy of the proposed method was investigated with a variety of prediction time horizons using normalized root mean squared error (NRMSE) values in comparison with the alternate recurrent neural network (RNN). We have also evaluated the prediction accuracy using the marginal value that can be used as the reference value to judge how many signals lie outside the confidence level. The experimental results showed that the proposed CNN can outperform RNN with respect to the prediction accuracy with an improvement of 50 %.

www.manaraa.com

5.1 INTRODUCTION

Current developments in radiotherapy systems open a new era for treatment with accurate dosimetry of thoracic and abdominal tumors [1] [23] [24]. Effective radiation treatment requires motion compensation for uncertainty and irregularity originating from systematic or random physiological phenomena [19] [269]. Respiratory motion severely affects precise radiation dose delivery because thoracic and abdominal tumors may change locations by as much as three centimeters during radiation treatment [2]. In patients with a wide range of respiratory motion, radiation treatment can be delivered by dynamic gating, where radiation is activated only when the respiratory motion is within a predefined amplitude or phase level [2] [25].

In addition to the respiratory motion, system latency attributable to hardware limitations and software processing time may affect the accurate radiation delivery for tumor tracking techniques [1] [26] [27]. If the acquisition of tumor position and the repositioning of the radiation beam are not well synchronized, a large volume of healthy tissue may be irradiated unnecessarily and tumor may be underdosed [20] [21]. Due to the latency, for real-time tumor tracking, the tumor position should be predicted in advance, so that the radiation beams can be adjusted accordingly to the predicted target position during treatment [1] [9]. Therefore, we propose a prediction method for respiratory motion to compensate for uncertainty in respiratory patterns with the correlation of patients breathing datasets.

A number of prediction methods for respiratory motion have been investigated based on surrogate markers and tomographic images [2] [4] [5] [6] [9] [11] [13] [14] [15] [16] [17] [28] [48] [96] [254] [270]. The previous methods can be further categorized into two

www.manaraa.com

approaches: 1) those that are "model-based," which use a specific biomechanical or mathematical model for respiratory motion functions or models [4] [5] [17] [48] [96]; and 2) those that are "model-free" heuristic learning algorithms that are trained based on the observed respiratory patterns [14] [36] [254]. Generally, model-based methods include linear approaches and Kalman filter variables that are widely used for the fundamental prediction of respiratory motion among a variety of investigated methods [4] [5] [96].

A potential drawback of model-based approaches is their inability to learn highly irregular breathing patterns from training samples [36]. For accurate prediction of respiratory motion, the breathing pattern information should apply the respiratory motion prediction to improve prediction accuracy [34]. Based on previous studies, the model-free heuristic learning algorithm can be a key approach for prediction; but, it needs a correction method to compensate for irregular breathing signals that characterized a variety of breathing patterns. Accordingly, we have pursued the use of heuristic algorithms to develop system adaptive loops that have the most general approach, *i.e.*, neural networks (NN).

The contribution of this study is to adopt a clustering method for multiple patients to get more practical breathing pattern information and to find an accurate prediction process for an individual class. For the clustering based on breathing patterns, we present the feature selection metrics. With each feature metric, we can define a variety of feature combinations and select an optimal feature combination, *i.e.*, dominant feature selection ($\hat{I}$), and then we can select the appropriate class number ($\hat{c}$) for the analysis of breathing patterns of multiple patients. Finally, we can predict the respiratory motion based on

www.manaraa.com

multiple patient interactions, *i.e.*, class-based respiratory motion prediction using interactive degree and neuron number selection of RNN.

www.manaraa.com

5.2 Prediction Process for Each Patient

For the respiratory motion prediction, we propose to use a supervised-training feedback system as shown in Fig. 44. The computational complexity of the EKF depends on the requirement capacity to store and update the filtering-error covariance matrix. If an RNN has p output nodes and s weights, the asymptotic growth rate for the computational complexity and storage requirement of the network would be proportional to output nodes (p) and weights to the second power (s^2). Here, output nodes and weights correspond to the patient number for predicting the respiratory motion and the state number for the prediction process. For large weights s, we may need highly demanding computational resources for these requirements to predict respiratory motions. We may partially release such requirements by using the Decoupled Extended Kalman filter (DEKF) as a practical remedy to overcome computational limitations with the computational complexity of an order of $p\times(s/p)^2$ [145] [146].

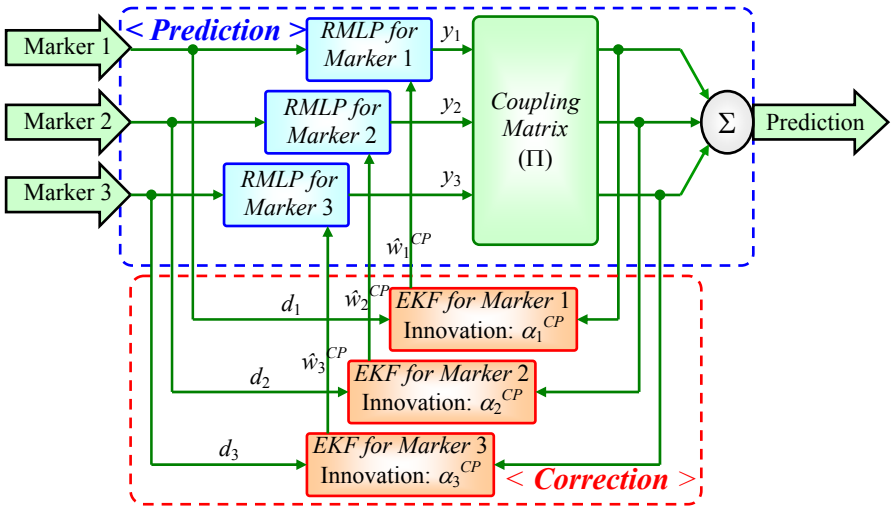

Figure 44. Multiple marker interactions for the individual patient.
The respiratory motion prediction for each patient is composed of the prediction process and the correction process. The prediction process is comprehensively organized with the multiple markers by adapting the coupling Matrix.

www.manaraa.com

The key idea of the DEKF is to use interactive state estimates of certain weight groups based on the neural node in such a way that the prediction process operates so-called mutually exclusive weight groups in the recurrent network [145]. That leads to the impairing of the computational accuracy of predicting respiratory motions based on the recurrent network because it ignores interactions of excluded weight states. Therefore, we propose a prediction process for each patient based on RNN using a coupling matrix, in which we adapt the coupling technique to comprehensively organize state estimates of multiple markers for predicting respiratory motions. That approach creates multiple recurrent multilayer perceptron (RMLP) as a part of predictive excitation for separate input markers in Fig. 44.

In Fig. 44 we denote the marker number (i) as the designated marker number for the mutually exclusive groups, where an individual RMLP corresponds to each marker that consists of breathing motion vectors (three-dimensional coordinates) with time sequence k. After finishing the first step of the prediction process for each marker, we define the innovation process $\alpha_i^{CP}(k)$ (53) and the filtered weight vector $\hat{w}_i^{CP}(k)$ (55), as shown at the EKF block for each marker in Fig. 44.

For the interactive process of multiple markers, we use the coupling degree μ_{ij} representing the degree to which component (i) depend on one another (j), as shown at the coupling matrix block in Fig. 44. The coupling degree μ_{ij} and the coupling matrix Π with $p \times p$ matrix including all components of coupling degree can be defined using Eq. (52). We may expect combined relationships between marker i and j if the coupling degree μ_{ij} is close to one, *i.e.*, tight coupling, whereas we may expect released relationships if the coupling degree is far from one, *i.e.*, loose coupling. With these

www.manaraa.com

coupling effects in mind, the prediction system for multiple patients should organize the whole respiratory motion datasets into some specific breathing motions that associate together in a group based on the respiratory patterns. For such associate processes of the multiple patient interactions, we would like to analyze respiratory patterns and extract usable prediction parameters which are repeatedly utilized in the training data of a group in a manner going back to the learning process of the respiratory prediction.

www.manaraa.com

5.3 Proposed Filter Design for Multiple Patients

This Chapter explains the detailed modeling prediction process based on the breathing patterns of multiple patients. The procedure for the interactive prediction consists of the preprocedure (interactive process for multiple patients) and the intraprocedure (prediction and correction process). We show the interactive process for multiple patients in Fig. 45.

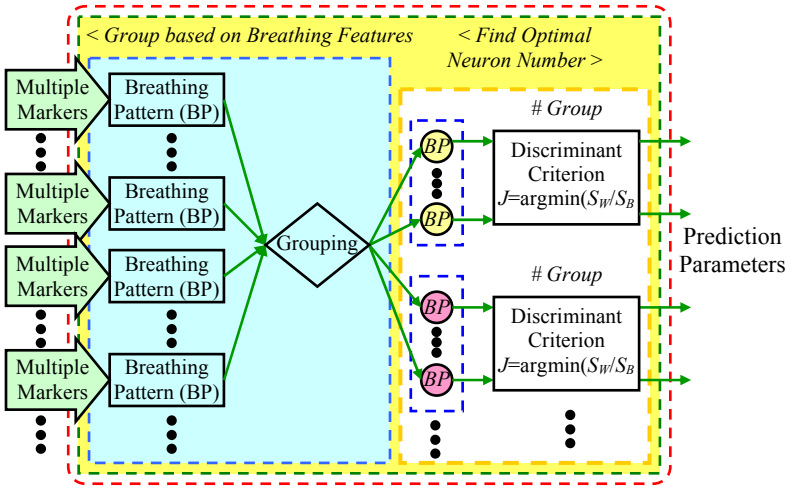

Figure 45. Interactive process for multiple patients.
Here, a multiple markers input in Fig 45 corresponds to three markers in Fig. 44. The preprocedure (interactive process for multiple patients) can provide the clustering of breathing pattern based on multiple patients and the prediction parameters for each class.

In the preprocedure we would like to get the clustering of respiratory motion based on the breathing patterns of the multiple patients. After the clustering, each class can have the prediction parameters (neuron number for prediction and coupling parameters) for each class. The intraprocedure corresponds to the prediction process for each patient in Fig. 44. With the prediction parameters of the preprocedure, the intraprocedure can operate to predict the respiratory motion of each patient. Chapters 5.3.1 and 5.3.2 explain the clustering method for the group, based on breathing patterns and how to find an optimal neuron number of the prediction process for each class, respectively.

www.manaraa.com

5.3.1 GROUPING BREATHING PATTERN FOR PREDICTION PROCESS

Fig. 45 illustrates the interactive process, involved in forming a clustering based on the breathing patterns of multiple patients. For the first step of CNN, we need to classify the breathing patterns of multiple patients. To extract the breathing patterns, we show feature selection metrics in Table 14. Murthy *et al* showed that the breathing stability can be quantified by autocorrelation coefficient and delay time [254]. Respiratory motion signal may be represented by sinusoidal curve [37] so that each breathing pattern can have variable measurements of breathing signal amplitude including acceleration, velocity, and standard deviation [148]. The typical vector-oriented feature extraction, exemplified by principal component analysis (PCA) and multiple linear regressions (MLR), has been widely used [271] [272]. Table 14 shows the feature selection metrics for the clustering of breathing patterns. Breathing frequency also showed diversity in individuals [269]. We create Table 14 based on previous existences of breathing features, so that the table can be variable. We randomly selected 7800 sampling frames (five minutes) for the feature extraction with three marker breathing datasets of each patient.

Table 14. Feature selection metrics with description

Index (x, y, z)	Name	Description
1	AMV	Autocorrelation MAX value
2	ADT	Autocorrelation delay time
3	ACC	Acceleration variance value
4	VEL	Velocity variance value
5	BRF	Breathing Frequency
6	FTP	Max Power of Fourier transform
7	PCA	Principal Component Analysis Coefficient
8	MLR	Multiple Linear Regression Coefficient
9	STD	Standard deviation of time series data
10	MLE	Maximum Likelihood Estimates

www.manaraa.com

We pick up feature extraction criteria that are currently available for the breathing patterns in the previous works [37] [148] [254] [271] [272]. The feature extraction criteria listed in Table 14 may be duplicated, but we introduce the following discriminant criteria to find out the most reliable feature set, *e.g.* dominant feature vector $I=(I_x, I_y, I_z)$, as three coordinate combinations selected from 10 feature metrics, where I_x, I_y and I_z correspond to each of the 10 feature metric values indexed in Table 14, so that we can have ${}_{10}C_3$ (=120) feature combination vectors. The feature metrics for the appropriate clustering of breathing patterns have yet to be determined. The objective of this Chapter is to select the effective feature combination metric ($\hat{I}$) from the candidate feature combination vector (I). For the selection of the estimated feature metrics, we use the objective function based on clustered degree using *within-class* scatter (S_W) and *between-class* scatter (S_B) [206]. Here, the S_W is proportional to the number of class (c) and the covariance matrix of feature samples based on each class. Accordingly, the S_W can be expressed as $S_W = c\times\Sigma^c_{i=1}(S_i)$, where c is the number of class and S_i is the covariance matrix based on feature combination vectors in the i^{th} class. The S_B is proportional to the covariance matrix of the mean (m_i) for the feature combination vector and can be expressed as $S_B = \Sigma^c_{i=1}(n_i\times(m_i-m)^2)$, where n_i is the sample number of the feature combination vector in the i^{th} class. m_i and m are means of the total feature combination vector and the feature combination vector in the i^{th} class, respectively.

Finally, the objective function J based on the S_W and the S_B to select the optimal feature combination vector can be written as $J(I, \hat{c}) = \text{argmin}(S_W/S_B)$, where I is the candidate feature combination vector for breathing patterns clustering based on the given feature selection metrics, and $\hat{c}$ is the estimated class number to get the minimum value of the

www.manaraa.com

objective function. To select the optimal combination experimentally, we calculate the objective function ($J(\cdot)$) with fixing the candidate feature combination vector (I) and increasing the class number c (in our simulation from 2 to 7) in the following equation:

$$\hat{I} = \arg\min_{I} H(I), \quad H(I) = \sum_{c=2}^{7} J(I, c). \tag{77}$$

With the above equation, we can select the estimated feature combination vector ($\hat{I}$) from the candidate feature combination vector (I) with the minimum value of Eq. (77). In the experimental Chapter 5.4.2, we will show how to select the estimated feature combination vector ($\hat{I}$) with our simulation results, followed by the estimated number of classes as c.

5.3.2 Neuron Number Selection

After grouping based on the breathing patterns, we find the optimal neuron number for each group using the Fisher Linear Discriminant [206]. We can design the RMLP with multiple hidden layers based on the specific application. In addition, we need to find an optimal hidden neuron number to design for multiple layers so that we can make the proper RMLP design to minimize the calculation cost and to maximize the prediction accuracy. The objective of this Chapter is to select the proper neuron number for hidden layers from a set of n D-dimensional samples identical to the filtering-error covariance matrices for each group. After calculating the D-dimensional sample means for each group, we can obtain the optimization objective function $J(g)$ based on the Fisher Linear Discriminant as $J(g)$=argmin(S_W/S_B), where g is the number of groups in the given samples.

www.manaraa.com

The criterion based on $J(g)$ reminds us that the filtering-error covariance matrices within each group should be minimized and the filtering-error covariance matrices between groups should be maximized in the given number [206]. With the objective function $J(g)$ in mind, we can find the optimized number of group (g) for the respiratory prediction in the recurrent network in a manner selecting the smallest $J(\cdot)$ as the optimized group number (g). We may decide that the proposed prediction method could be more discriminated by comparing the objective function values $J(\cdot)$ as the discriminant degree at the selected (g) [7]. This value can be incorporated to train recurrent networks and predict respiratory motions of multiple patients for the proposed prediction process.

www.manaraa.com

5.4 Experimental Results

5.4.1 Breathing Motion Data

For the prediction of respiratory motion, we used patient breathing datasets recorded at the Georgetown University CyberKnife treatment facility. Each breathing recording has three marker breathing datasets, with a 26*Hz* sampling frequency, where each maker has three-coordinates. That means potential inputs are as follows: (x_1, y_1, z_1), (x_2, y_2, z_2), and (x_3, y_3, z_3). The output is the position of breathing motion corresponding to 3-coordinates. The total 130 patients breathing recordings are randomly selected so that breathing datasets can be mixed up with highly unstable and irregular breathing motions.

Table 15. The characteristics of the breathing datasets

Total Patients	Average Records	Minimum Records	Maximum Records
130	66 minutes	25 minutes	2.2 hours

Table 15 shows the characteristics of the breathing datasets. The breathing recording times average 66 minutes in duration, where the minimum and the maximum recording times are 25 minutes and 2.2 hours, respectively. Each patient's recording was used to train and predict respiratory motion. We used 5 minute sampling data for the feature extraction.

5.4.2 Feature Selection Metrics

We can derive 120 ($=_{10}C_3$) feature combination vectors, *i.e.*, choose three out of the 10 features defined in Table 14, so that we can span three axis vectors corresponding to the features chosen (shown in the next Chapter). As shown in the following figure, using

www.manaraa.com

results of the minimum value of $H(I)$, we can select the combination number (105, 106, 107, 108, 109, 110, 117, 118, 119, 120) corresponding to the estimated feature combination vectors ($\hat{I}$), *i.e.*, the feature combinations with Breath Frequency (BRF), Principal Component Coefficient (PCA), Maximum Likelihood Estimates (MLE), Multiple Linear Regression Coefficient (MLR), and Standard Deviation (STD). This result also confirms that the three chosen axes can provide the distinct discriminate feature distribution.

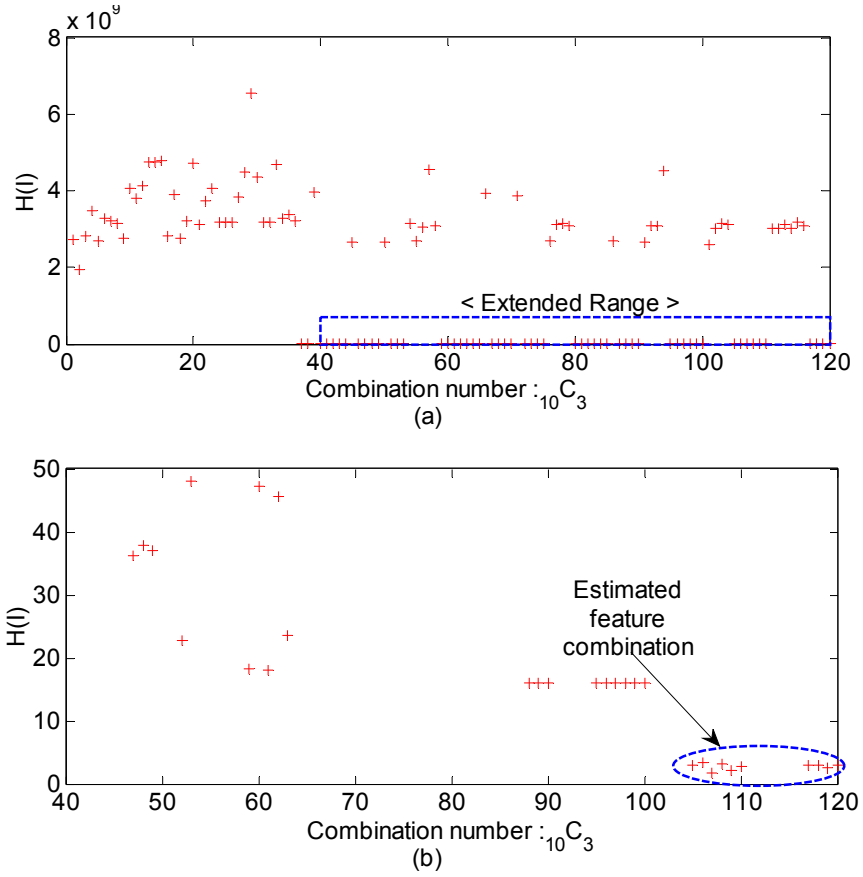

Figure 46. Dominant feature selection with Feature Combination Vector.
(a) whole range and (b) extended range. We can define 120 feature combination vectors (I) with 10 feature selection metrics as shown in Table 14 and select the dominant feature combination vectors ($\hat{I}$) with the minimum value of H(I), *i.e.* the feature combinations with Breath Frequency (BRF), Principal Component Coefficient (PCA), Maximum Likelihood Estimates (MLE), Multiple Linear Regression Coefficient (MLR) and Standard Deviation (STD).

www.manaraa.com

Now, we would like to choose the class number (c) with the minimum value of the objective function ($J(c)$). The figure shows the clustering of the estimated feature combination vector ($\hat{I}$) with respect to the class number (c=2,…, 7). We calculate the objective function value ($J(c)$) with a different class number. The class number (c=5) is chosen to minimize the criterion J with the corresponding class.

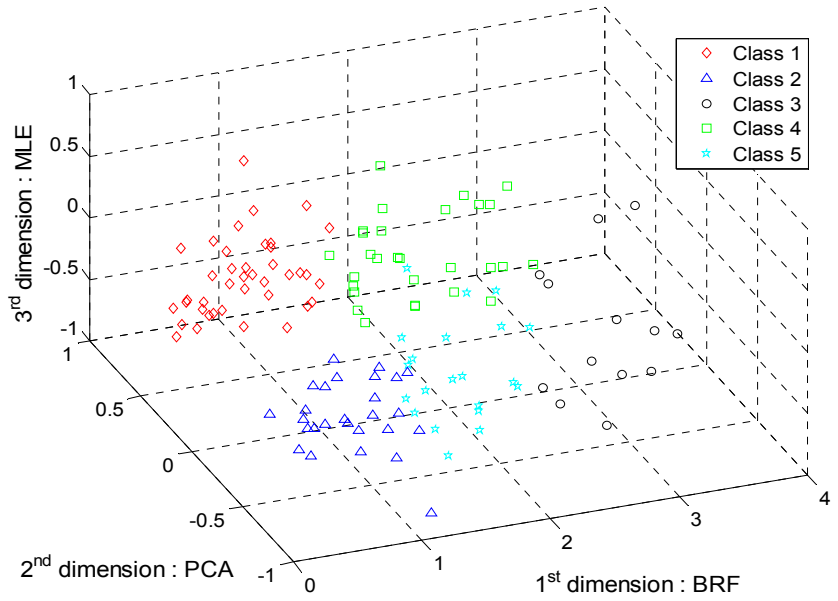

Figure 47. Clustering of 130 patients datasets with the dominant class number.
After calculating the objective function value ($J(c)$) with different class number, we can select the dominant class number with the minimum value of $J(c)$. Therefore, with the dominant class number (c=5), we can make the clustering of 130 patients datasets in Fig. 47. Here, each class (*i.e.* 1, 2, 3, 4, and 5) has different number of patients (*i.e.* 40, 27, 13, 29, and 21, respectively).

With increasing the cluster number (c), the estimated class number ($\hat{c}$) is selected to get the minimum of the objective function value ($J(c)$). We can notice that the objective function has the minimum when c = 5, as the estimated class number ($\hat{c}$) to 5. Now, we can make the clustering with the estimated class number ($\hat{c}$ = 5). Accordingly, for the clustering with 130 patient datasets, we have made a clustering with the feature combination vector (BRF, PCA, and MLE) and the estimated class number ($\hat{c}$ = 5). That means 130 patient datasets are placed into five classes as shown in Fig. 47. For the

www.manaraa.com

prediction process, each class has prediction parameters, *i.e.*, the optimal neuron number for RMLP based on Fisher Linear Discriminant (explained in Chapter 5.3.2) and coupling parameters Eq. (52) that can be experimentally derived for each class.

5.4.3 Comparison on Estimation Performance

We have evaluated the estimation of the respiratory motion by comparing the proposed method CNN with the alternative recurrent neural network (RNN). For the RMLP implementation of the proposed CNN, we used a multilayer perceptron with two hidden layers, where the first hidden layer is recurrent and the second one is not. Each hidden layer has two hidden neurons that were chosen based on the Fisher Linear Discriminant (in Chapter 4.3.3). For the alternate RNN analyzed in this study, we used two hidden layers with nine input neurons and one output neuron, where each input neuron is corresponding to one coordinate of three-dimensional position. For the network training on both methods, we used 3000 sampling frames with 26 *Hz*.

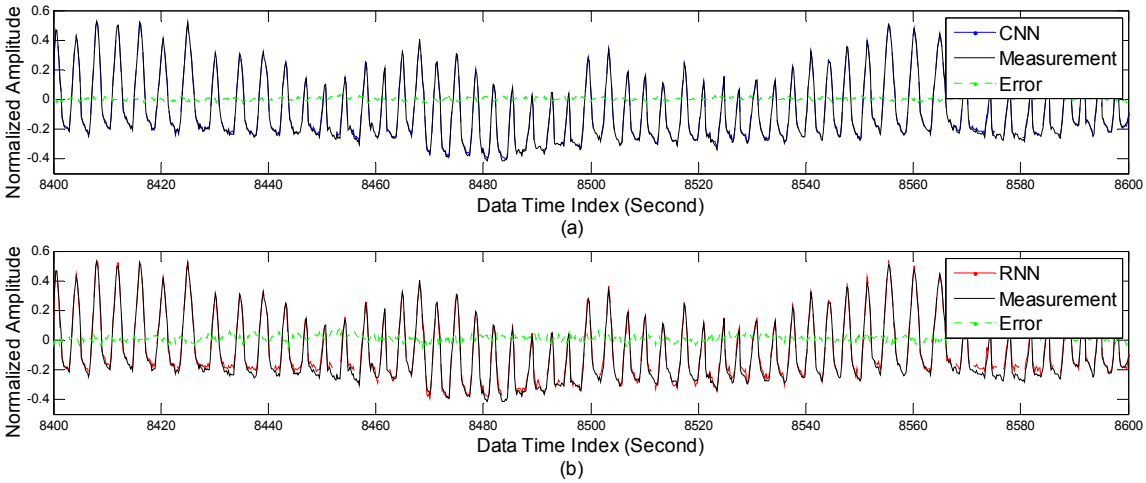

Figure 48. Comparison on estimation performance of DB89 with 192 ms latency.
(a) CNN estimation performance, and (b) RNN estimation performance. We can notice that the target estimation values of the proposed CNN align closer to the measurement values than those of RNN. Here, the standard deviation values of CNN and RNN are 0.010 and 0.021, respectively in this specific data with the 200-second recordings.

www.manaraa.com

Fig. 48 shows the estimation performance of the respiratory motion, *i.e.*, CNN and RNN, including the measurement and error values. The unit of vertical axis in Fig. 48 is normalized for the amplitude of the sensor position corresponding to the real measurement dataset range [-1.7021×10^3, -1.6891×10^3], *i.e.* the maximum value as 1 and the minimum value as −1. As can be seen in Fig. 48(a), the proposed method CNN aligns closer to the measurement values than the other values in Fig. 48(b). The standard deviation values of CNN and RNN for these specific data with the 200-second recordings are 0.010 and 0.021, respectively. That means CNN reduces the prediction error by as much as two times compared to RNN.

5.4.4 Prediction Accuracy with Time Horizontal Window

Prediction Time Horizon represents the time interval window to predict the future sensory signal. For comparison with RNN, we compare the error performance with respect to a variety of prediction time horizons using the normalized root mean squared error, Eq. (76) in Chapter 4.4.5.

www.manaraa.com

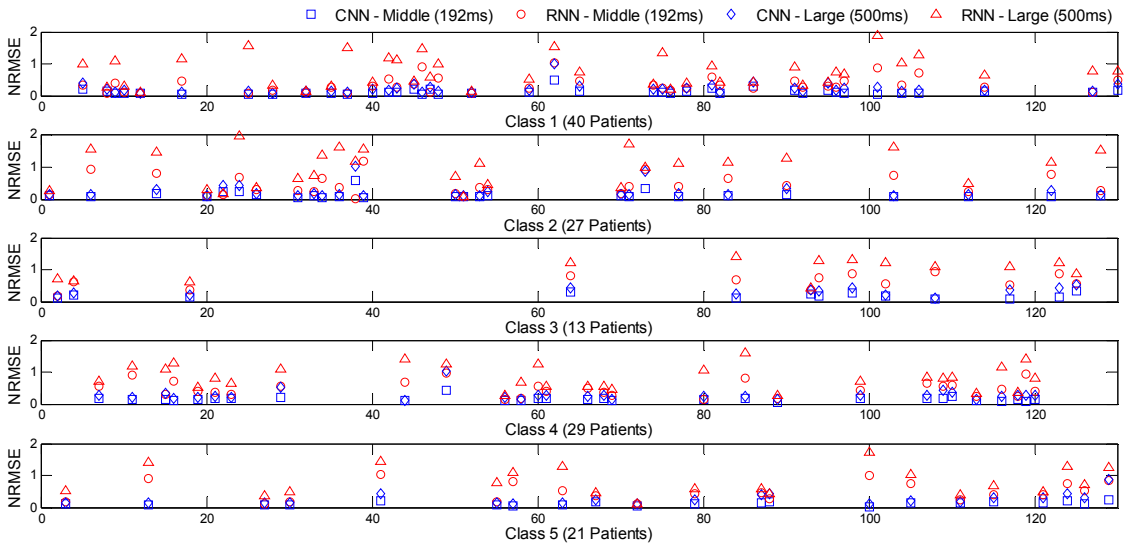

Figure 49. Error Performance with different Prediction Time Horizons (CNN vs RNN).
Here, the average NRMSEs for CNN and RNN are 0.16 and 0.33, respectively.

Fig. 49 shows the NRMSE values of all the classes with respect to middle (192ms), and large (500ms) time prediction. The red symbols for RNN have more errors than the blue symbols for the proposed CNN over all the classes. The NRMSE for CNN was improved in all of the patients except two in class 1 (patient numbers 8 and 86) and three in class 2 (patient numbers 22, 38 and 51). We also show the average error performance for each class in Table 16. In the short time prediction (38ms), all the classes have improved more than 30%. The 50% improvement was achieved in classes 3, 4, and 5 of the large time prediction (500ms). As shown in prediction error of Table 16, the proposed CNN works for any five classes, thus there are no particular differences of error among the five classes because the criterion of feature selections in CNN is designed to minimize the error.

Table 16. Average Error Performance among a variety of Prediction Time (CNN vs RNN)

	Prediction Time Horizon (CNN/RNN)						
	38.46ms	115.38 ms	192.3 ms	269.23 ms	346.15 ms	423.07 ms	500 ms
Class 1	0.088/0.262	0.104/0.299	0.121/0.344	0.137/0.388	0.157/0.455	0.179/0.551	0.222/0.766
Class 2	0.089/0.260	0.109/0.349	0.130/0.430	0.150/0.510	0.171/0.588	0.198/0.708	0.237/0.991
Class 3	0.144/0.491	0.160/0.541	0.177/0.617	0.192/0.675	0.214/0.738	0.255/0.863	0.314/1.012
Class 4	0.125/0.354	0.139/0.387	0.156/0.472	0.173/0.524	0.191/0.610	0.220/0.701	0.274/0.847
Class 5	0.098/0.294	0.110/0.440	0.125/0.495	0.145/0.558	0.175/0.625	0.208/0.708	0.260/0.815

www.manaraa.com

(Unit: NRMSE)

To compare the experimental results with other peer studies, we used experimental results of *i*) optimized adaptive neural network prediction (O-ANN) [254] that is individually optimized to each patient, *ii*) adaptive linear prediction (ALP) as a benchmark method, and *iii*) kernel density estimation-based prediction (KDE) [5] that is a statistical method to estimate the joint probability distribution of the covariate and response variable using kernel density approximation. The NRMSE using *i*) O-ANN was applied to the patient breathing data of the CyberKnife treatment facility at Georgetown University, and *ii*) ALP and *iii*) KDE were applied to patient data acquired with real-time position management, called the RPM system by Varian Medical, Palo Alto, CA. The error performance for these studies can be improved from the standard RNN; the proposed CNN 47.21% (the best improvement), O-ANN 25.27%, ALP 23.79% and KDE 33.83%, respectively.

5.4.5 Prediction Overshoot Analysis

We would like to evaluate the prediction accuracy with evaluation criteria using the marginal value (γ) (67) in Chapter 4.3.4. We add and subtract the marginal value from the measurement values, so that we can get the upper and lower bounds for each patient; for example, Patient DB35 and DB88 shown in Fig. 50.

www.manaraa.com

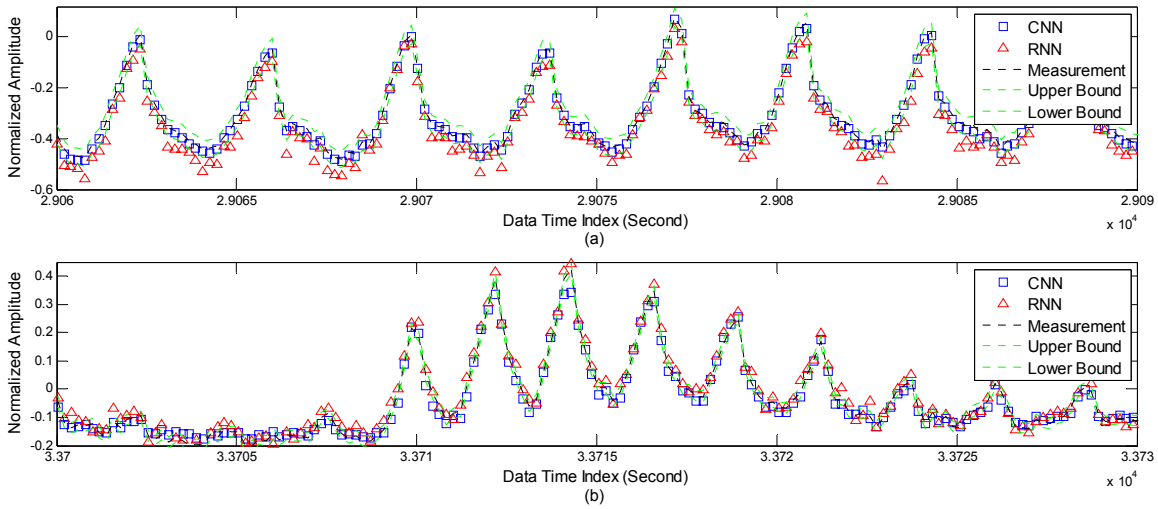

Figure 50. Prediction Overshoot Comparison
(a) Patient DB35 of Class 1 (time index: $2.906\times10^4 \sim 2.909\times10^4$), and Patient DB88 of Class 5 (time index: $3.37\times10^4 \sim 3.373\times10^4$) with the sampling rate of 5 Hz. The RNN presents more prediction overshoots in comparison to CNN. The proposed CNN has no prediction overshoot, whereas the overshoot percentage of RNN is more than 50 % in the regular breathing pattern (a). In the irregular breathing pattern (b), the overshoot percentages of CNN and RNN are 23 % and 46 %, respectively, in this particular time index.

Fig. 50 shows the prediction overshoots of regular motion (DB35 in Class 1) and irregular motion (DB88 in Class 5). In the regular breathing patterns of Fig. 50(a), the proposed CNN has no prediction overshoot, whereas the overshoot percentage of RNN is more than 40 %. In the irregular breathing pattern of Fig. 50(b), the figure shows that most estimation values of the proposed CNN are within the upper and lower bounds, whereas some estimation values of RNN lie outside the confidence level. The time index duration out of the overshoot marginal value (γ) for this particular patient is 23.22 % using CNN and 46.45 % using RNN, respectively.

www.manaraa.com

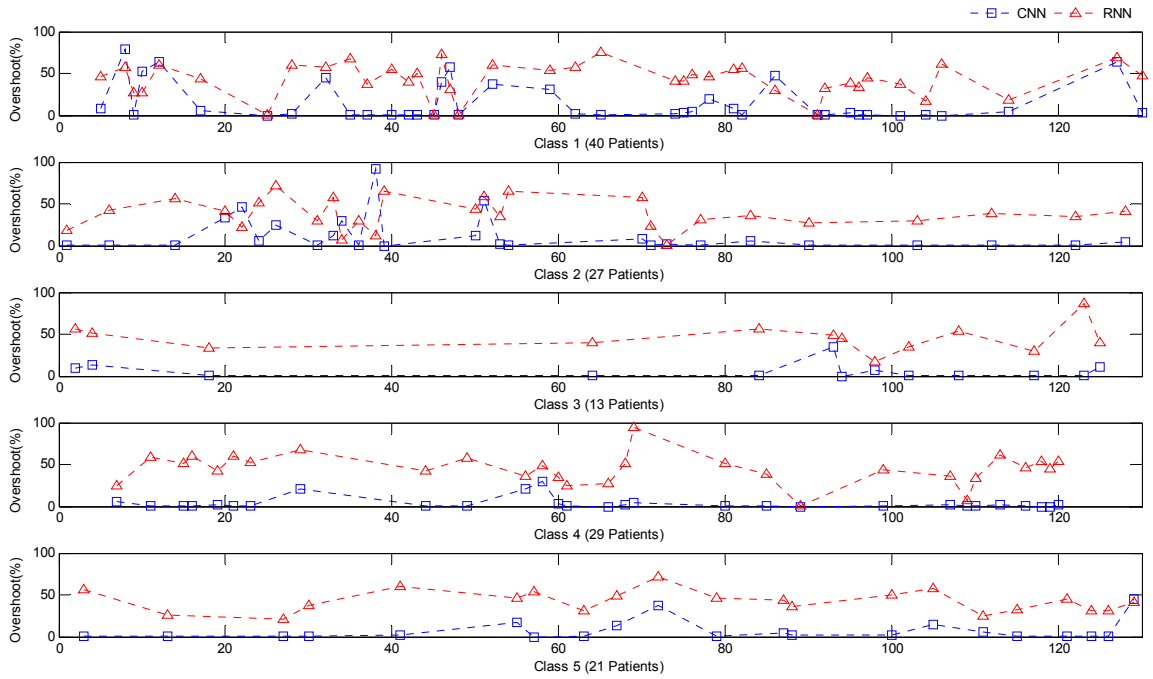

Figure 51. Prediction overshoot comparison over all the patients with 192ms latency.
The prediction overshoots with CNN were improved in the most of the patients except five patients in the class 1 (patient numbers 8, 10, 12, 47 and 86), and three patients in the class 2 (patient numbers 20, 22 and 38).

For the prediction overshoot comparison of all 130 patients, we calculated the number of the total frame and overshoot frames for each patient and show the prediction overshoot frames for the proposed CNN and the alternate RNN with respect to all the classes in Fig. 51. Fig. 51 shows that most of the prediction overshoot numbers for CNN are much smaller than those for RNN over all the patients, even though there are some exceptions, *i.e.* five patients in the class 1 (patient numbers 8, 10, 12, 47 and 86), and three patients in the class 2 (patient numbers 20, 22 and 38). For five classes among the 130 patients, we calculate the averaged overshoot frames over the total frame with respect to the prediction time horizon as shown in Table 17. As shown in prediction overshoot of Table 17, the proposed CNN does not directly address the criterion of overshoot regarding the class selection among multiple patients; therefore the larger size of patients may have relatively large overshoot for in the particular class.

www.manaraa.com

Table 17. Averaged Prediction Overshoot (CNN vs RNN)

	Prediction Overshoot Percentage (CNN/RNN)						
	38.46ms	115.38 ms	192.3 ms	269.23 ms	346.15 ms	423.07 ms	500 ms
Class 1	12.20/34.99	13.98/45.33	15.90/43.75	16.74/42.11	19.88/36.60	13.77/31.81	19.91/41.62
Class 2	10.86/35.32	15.39/36.63	14.61/41.59	17.92/37.65	14.61/29.96	21.88/35.62	17.06/37.31
Class 3	5.20/28.54	4.16/34.15	4.93/35.17	5.20/40.13	10.36/30.37	15.13/32.26	15.14/38.16
Class 4	3.35/38.04	4.11/37.54	4.90/40.76	9.03/40.97	9.56/39.96	11.19/39.21	15.53/37.86
Class 5	6.71/34.45	7.24/34.10	6.72/32.31	7.24/35.41	7.28/35.22	10.34/31.93	10.69/36.93

(Unit: # Overshoot frame/ # Total frame: %)

The averaged overshoot frames of RNN are more than 35% overall the classes, whereas the averaged overshoot frames of the proposed CNN are within 13% in the short time prediction. Note that averaged overshoot frames are less than 7% in the short and middle time prediction of classes 3, 4 and 5. Based on Table 17, the proposed CNN shows more reliable prediction in comparison with the alternate RNN over all the patients.

5.4.6 Comparisons on Computational Complexity

In this Chapter, we would like to evaluate the computational complexity of the proposed method. For the comparisons of the computational complexity, we calculate the CPU time used for prediction process over all the total frames.

Table 18. Comparisons on Computational Complexity

Methods	C-NN	R-NN
CPU Time used (Unit: Millisecond/#Total frame)	15.11	14.80

Table 18 shows the average CPU time used for computational complexity over all the patients. The proposed method needs more computational time for the prediction process because it is working with three independent RMLPs for each marker, whereas RNN

www.manaraa.com

operates with single target datasets. Moreover, the proposed CNN has a coupling matrix to organize three independent processes for each marker. Even though the proposed CNN required more computational time, the prediction accuracy should compensate for the computational complexity. With enough computer power these days, the computer time will probably be reduced to RNN levels within two years. We set the prediction time horizon in this study from 38.46ms to 500ms so that any motion can happen within 15ms on average for the improved prediction.

www.manaraa.com

5.5 SUMMARY

In this Chapter, we proposed a respiratory motion prediction for multiple patient interactions using EKF for RNN. When the breathing patterns for the multiple patients are available, all the patients can be classified into several classes based on breathing features. After this clustering, appropriate parameter selections with respect to each class—*e.g.*, optimal neuron number for the prediction process of the neural network and/or interactive (coupling) degree for the multiple breathing information and so forth—can improve the prediction accuracy in comparison to the previous prediction method, because the multiple respiratory information does not have identical relationships, but relationships that closely resemble one another. That means that when the system for respiratory prediction considers the breathing patterns of multiple patients, it can yield a more accurate prediction performance than when it does not.

For the evaluation criteria of prediction, we showed NRMSE (which is a normalized error value between the predicted and actual signal over all the samples), and prediction overshoot as the reference value to judge how many signals lie outside the confidence level. Our experimental results reveal that the proposed CNN needs more computational time to process due to the abundant breathing information and the additional signal processing and correction process for each RMLP. The proposed CNN, however, can improve NRMSE values by 50% in contrast to the RNN. Moreover, the proposed CNN decreases the number of average prediction overshoot values by 8.37%, whereas the RNN generates prediction overshoot values in more than 40% over all the patients.

www.manaraa.com

CHAPTER 6 IRREGULAR BREATHING CLASSIFICATION FROM MULTIPLE PATIENT DATASETS

Complicated breathing behaviors including uncertain and irregular patterns can affect the accuracy of predicting respiratory motion for precise radiation dose delivery [44] [36] [13] [43] [35] [37]. So far investigations on irregular breathing patterns have been limited to respiratory monitoring of only extreme inspiration and expiration [148]. Using breathing traces acquired on a Cyberknife treatment facility, we retrospectively categorized breathing data into several classes based on the extracted feature metrics derived from breathing data of multiple patients. The novelty of this study is that the classifier using neural networks can provide clinical merit for the statistically quantitative modeling of irregular breathing motion based on a regular ratio representing how many regular/irregular patterns exist within an observation period. We propose a new approach to detect irregular breathing patterns using neural networks, where the reconstruction error can be used to build the distribution model for each breathing class. The sensitivity, specificity and receiver operating characteristic (ROC) curve of the proposed irregular breathing pattern detector was analyzed. The experimental results of 448 patients' breathing patterns validated the proposed irregular breathing classifier.

www.manaraa.com

6.1 INTRODUCTION

Rapid developments in image-guided radiation therapy offer the potential of precise radiation dose delivery to most patients with early or advanced lung tumors [1] [13] [36] [43] [44] [273]. While early stage lung tumors are treated with stereotactic methods, locally advanced lung tumors are treated with highly conformal radiotherapy, such as intensity modulated radiotherapy (IMRT) [273]. Both techniques are usually planned based on four-dimensional computed tomography [1]. Thus, the prediction of individual breathing cycle irregularities is likely to become very demanding since tight safety margins will be used. Safety margins are defined based on the initial planning scan that also analyzes the average extent of breathing motion, but not the individual breathing cycle. In the presence of larger respiratory excursions, treatment can be triggered by respiration motion in such a way that radiation beams are only on when respiration is within predefined amplitude or phase [2]. Since margins are smaller with more conformal therapies, breathing irregularities might become more important unless there is a system in place that can stop the beam in the presence of breathing irregularities. Real-time tumor-tracking, where the prediction of irregularities really becomes relevant [35], has yet to be clinically established.

The motivation and purpose of respiratory motion classification for irregular breathing patterns are that the irregular respiratory motion can impact the dose calculation for patient treatments [3] [149]. A highly irregularly breathing patient may be expected to have a much bigger internal target volume (ITV) than a regular breathing patient, where ITV contains the macroscopic cancer and an internal margin to take into account the variations due to organ motions [149]. Thus, the detection of irregular breathing motion

www.manaraa.com

before and during the external beam radiotherapy is desired for optimizing the safety margin [3]. Only a few clinical studies, however, have shown a deteriorated outcome with increased irregularity of breathing patterns [1] [3] [35], probably due to the lack of technical development in this topic. Other reasons confounding the clinical effect of irregular motion such as variations in target volumes or positioning uncertainties also influence the classification outcomes [3] [35] [149] [269]. The newly proposed statistical classification may provide clinically significant contributions to optimize the safety margin during external beam radiotherapy based on the breathing regularity classification for the individual patient. An expected usage of the irregularity detection is to adapt the margin value, *i.e.*, the patients classified with regular breathing patterns would be treated with tight margins to minimize the target volume. For patients classified with irregular breathing patterns safety margins may need to be adjusted based on the irregularity to cope with baseline shifts or highly fluctuating amplitudes that are not covered by standard safety margins [3] [149].

There exists a wide range of diverse respiration patterns in human subjects [3] [149] [269] [275] [276] [277] [278]. However, the decision boundary to distinguish the irregular patterns from diverse respirations is not clear yet [148] [269]. For example, some studies defined only two (characteristic and uncharacteristic [3]) or three (small, middle, and large [149]) types of irregular breathing motions based on the breathing amplitude to access the target dosimetry [3] [149]. In this study, respiratory patterns can be classified as normal or abnormal patterns based on a regular ratio (γ) representing how many regular/irregular patterns exist within an observation period [148]. The key point of the classification as normal or abnormal breathing patterns is how to extract the

www.manaraa.com

dominant feature from the original breathing datasets [271] [272] [279] [280] [286] [287] [289]. For example, Lu *et al*. calculated a moving average curve using a fast Fourier transform to detect respiration amplitudes [148]. Some studies showed that the flow volume curve with neural networks can be used for the classification of normal and abnormal respiratory patterns [276] [277]. However, spirometry data are not commonly used for abnormal breathing detection during image-guided radiation therapy [276].

To detect irregular breathing, we present a method that retrospectively classifies breathing patterns using multiple patients-breathing data originating from a Cyberknife treatment facility [281]. The multiple patients-breathing data contain various breathing patterns. For the analysis of breathing patterns, we extracted breathing features, *e.g*. vector-oriented feature [271] [272], amplitude of breathing cycle [37] [148] and breathing frequency [269], *etc*., from the original dataset, and then classified the whole breathing data into classes based on the extracted breathing features. To detect irregular breathing, we introduce the reconstruction error using neural networks as the adaptive training value for anomaly patterns in a class.

The contribution of this study is threefold: First, we propose a new approach to detect abnormal breathing patterns with multiple patients-breathing data that better reflect tumor motion in a way needed for radiotherapy than the spirometry. Second, the proposed new method achieves the best irregular classification performance by adopting Expectation-Maximization (EM) based on the Gaussian Mixture model with the usable feature combination from the given feature extraction metrics. Third, we can provide clinical merits with prediction for irregular breathing patterns, such as to validate classification accuracy between regular and irregular breathing patterns from ROC curve analysis, and

www.manaraa.com

to extract a reliable measurement for the degree of irregularity. This study is organized as follows. In Chapter 6.2, the theoretical background for the irregular breathing detection is discussed briefly. In Chapter 6.3, the proposed irregular breathing detection algorithm is described in detail with the feature extraction method. The evaluation criteria of irregular classifier and the experimental results are presented in Chapter 6.4 and 6.5. A summary of the performance of the proposed method and conclusion are presented in Chapter 6.6.

www.manaraa.com

6.2 Related Work

Modeling and prediction of respiratory motion are of great interest in a variety of applications of medicine [7] [15] [17] [100] [274]. Variations of respiratory motions can be represented with statistical means of the motion [17] which can be modeled with finite mixture models for modeling complex probability distribution functions [190]. This study uses expectation-maximization (EM) algorithm for learning the parameters of the mixture model [188] [290]. In addition, neural networks are widely used for breathing prediction and for classifying various applications because of the dynamic temporal behavior with their synaptic weights [35] [36] [282] [283] [288]. Therefore, we use neural networks to detect irregular breathing patterns from feature vectors in given samples.

6.2.1 Expectation-Maximization (EM) based on Gaussian Mixture model

A Gaussian mixture model is a model-based approach that deals with clustering problems in attempting to optimize the fit between the data and the model. The joint probability density of the Gaussian mixture model can be the weighted sum of $m > 1$ components $\phi(x|\ \mu_m, \Sigma_m)$. Here ϕ is a general multivariate Gaussian density function, expressed as follows [188]:

$$\phi(x \mid \mu_m, \Sigma_m) = \frac{\exp\left[-\frac{1}{2}(x-\mu_m)^T \sum_m^{-1}(x-\mu_m)\right]}{(2\pi)^{d/2}\left|\Sigma_m\right|^{1/2}}, \tag{78}$$

where x is the d-dimensional data vector, and μ_m and Σ_m are the mean vector and the covariance matrix of the m^{th} component, respectively. A variety of approaches to the problem of mixture decomposition has been proposed, many of which focus on maximum likelihood methods such as an EM algorithm [290].

www.manaraa.com

An EM algorithm is a method for finding maximum likelihood estimates of parameters in a statistical model. EM alternates between an expectation step, which computes the expectation of the log-likelihood using the current variable estimate, and a maximization step, which computes parameters maximizing the expected log-likelihood collected from E-step. These estimated parameters are used to select the distribution of variable in the next E-step [190].

6.2.2 NEURAL NETWORK (NN)

A neural network is a mathematical model or computational model that is inspired by the functional aspects of biological neural networks [146]. A simple NN consists of an input layer, a hidden layer, and an output layer, interconnected by modifiable weights, represented by links between layers. Our interest is to extend the use of such networks to pattern recognition, where network input vector (x_i) denotes elements of extracted breathing features from the breathing dataset and intermediate results generated by network outputs will be used for classification with discriminant criteria based on clustered degree. Each input vector x_i is given to neurons of the input layer, and the output of each input element makes equal to the corresponding element of the vector. The weighted sum of its inputs is computed by each hidden neuron j to produce its net activation (simply denoted as net_j). Each hidden neuron j gives a nonlinear function output of its net activation $\Phi(\cdot)$, *i.e.*, $\Phi(net_j) = \Phi(\Sigma^N_{i=1} x_i w_{ji} + w_{j0})$ in Eq. (79). The process of output neuron (k) is the same as the hidden neuron. Each output neuron k calculates the weighted sum of its net activation based on hidden neuron outputs $\Phi(net_j)$ as follows [206]:

www.manaraa.com

$$net_k = \sum_{j=1}^{H} w_{kj} \Phi\left(\sum_{i=1}^{N} x_i w_{ji} + w_{j0} \right) + w_{k0} , \tag{79}$$

where N and H denote neuron numbers of the input layer and hidden layer. The subscript i, j and k indicate elements of the input, hidden and output layers, respectively. Here, the subscript 0 represents the bias weight with the unit input vector (x_0=1). We denote the weight vectors w_{ji} as the input-to-hidden layer weights at the hidden neuron j and w_{kj} as the hidden-to-output layer weights at the output neuron k. Each output neuron k calculates the nonlinear function output of its *net* activation $\Phi(net_k)$ to give a unit for the pattern recognition.

www.manaraa.com

6.3 Proposed Algorithms on Irregular Breathing Classifier

As shown in Fig. 52, we first extract the breathing feature vector from the given patient datasets in Chapter 6.3.1. The extracted feature vector can be classified with the respiratory pattern based on EM in Chapter 6.3.2. Here, we assume that each class describes a regular pattern. In Chapter 6.3.3, we will calculate a reconstruction error for each class using neural network. Finally, in Chapter 6.3.4, we show how to detect the irregular breathing pattern based on the reconstruction error.

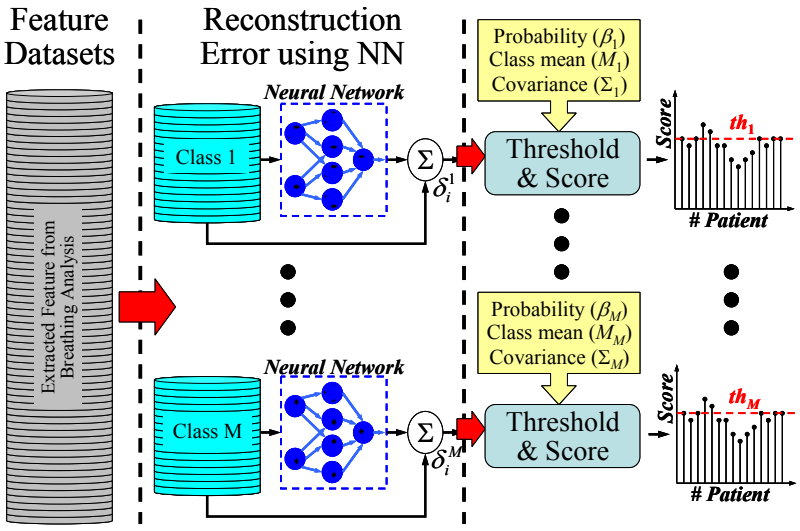

Figure 52. Irregular Breathing Pattern Detection with the proposed algorithm.

6.3.1 Feature Extraction from Breathing Analysis

Feature extraction is a preprocessing step for classification by extracting the most relevant data information from the raw data [271]. In this study, we extract the breathing feature from patient breathing datasets for the classification of breathing patterns. The typical vector-oriented feature extraction including principal component analysis (PCA) and multiple linear regressions (MLR) have been widely used [271] [272]. Murphy *et al.* showed that autocorrelation coefficient and delay time can represent breathing signal

www.manaraa.com

features [35]. Each breathing signal may be sinusoidal variables [37] so that each breathing pattern can have quantitative diversity of acceleration, velocity, and standard deviation based on breathing signal amplitudes [148]. Breathing frequency also represents breathing features [269].

Table 19. Feature Extraction metrics including the formula and notation

Name	Formula
AMV	$\max[R_{xx}],\quad R_{xx}(\tau)=\frac{1}{2T}\int_{-T}^{T}x(t)x(t+\tau)dt$ (T : period of observation)
ADT	$\arg\max_{\tau} R_{xx}(\tau)-\arg\min_{\tau} R_{xx}(\tau)$
ACC	$\mathrm{var}(\Delta x/\Delta t^2)$ (x : observed breathing data)
VEL	$\mathrm{var}(\Delta x/\Delta t)$
BRF	$mean(1/BC_i)$ BC_i : i^{th} breathing cycle range
FTP	$\max X,\quad X(k)=\sum_{n=1}^{N}x(n)e^{-j2\pi(k-1)\left(\frac{n-1}{N}\right)}$ (N : vectors of length N, $1 \le k \le N$)
PCA	Y = *PrinComp*(X) (*PrinComp*(·):PCA function, X: data matrix ($N\times M$, M=3), Y: coefficient matrix ($M\times M$))
MLR	$\left(Z^T Z\right)^{-1} Z^T y$ (Z: predictor, y: observed response)
STD	$s_N=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2}$
MLE	$\hat{\theta}_{mle}=\arg\max_{\theta\in\Theta}\hat{\ell}(\theta\mid x_1,\ldots,x_N),\quad \hat{\ell}=\frac{1}{N}\sum_{i=1}^{N}\ln f(x_i\mid\theta)$ $f(\cdot\mid\theta)$: Normal Distribution

Table 19 shows the feature extraction metrics for the breathing pattern classification. We create Table 19 based on previous entities for breathing features, so that the table can be variable. The feature extraction metrics can be derived from multiple patient datasets with the corresponding formula. To establish feature metrics for breathing pattern classification, we define the candidate feature combination vector ($\bar{x}$) from the combination of feature extraction metrics in Table 19. We defined 10 feature extraction metrics in Table 19. The objective of this Chapter is to find out the estimated feature metrics ($\hat{x}$) from the candidate feature combination vector ($\bar{x}$) using discriminant criterion based on clustered degree. We can define the candidate feature combination vector as $\bar{x}=(x_1,\ldots, x_z)$, where variable z is the element number of feature combination

www.manaraa.com

vector, and each element corresponds to each of the feature extraction metrics depicted in Table 19. For example, the feature combination vector may be defined as $\bar{x}=(x_1, x_2, x_3)$ if the feature combination vector has feature extraction of BRF, PCA, and MLR. The total number (Λ) of feature combination vector using feature extraction metrics can be expressed as follows:

$$\Lambda = \sum_{z=2}^{10} C(10, z), \tag{80}$$

where, the combination function $C(10, z)$ is the number of ways of choosing z objects from ten feature metrics. For the intermediate step, we may select which features to use for breathing pattern classification with the feature combination vectors, *i.e.*, the estimated feature metrics ($\hat{x}$). For the efficient and accurate classification of breathing patterns, selection of relevant features is important [289]. In this study, the discriminant criterion based on clustered degree can be used to select the estimated feature metrics, *i.e.*, objective function $J(\cdot)$ using within-class scatter (S_W) and between-class scatter (S_B) [206] [284]. Here we define the S_W as follows:

$$S_W = \frac{1}{z}\sum_{i=1}^{G} S_i, \quad S_i = \sum_{j=1}^{n_i} (\bar{x}_{ij} - u_i)^2, \quad u_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \bar{x}_{ij}, \tag{81}$$

where z is the element number of a feature combination vector in S_W, G is the total number of class in the given datasets and n_i is the data number of the feature combination vector in the *i-th* class. We define the S_B as follows:

$$S_B = \sum_{i=1}^{G} n_i \times (u_i - u)^2, \quad u = \frac{1}{n}\sum_{i=1}^{n} \bar{x}_i, \tag{82}$$

where n is the total data number of the feature combination vector. The objective function J to select the optimal feature combination vector can be written as follows:

www.manaraa.com

$$J(\hat{x}) = \arg\min_{\bar{x}} \left(\frac{S_W}{S_B} \right), \tag{83}$$

where $\hat{x}$ can be the estimated feature vector for the rest of the modules for breathing patterns classification.

6.3.2 CLUSTERING OF RESPIRATORY PATTERNS BASED ON EM

After extracting the estimated feature vector ($\hat{x}$) for the breathing feature, we can model the joint probability density that consists of the mixture of Gaussians $\phi(\hat{x} \mid \mu_m, \Sigma_m)$ for the breathing feature as follows [188] [190]:

$$p(\hat{x}, \Theta) = \sum_{m=1}^{M} \alpha_m \phi(\hat{x} \mid \mu_m, \Sigma_m), \quad \alpha_m \geq 0, \quad \sum_{m=1}^{M} \alpha_m = 1, \tag{84}$$

where $\hat{x}$ is the d-dimensional feature vector, α_m is the prior probability, μ_m is the mean vector, Σ_m is the covariance matrix of the m^{th} component data, and the parameter $\Theta \equiv \{\alpha_m, \mu_m, \Sigma_m\}_{m=1}^{M}$ is a set of finite mixture model parameter vectors. For the solution of the joint distribution $p(\hat{x}, \Theta)$, we assume that the training feature vector sets $\hat{x}_k$ are independent and identically distributed, and our purpose of this Chapter is to estimate the parameters $\{\alpha_m, \mu_m, \Sigma_m\}$ of the M components that maximize the log-likelihood function as follows [188] [290]:

$$L(M) = \sum_{k=1}^{K} \log p(\hat{x}_k, \Theta), \tag{85}$$

where M and K are the total cluster number and the total number of patient datasets, respectively. Given an initial estimation $\{\alpha_0, \mu_0, \Sigma_0\}$, E-step in the EM algorithm calculates the posterior probability $p(m \mid \hat{x}_k)$ as follows:

www.manaraa.com

$$p(m \mid \hat{x}_k) = \alpha_m^{(t)} \phi(\hat{x}_k \mid \mu_m^{(t)}, \Sigma_m^{(t)}) \Big/ \sum_{m=1}^{M} \alpha_m^{(t)} \phi(\hat{x}_k \mid \mu_m^{(t)}, \Sigma_m^{(t)}), \tag{86}$$

and then M-step is as follows:

$$\begin{aligned}
\alpha_m^{(t+1)} &= \frac{1}{K} \sum_{k=1}^{K} p(m \mid \hat{x}_k) \\
\mu_m^{(t+1)} &= \frac{\sum_{k=1}^{K} p(m \mid \hat{x}_k) \hat{x}_k}{\sum_{k=1}^{K} p(m \mid \hat{x}_k)} = \frac{1}{\alpha_m K} \sum_{k=1}^{K} p(m \mid \hat{x}_k) \hat{x}_k \\
\Sigma_m^{(t+1)} &= \frac{1}{\alpha_m K} \sum_{k=1}^{K} p(m \mid \hat{x}_k) \left[(\hat{x}_k - \mu_k^{(t+1)}) (\hat{x}_k - \mu_k^{(t+1)})^T \right]
\end{aligned} \tag{87}$$

With Eq. (86) in the E-step, we can estimate the t^{th} posterior probability $p(m|\hat{x}_k)$. Based on this estimate result the prior probability (α_m), the mean (μ_m) and the covariance (Σ_m) in the $(t+1)^{th}$ iteration can be calculated using Eq. (87) in the M-step. Based on clustering of respiratory patterns, we can make a class for each breathing feature with the corresponding feature vector ($\hat{x}^m$) of class m. With the classified feature combination vector ($\hat{x}^m$), we can get the reconstruction error for the preliminary step to detect the irregular breathing pattern.

6.3.3 Reconstruction Error for Each Cluster using NN

Using the classification based on EM, we can get M class of respiratory patterns, as shown in Fig. 52. With the classified feature vectors ($\hat{x}^m$), we can reconstruct the corresponding feature vectors (o^m) with the neural networks in Fig. 53 and get the following output value,

$$o^m = \Phi\left(\sum_{j=1}^{H} w_{kj} \Phi\left(\sum_{i=1}^{N} \hat{x}_i^m w_{ji} + w_{j0} \right) + w_{k0} \right), \tag{88}$$

www.manaraa.com

where Φ is the nonlinear activation function, and N and H denote the total neuron number of input and hidden layers, respectively. The neural weights (w) are determined by training samples of multiple patient datasets for each class M. Then, the neural networks calculate the reconstruction error (δ^m) for each feature vector $\hat{x}_i$ using a multilayer perceptron for each class in Fig. 53, as follows [282]:

$$\delta_i^m = \frac{1}{F}\sum_{f=1}^{F}\left(\hat{x}_{if}^m - o_{if}^m\right)^2, \tag{89}$$

where i is the number of patient datasets in a class m, and f is the number of features. After calculating the reconstruction error (δ^m) for each feature vector in Fig. 53, δ^m can be used to detect the irregular breathing pattern in the next Chapter.

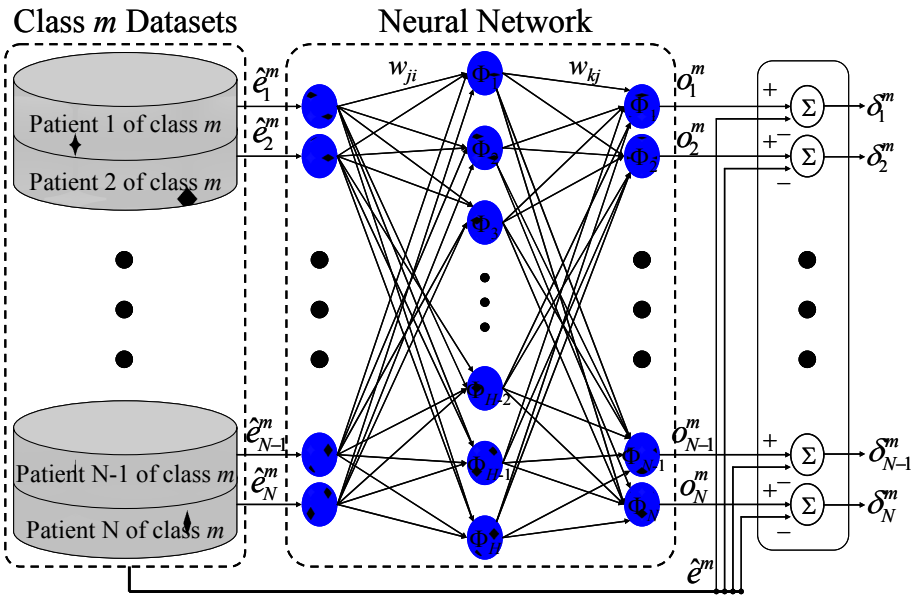

Figure 53. Reconstruction Error to detect the irregular pattern using NN.

6.3.4 Detection of Irregularity Based on Reconstruction Error

For the irregular breathing detection, we introduce the reconstruction error (δ^m), which can be used as the adaptive training value for anomaly pattern in a class m. With the

www.manaraa.com

reconstruction error (δ^m), we can construct the distribution model for each cluster m. That means the patient data with small reconstruction error can have a much higher probability of becoming regular than the patient data with many reconstruction errors in our approach. For class m, the probability (β_m), class means (ν_m) and covariance Σ_m can be determined as follows:

$$\beta_m = \frac{1}{K}\sum_{i=1}^{K} I(m \mid \hat{x}_i), \tag{90}$$

$$\nu_m = \frac{\sum_{i=1}^{K} I(m \mid \hat{x}_i)\delta_i^m}{\sum_{i=1}^{K} I(m \mid \hat{x}_i)} = \frac{1}{\beta_m K}\sum_{i=1}^{K} I(m \mid \hat{x}_i)\delta_i^m, \tag{91}$$

$$\Sigma_m = \frac{1}{\beta_m K}\sum_{i=1}^{K} I(m \mid \hat{x}_i)\left[(\hat{x}_i - M_m)(\hat{x}_i - M_m)^T\right], \tag{92}$$

where $I(m|\hat{x}_i)$=1 if $\hat{x}_i$ is classified into class m; otherwise $I(m|\hat{x}_i)$=0, M_m is the mean value of the classified feature vectors ($\hat{x}^m$) in class m, and K is the total number of the patient datasets. To decide the reference value to detect the irregular breathing pattern, we combine the class means (91) and the covariance (92) with the probability (90) for each class as follows:

$$\bar{\nu} = \frac{1}{M}\sum_{m=1}^{M}\beta_m \nu_m, \quad \bar{\Sigma} = \frac{1}{M}\sum_{m=1}^{M}\beta_m \Sigma_m . \tag{93}$$

With Eq. (93), we can make the threshold value (ξ_m) to detect the irregular breathing pattern in Eq. (94), as follows:

$$\xi_m = \frac{(\nu_m - \bar{\nu})\sqrt{\bar{\Sigma}}}{L_m}, \tag{94}$$

where L_m is the total number of breathing data in class m. For each patient i in class m, we define P_m as a subset of the patient whose score ($\delta^m{}_i$) is within the threshold value (ξ_m) in

www.manaraa.com

class m and $1-P_m$ as a subset of the patient whose score ($\delta^m{}_i$) is greater than the threshold value (ξ_m) in class m, as shown in Fig. 54.

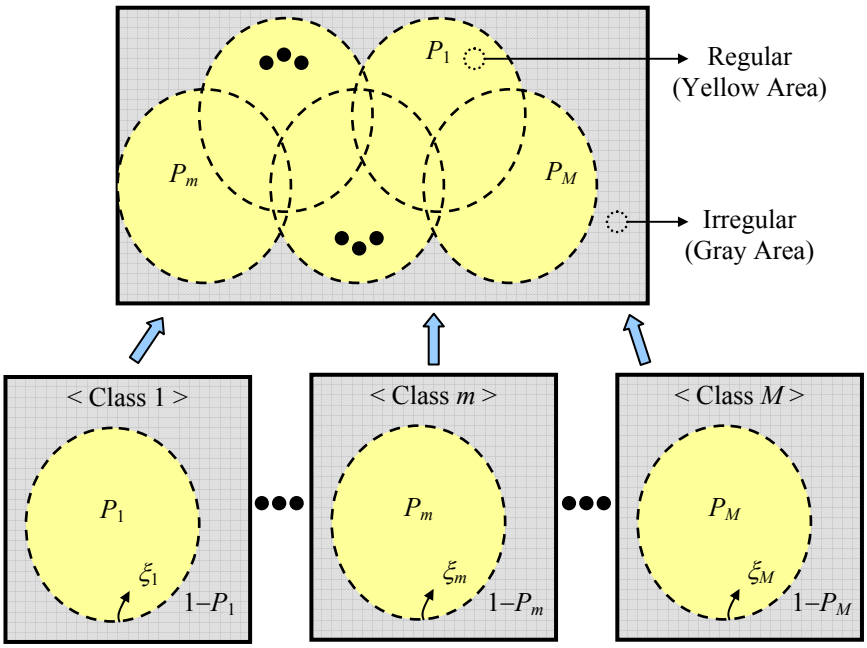

Figure 54. Detection of regular/irregular patterns using the threshold value (ξ_m)

The digit "1" represents the entire patient set for class m in Fig. 54. With Fig. 54 we can detect the irregular breathing patterns in the given class m with the threshold value (ξ_m). Accordingly, all the samples within the threshold value highlighted with yellow in Fig. 54 can be the regular respiratory patterns, whereas the other samples highlighted with gray in Fig. 54 can become the irregular respiratory patterns.

Fig. 54 shows that the threshold value (ξ_m) depicted by dotted lines can divide the regular respiratory patterns (P_m) from the irregular respiratory patterns ($1-P_m$) for each class m. As shown in the upper left corner in Fig. 54, we can summarize the process of the regular/irregular breathing detection, and denote the regular respiratory patterns highlighted with yellow as $\cup^M{}_{m=1}(P_m)=P_1\cup\cdots\cup P_m\cup\cdots\cup P_M$ and the irregular respiratory

www.manaraa.com

patterns highlighted with gray as $\cap^{M}_{m=1}(1-P_m)= (1-P_1)\cap\cdots\cap(1-P_m)\cap\cdots\cap(1-P_M)$. We will use these notations for the predicted regular/irregular patterns in the following Chapter.

www.manaraa.com

6.4 Evaluation Criteria for Irregular Breathing Classifier

6.4.1 Sensitivity and Specificity

We apply standard sensitivity and specificity criteria as statistical measures of the performance of a binary classification test for irregularity detection. The classifier result may be positive, indicating an irregular breathing pattern as the presence of an anomaly. On the other hand, the classifier result may be negative, indicating a regular breathing pattern as the absence of the anomaly. Sensitivity is defined as the probability that the classifier result indicates a respiratory pattern has the anomaly when in fact they do have the anomaly. Specificity is defined as the probability that the classifier result indicates a respiratory pattern does not have the anomaly when in fact they are anomaly-free, as follows [285]:

$$\text{Sensitivity} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

For the sensitivity and specificity, we can use Fig. 54 as the hypothesized class, *i.e.*, the predicted regular or irregular pattern, as follows:

$$FN + TN = \bigcup_{m=1}^{M} P_m, \quad TP + FP = \bigcap_{m=1}^{M} (1 - P_m) \cdot \tag{95}$$

The proposed classifier described in Chapter 6.3 should have high sensitivity and high specificity. Meanwhile, the given patient data show that the breathing data can be mixed up with the regular and irregular breathing patterns in Fig. 55. During the period of observation (T), we notice some irregular breathing pattern. Let us define BC_i as the

www.manaraa.com

breathing cycle range for the patient i as shown in Table 19 and ψ_i as the number of irregular breathing pattern region between a maximum (peak) and a minimum (valley).

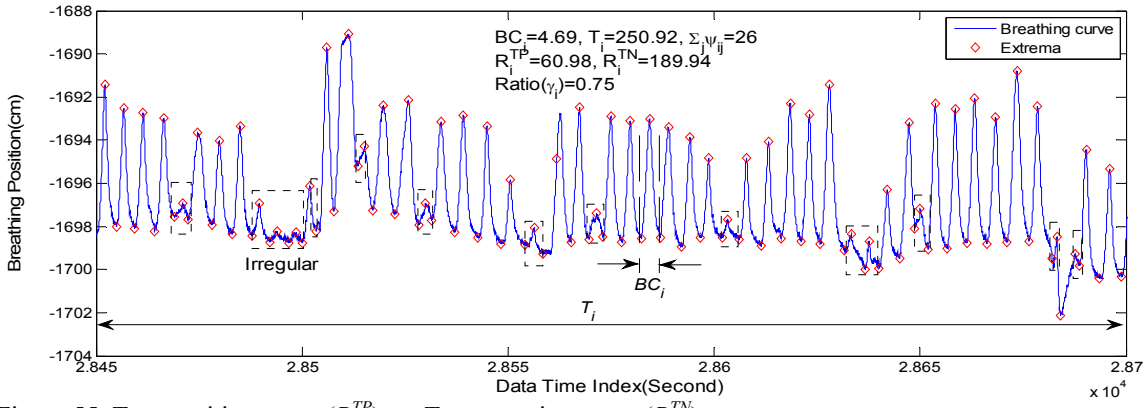

Figure 55. True positive range (R^{TP}) vs. True negative range (R^{TN}).
This figure shows how to decide R^{TP} or R^{TN} of patient i (DB17). In this example, the breathing cycle (BC_i), the period of observation (T_i), and the sum of ψ_i ($\Sigma_j \psi_{ij}$) are given by the numbers of 4.69, 250.92, and 26, respectively. Accordingly, we can calculate the ratio (γ_i) of the true negative range (R_i^{TN}) to the period of observation (T_i), *i.e.*. 0.75. That means 75% of the breathing patterns during the observation period show regular breathing patterns in the given sample.

For the patient i, we define the true positive/negative ranges (R_i^{TP}/R_i^{TN}) and the regular ratio (γ_i) as follows:

$$R_i^{TP} = \frac{BC_i}{2}\sum_j \psi_{ij}, \quad R_i^{TN} = T_i - \frac{BC_i}{2}\cdot\sum_j \psi_{ij}, \quad \gamma_i = \frac{R_i^{TN}}{T_i}, \tag{96}$$

where the ratio (γ_i) is variable from 0 to 1. For the semi-supervised learning of the TP and TN in the given patient datasets, we used the ratio (γ_i) of the true negative range (R_i^{TN}) to the period of observation (T_i) in Eq. (96). Let us denote Ψ_{th} as the regular threshold to decide whether the patient dataset is regular or not. For patient i, we would like to decide TP or TN based on values with the ratio (γ_i) and the regular threshold (Ψ_{th}), *i.e.*, if the ratio (γ_i) of patient i is greater than the regular threshold (Ψ_{th}), the patient is true negative, otherwise ($\gamma_i \leq \Psi_{th}$) true positive. We should notice also that the regular threshold can be

www.manaraa.com

variable from 0 to 1. Accordingly, we will show the performance of sensitivity and specificity with respect to the variable regular threshold in Chapter 6.5.5.

6.4.2 Receiver Operating Characteristics (ROC)

An ROC curve is used to evaluate irregular breathing pattern with true positive rate vs. regular breathing pattern with false positive rate. For the concrete analysis of the given breathing datasets, we would like to show a ROC curve with respect to different regular thresholds. In addition, we will change the discrimination threshold by the period of observation (T_i) to validate the performance of the proposed binary classifier system.

To predict the irregular breathing patterns from the patient datasets, we may evaluate the classification performance by showing the following two ROC analysis:

As the first ROC, we may increase the threshold value ξ_m defined in (94) in Chapter 6.3.4, from 0.1 to 0.99. By changing the observation period T_i of 900, 300, and 100 seconds, the system may include the irregular breathing patterns extracted under the different parameters of ξ_m. Specifically, depending on the observation period T_i, we would like to adjust the threshold value ξ_m for the ROC evaluation of the proposed classifier.

As the second ROC, we may increase the regular threshold (Ψ_{th}) so that the patient datasets with the ratio (γ_i) of patient i may be changed from true negative to true positive. For the analysis based on the regular threshold, we extract the ratio (γ_i) of patient i by changing the observation period T_i of 900, 300, and 100 seconds. The regular threshold Ψ_{th} can be variable from 0.1 to 0.99, especially by changing the regular threshold Ψ_{th} of

www.manaraa.com

0.80, 0.85, and 0.90, defined in Chapter 6.4.1. Depending on the regular threshold (Ψ_{th}), ROC is analyzed for the performance of the proposed classifier.

www.manaraa.com

6.5 Experimental Results

6.5.1 Breathing Motion Data

Three channel breathing datasets with a sampling frequency of 26 Hz are used to evaluate the performance of the proposed irregular breathing classifier. Here each channel makes a record continuously in three dimensions for 448 patient datasets. The breathing recording time for each patient is distributed from 18 minutes to 2.7 hours, with 80 minutes as the average time at the Georgetown University Cyberknife treatment facility. In Fig. 56 we restricted the breathing recording times to discrete values with the unit of five minutes. That means 18 minutes recording time is quantized to 20 minutes for a variable quantity. Fig. 56 shows the frequency distribution of breathing recording time.

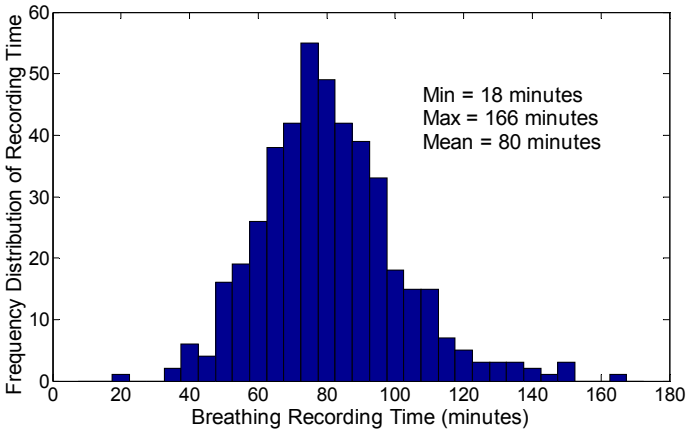

Figure 56. Frequency distribution of recording times for the breathing datasets.
The breathing recordings lasted anywhere from 18 minutes (min) to 166 minutes (max), with 80 minutes as the average time.

The minimum and the maximum recording times are 18 and 166 minutes in Fig. 56. To extract the feature extraction metrics in Table 19, therefore, we randomly selected 18 minute samples from the whole recording time for each breathing dataset because the minimum breathing recording time is 18 minutes. That means we use 28,080 samples to get the feature extraction metrics for each breathing dataset. Meanwhile, every dataset for

www.manaraa.com

each patient is analyzed to predict the irregular breathing patterns. That means we inspect all the datasets to detect the irregular pattern (ψ_i) within the entire recording time. The detected irregular patterns can be used to calculate the true positive/negative ranges (R_i^{TP}/R_i^{TN}) and the ratio (γ_i) for the patients.

6.5.2 Selection of the Estimated Feature Metrics ($\hat{x}$)

The objective of this Chapter is to find out the estimated feature metrics ($\hat{x}$) from the candidate feature combination vector ($\bar{x}$) using discriminant criteria based on clustered degree. Fig. 57(a) shows all the results of the objective function (J) with respect to the feature metrics number. That means each column in Fig. 57 represents the number of feature extraction metrics in Table 19. For example, let us define the number of feature extraction metrics as three (z=3). Here, the feature combination vector can be $\bar{x}$ =(BRF, PCA, STD) with three out of 10 feature metrics, having the number of feature combination vector (C(10, 3)=120). The red spot shows the objective function $J(\cdot)$ for each feature combination vector ($\bar{x}$), whereas the black and the blue spots represent the averaged objective function and the standard deviation of the objective function with respect to the feature metrics number. In Fig. 57(b) we notice that two feature combination vector can have a minimal feature combination vector. Even though z=9 has the minimum standard deviation in Fig. 57(a), a minimum objective function (J) of z=9 is much bigger than those in z=3, 4, 5 and 6 shown in Fig 57(b). The interesting result is that the combinations of BRF, PCA, MLR, and STD have minimum objective functions in z=3 and 4. Therefore, we would like to use these four feature extraction metrics, *i.e.*,

www.manaraa.com

BRF, PCA, MLR, and STD as the estimated feature vector ($\hat{x}$) for the rest of modules for breathing patterns classification.

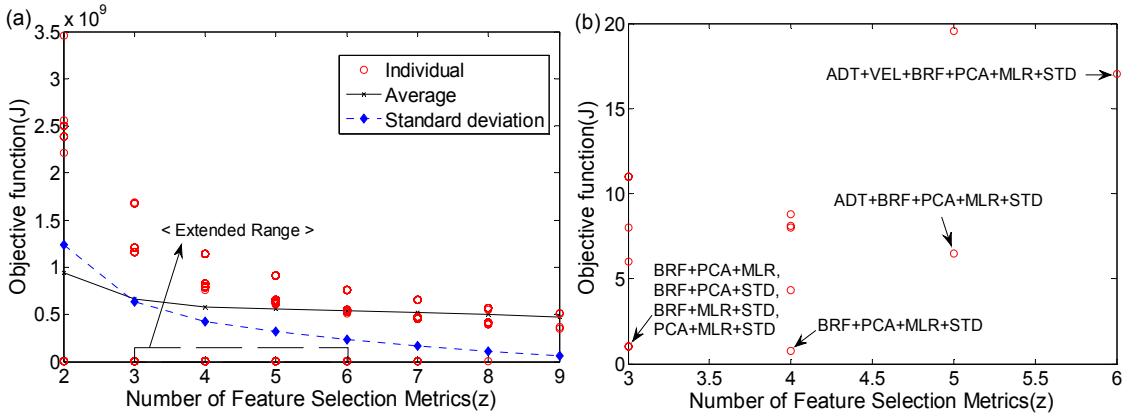

Figure 57. Objective functions for selection of feature metrics.
This figure shows objective functions with respect to the feature metrics number to select the estimated feature metrics ($\hat{x}$); (a) the whole range, and (b) extended range.

6.5.3 Clustering of Respiratory Patterns based on EM

In this Chapter, the breathing patterns will be arranged into groups with the estimated feature vector ($\hat{x}$) for the analysis of breathing patterns. For the quantitative analysis of the cluster models we used two criteria for model selection, *i.e.*, Akaike information criterion (AIC) and Bayesian information criterion (BIC), among a class of parametric models with different cluster numbers [215]. Both criteria measure the relative goodness of fit of a statistical model. In general, the AIC and BIC are defined as follows:

www.manaraa.com

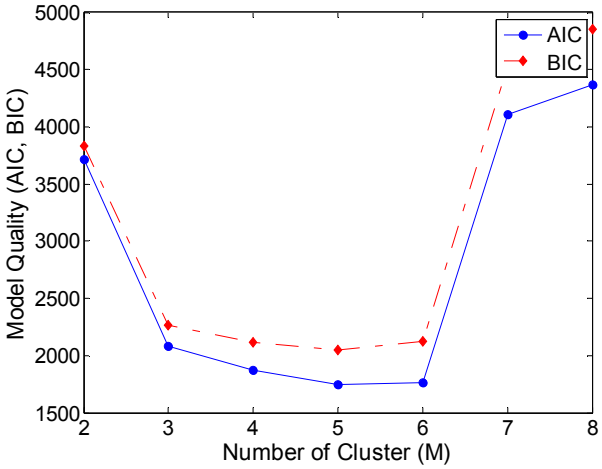

Figure 58. Quantitative model analysis for the selection of cluster number.

$$AIC = 2k - 2\ln(L), \quad BIC = -2 \cdot \ln L + k \ln(n),$$

where n is the number of patient datasets, k is the number of parameters to be estimated, and L is the maximized log-likelihood function for the estimated model that can be derived from Eq. (85).

In Fig. 58, we can notice that both criteria have selected the identical clustering number; M=5. Therefore, we can arrange the whole pattern datasets into five different clusters of breathing patterns based on the simulation results.

6.5.4 Breathing Pattern Analysis to Detect Irregular Pattern

We have shown that breathing patterns are a mixture of regular and irregular patterns for a patient in Fig. 55. Before predicting irregular breathing, we analyze the breathing pattern to extract the ratio (γ_i) with the true positive and true negative ranges for each patient. For the breathing cycle (BC_i) we search the breathing curves to detect the local maxima and minima. After detecting the first extrema, we set up the searching range for the next extrema as 3~3.5 seconds [269]. Accordingly, we can detect the next extrema within half a breathing cycle because one breathing cycle is around 4 seconds [148]. The

www.manaraa.com

BC_i is the mean value of the consecutive maxima or minima. Fig. 59 shows the frequency distribution of BC_i for the breathing datasets. The breathing cycles are distributed with a minimum of 2.9 seconds/cycle and a maximum of 5.94 seconds/cycle. The average breathing cycle of the breathing datasets is 3.91 seconds/cycle.

There are yet no gold standard ways of labeling regular or irregular breathing signals. Lu *et al*. showed, in a clinical way, that moving average value can be used to detect irregular patterns, where inspiration or expiration was considered as irregular if its amplitude was smaller than 20% of the average amplitude [148]. In this study, for the evaluation of the proposed classifier of abnormality, we define all the breathing patterns that are smaller than half the size of the average breathing amplitude as irregular patterns, shown with dotted lines in Fig. 55.

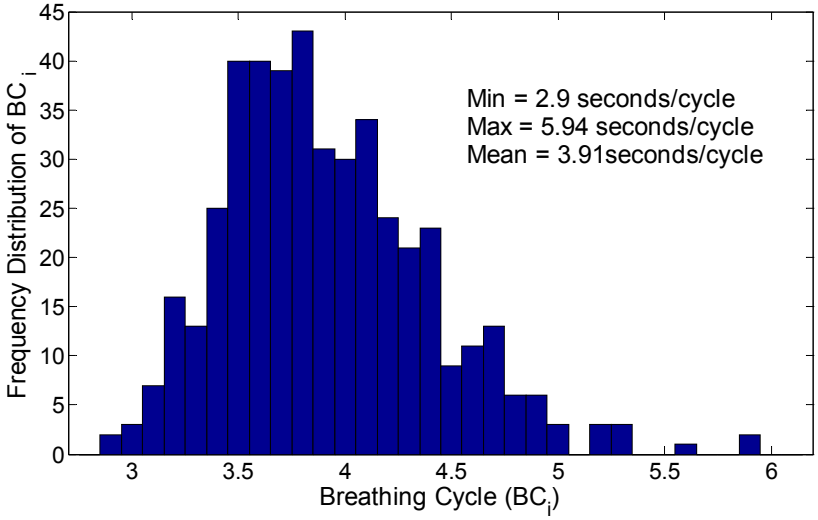

Figure 59. Frequency distribution of breathing cycle (BC_i) for the breathing datasets.
The breathing cycles are variable from 2.9 seconds/cycle to 5.94 seconds/cycle, with 3.91 seconds/cycle as the average time.

www.manaraa.com

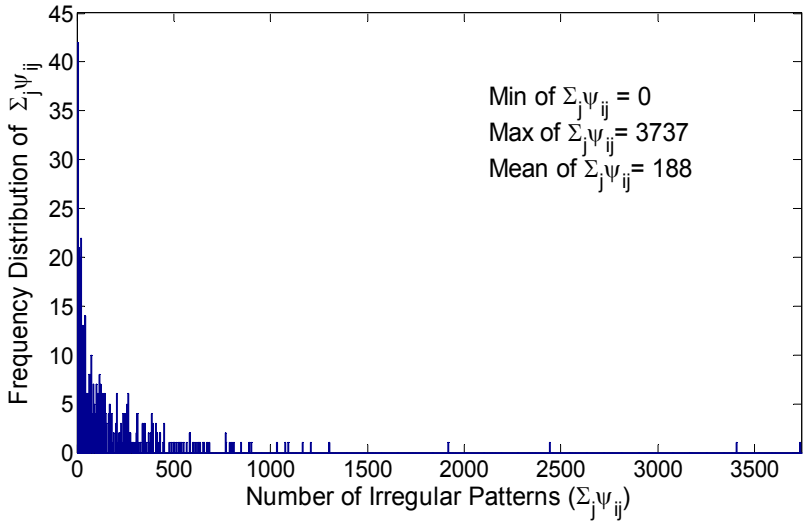

Figure 60. Frequency distribution of the number of irregular patterns ($\Sigma_j\psi_{ij}$).
The numbers of irregular patterns of each breathing dataset are distributed from 0 to 3737 with 188 as the average number.

Fig. 60 shows the frequency distribution of the number of irregular patterns. The numbers of irregular patterns are distributed with a minimum number of 0 and a maximum number of 3737 of irregular patterns. The average number of irregular patterns for the breathing datasets is 188. Accordingly, we can calculate the true positive/negative ranges (R_i^{TP}/R_i^{TN}) and the ratio (γ_i) for the patients after summarizing all the irregular patterns.

Fig. 61 shows the frequency distribution of the ratio (γ_i). Here γ_i is the ratio of the true negative range (R_i^{TN}) to the period of observation (T_i), thus it is dimensionless. The ratio (γ_i) for each breathing dataset is distributed from 0.02 to 1 with 0.92 as the average ratio value. In Fig. 61 we can see that the frequency number of the regular breathing patterns is much higher than that of the irregular breathing patterns in the given datasets. But we can also see that it is not a simple binary classification to decide which breathing patterns are regular or irregular because the frequency distribution of the ratio is analog. We define the vague breathing patterns with the ratio 0.8~0.87 as the gray-level breathing pattern.

www.manaraa.com

We have shown the regular/irregular gray-level breathing patterns among the entire dataset in the following figures.

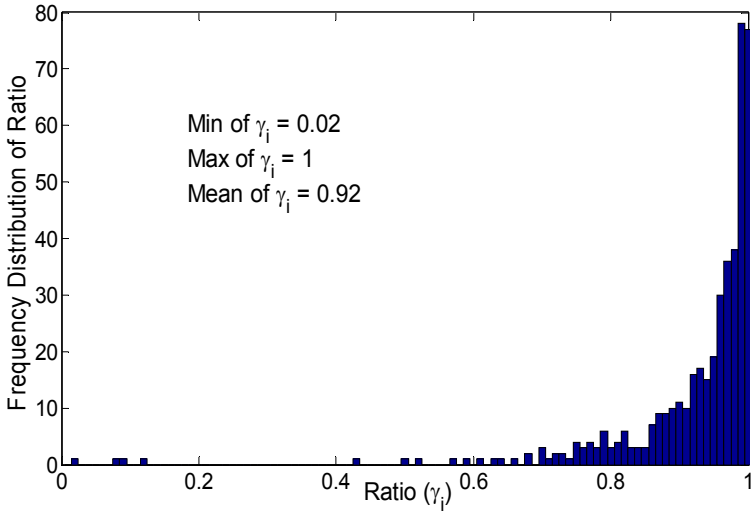

Figure 61. Frequency distribution of ratio (γ_i).
Here γ_i is the ratio of the true negative range (R_i^{TN}) to the period of observation (T_i), thus it is dimensionless. The ratio (γ_i) for each breathing dataset is distributed from 0.02 to 1 with 0.92 as the average ratio value.

Fig. 62 shows regular breathing patterns in the given datasets. There exist several irregular points depicted with green spots. But most of breathing cycles have the regular patterns of breathing curve. Note that the regular breathing patterns have a higher ratio (γ_i) in comparison to the irregular breathing patterns.

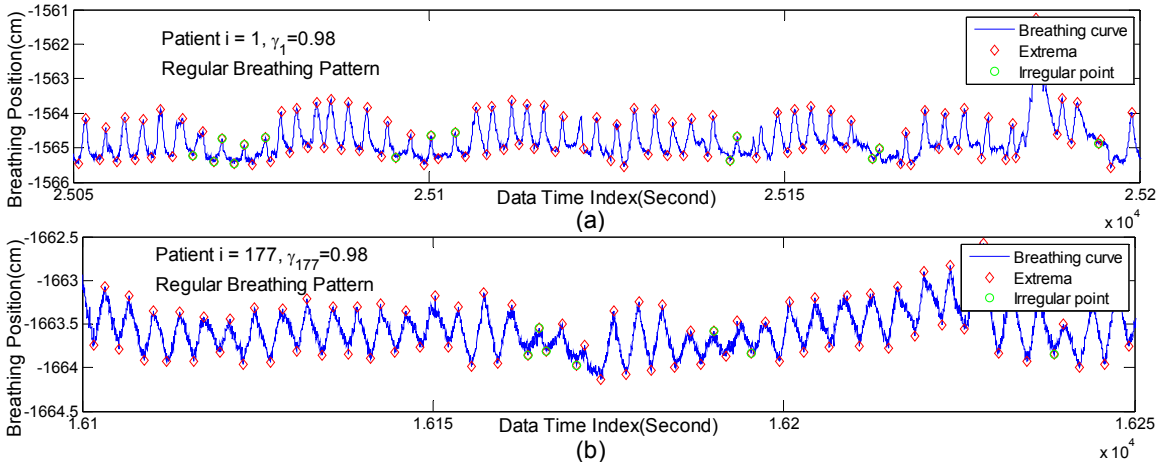

Figure 62. Representing regular breathing patterns.
(a) patient number 1 with the ratio γ_1=0.98; and (b) patient number 177 with the ratio γ_{177}=0.98.

www.manaraa.com

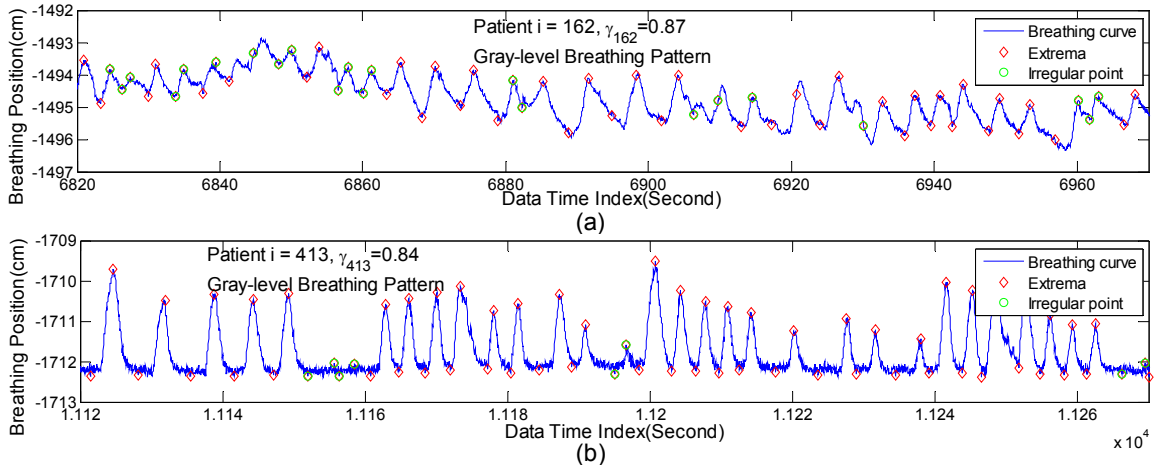

Figure 63. Representing gray-level breathing patterns.
(a) patient number 162 with the ratio γ_{162}=0.87; and (b) patient number 413 with the ratio γ_{413}=0.84.

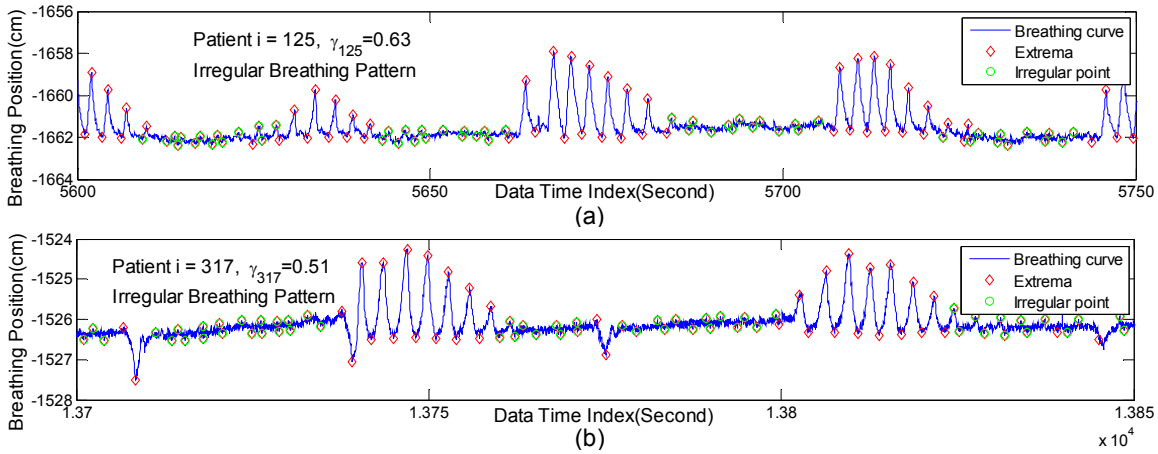

Figure 64. Representing irregular breathing patterns.
(a) patient number 125 with the ratio γ_{125}=0.63; and (b) patient number 317 with the ratio γ_{317}=0.51.

Fig. 63 shows gray-level breathing patterns in the given datasets. Even though the gray-level breathing patterns show some consecutive irregular points, the overall breathing patterns are almost identical as shown in Fig. 63. Fig. 64 shows irregular breathing patterns in the given datasets. Note that the breathing pattern in Fig. 64(b) with a very low ratio (γ_{317}=0.51) is void of regular patterns and that there exists a mass of irregular breathing points in Fig. 64.

www.manaraa.com

6.5.5 CLASSIFIER PERFORMANCE

We evaluate the classification performance whether the breathing patterns are irregular or regular to extract the true positive/negative ranges and the ratio as shown in Fig. 65. To decide the regular/irregular breathing pattern of the patient datasets, we have varied observation periods (T_i) for feature extraction with 900, 300, and 100 seconds.

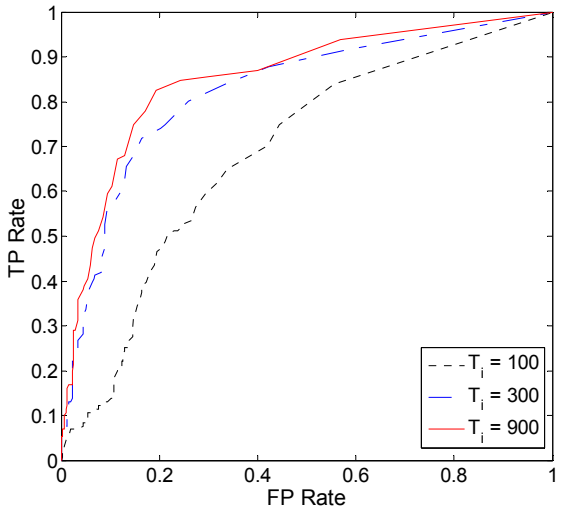

Figure 65. ROC graph of irregular detection with different observation period.

Fig. 65 shows ROC graphs to evaluate how different observation periods affect the classification performance. Here, we fixed the regular threshold Ψ_{th} of 0.92 that is the mean value of the ratio (γ_i), shown in Fig. 61. In Fig. 65 we can see that the proposed classifier shows a better performance with a long observation period (T_i). That means the classifier can be improved by extending the observation period for feature extraction.

www.manaraa.com

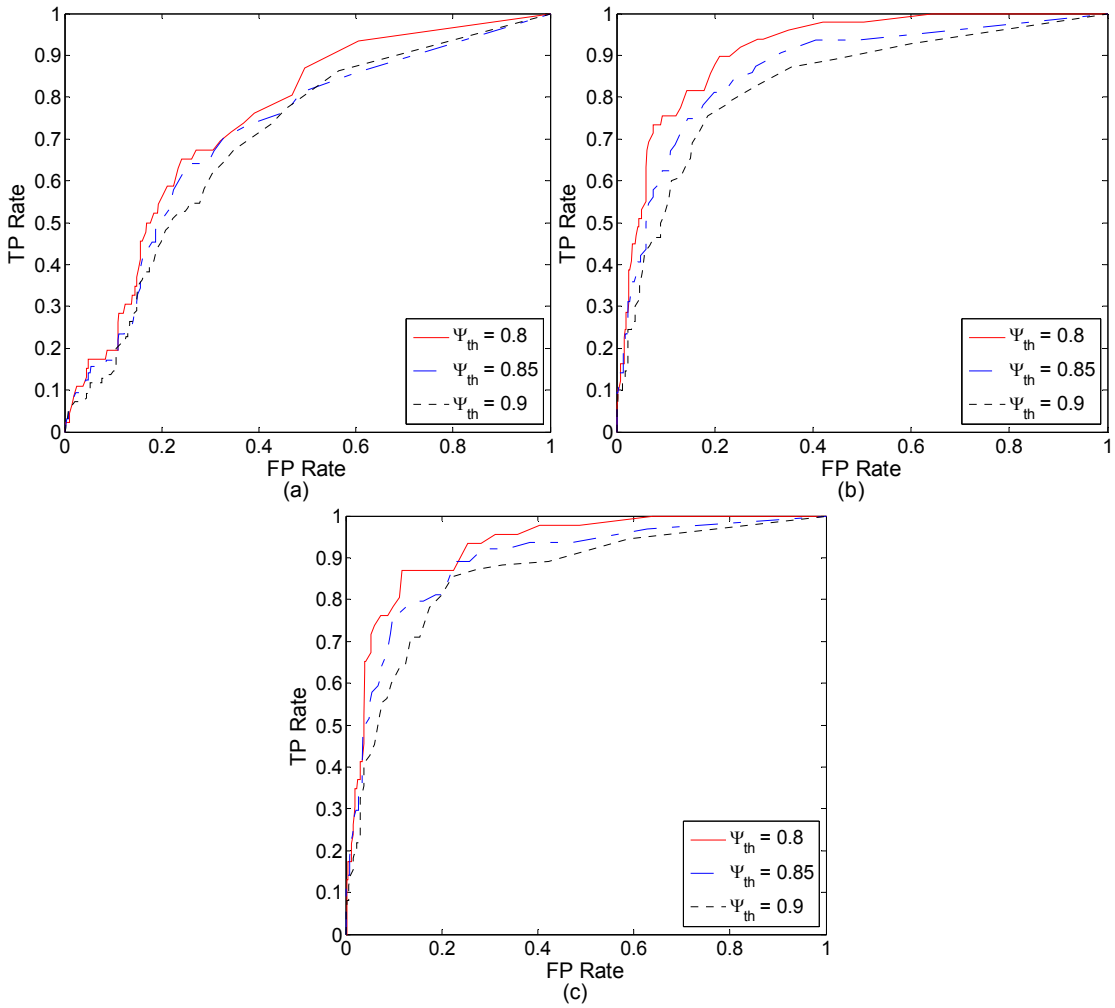

Figure 66. ROC graph of irregular detection with different regular thresholds and observation period (a) observation period T_i = 100 seconds, (b) observation period T_i = 300 seconds, and (c) observation period T_i = 900 seconds.

Fig. 66 shows ROC graphs of irregular detection with different regular thresholds Ψ_{th} of 0.8, 0.85, and 0.9. In this figure, the ratio (γ_i) of patients i are extracted with observation periods T_i of 100, 300, and 900 seconds.

The smaller the regular threshold Ψ_{th}, the better the classifier performance. Here, we notice that the true positive rate (TPR) for the proposed classifier is 97.83% when the False Positive Rate (FPR) is 50% in Fig. 66(c).

Based on the result of ROC graph in Fig. 66 (c), we notice that the breathing cycles of any given patient with a length of at least 900 seconds can be classified reliably enough

www.manaraa.com

to adjust the safety margin prior to therapy in the proposed classification. For the overall analysis of the curve, we have shown the area under the ROC curve (AUC) in Fig. 67. The AUC value can be increased by lowering the regular threshold Ψ_{th}. The maximum AUCs for observation period T_i of 100, 300, and 900 seconds are 0.77, 0.92, and 0.93, respectively. Based on Fig. 67, Fig. 66 (a)-(c) picked 0.8, 0.85, and 0.9 for Ψ_{th}.

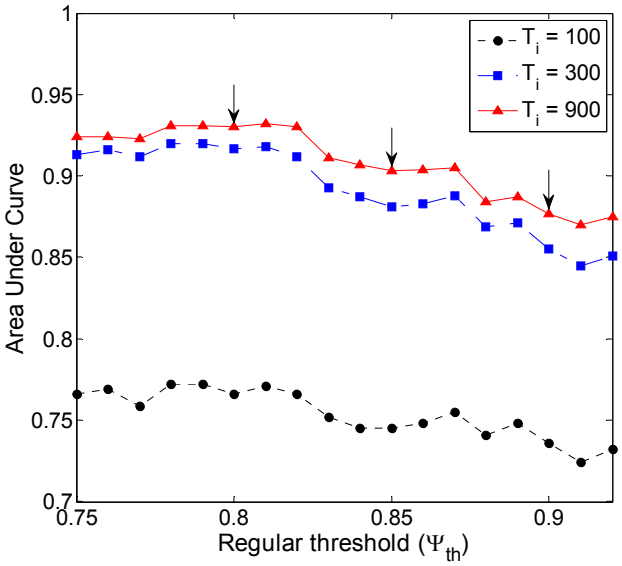

Figure 67. Area under the ROC curve.
The maximum AUCs for the observation period T_i of 100, 300, and 900 seconds are 0.77, 0.92, and 0.93, respectively.

Some studies investigated the classification of regular/irregular breathing patterns for the detection of lung diseases with spirometry [275] [276] [277] [278]. The irregular breathing patterns can also impact on the dosimetric treatment for lung tumors in stereotactic body radiotherapy [3] [43] [149]. However, there are few studies with the results on the classification of breathing irregularity in this area. The following table shows the classification performance of irregular breathing detection using a variety of respiratory measurement datasets.

www.manaraa.com

Table 20. Classifier Studies of Irregular Breathing Detection

Studies	Performance	Measurement Datasets	Methods
Regular/Irregular classification [276]	TPR: 92.6%	Spirometry data of 250	ANN-based
Regular/Irregular classification [277]	TPR: 97.5%	Spirometry data of 205	ANN-based
Regular/Irregular classification [278]	TP/(TP+FP): 98%	74 sleep disordered breathing data	ANN-based
Proposed classification	**TPR: 97.8%**	**Breathing motion data of 448**	**EM/ANN-based**

TPR: True Positive Rate, ANN: Artificial Neural Network

Table 20 shows the classification performances of the irregular detection. We notice that irregular breathing patterns can be detected with the performance of 97.5% TPR using the spirometry data and the ANN-based method [277]. Irregular breathing detection with sleep disordered breathing data [278] shows a better performance of 98% TP/(TP+FP). However, sleep-disorder data can not take the place of the breathing motion for lung cancer treatment [278]. Our proposed classification shows results of the classifier performance of 97.83% TPR with 448 samples breathing motion data. That means the proposed classifier can achieve acceptable results comparable to the classifier studies using the spirometry data.

www.manaraa.com

6.6 SUMMARY

In this Chapter we have presented an irregular breathing classifier that is based on the regular ratio (γ) detected in multiple patients-datasets. Our new method has two main contributions to classify irregular breathing patterns. The first contribution is to propose a new approach to detect abnormal breathing patterns with multiple patients' breathing data that better reflect tumor motion in a way needed for radiotherapy than the spirometry. The second contribution is that the proposed new method achieves the best irregular classification performance by adopting EM based on the Gaussian Mixture model with the usable feature combination from the given feature extraction metrics.

The recorded breathing motions of 448 patients include regular and irregular patterns in our testbed. With the proposed method, the breathing patterns can be divided into regular/irregular breathing patterns based on the regular ratio (γ) of the true negative range to the period of observation. The experimental results validated that our proposed irregular breathing classifier can successfully detect irregular breathing patterns based on the ratio, and that the breathing cycles of any given patient with a minimum length of 900 seconds can be classified reliably enough to adjust the safety margin prior to therapy in the proposed classification.

www.manaraa.com

CHAPTER 7 CONCLUSIONS AND CONTRIBUTIONS

7.1 CONCLUSIONS

The following conclusions can be made from the results obtained from Chapter 4:

7.1.1 HYBRID IMPLEMENTATION OF EXTENDED KALMAN FILTER

- RNN executes in the supervised-training part of the prediction, whereas EKF executes in the part of the correction with predicted or filtered estimation.
- The coupling technique using multiple sensory channel inputs can be used to compensate the computational accuracy.
- Fisher linear discriminant on the discriminant analysis can decide the optimized neuron number for RMLP in the given samples.
- The average percentage of prediction overshoot for HEKF is 3.72%, whereas the average percentage of prediction overshoot for DEKF is 18.61%.
- The proposed HEKF showed the better NRMSE performance across all variable prediction interval times.
- HEKF method needs more time comparing to DEKF because of the calculation of the coupling matrix and the separate neural network for channel number.

The following conclusions can be made from the results obtained from Chapter 5:

7.1.2 CUSTOMIZED PREDICTION OF RESPIRATORY MOTION WITH CLUSTERING

- For the preprocedure of prediction for individual patient, we construct the clustering (five classes) based on breathing patterns of multiple patients using the feature selection metrics that are composed of a variety of breathing features.

www.manaraa.com

- The proposed CNN can outperform RNN with respect to the prediction accuracy with an improvement of 50%.
- CNN works for any of the five classes; thus, there are no particular differences of error among the five classes because the criterion of feature sections in CNN is designed to minimize the error.
- CNN does not directly address the criterion of overshoot regarding the class selection among multiple patients; therefore, the larger size of patients may have relatively large overshoot in the particular class.

The following conclusions can be made from the results obtained from Chapter 6:

7.1.3 Irregular Breathing Classification from Multiple Patient Datasets

- Irregular breathing patterns can be detected using neural networks, where the reconstruction error can be used to build the distribution model for each breathing class.
- The classifier using neural networks can provide clinical merit for the statistically quantitative modeling of irregular breathing motion based on a regular ratio representing how many regular/irregular patterns exist within an observation period.
- The breathing data can be categorized into several classes based on the extracted feature metrics derived from the breathing data of multiple patients.
- The breathing cycles are distributed with a minimum of 2.9 seconds/cycle, a maximum of 5.94 seconds/cycle, and the average breathing cycle of 3.91 seconds/cycle.

www.manaraa.com

- The breathing pattern for each patient can be classified into regular/irregular breathing using the regular ratio, even though the breathing data are mixed up with the regular and irregular breathing patterns in the given samples.
- The true positive rate (TPR) for the proposed classifier is 97.83% when the False Positive Rate (FPR) is 50%.
- The breathing cycles of any given patient with a length of at least 900 seconds can be classified reliably enough to adjust the safety margin prior to therapy in the proposed classification.

www.manaraa.com

7.2 CONTRIBUTIONS

This study has three main contributions on the prediction of respiratory motion in radiation therapy.

The first contribution of this study is to present a new approach to split the whole RMLP with the complicated neuron number into a couple of RMLPs with the simple neuron number to adjust separate input channels. It also comprehensively organizes the multiple channel sensory process by adapting the coupling technique using multiple channel inputs for the mutually exclusive groups to compensate the computational accuracy.

The second contribution is to adopt a clustering method for multiple patients to get more practical breathing pattern information and to find an accurate prediction process for an individual class. With the clustering based on breathing patterns, we can get appropriate parameter selections with respect to each class—*e.g.*, optimal neuron number for the prediction process of the neural network and/or interactive (coupling) degree for the multiple breathing information. It can yield a more accurate prediction performance than when the clustering in not based on breathing patterns.

The third contribution is to propose a new approach to detect abnormal breathing patterns with multiple patient-breathing data. We retrospectively categorized breathing data into several classes based on the extracted feature metrics derived from breathing data of multiple patients. The newly proposed statistical classification may provide clinically significant contributions to optimize the safety margin during external beam radiotherapy based on the breathing regularity classification for the individual patient.

The prediction of respiratory motion traces has become an important research area due to the compensation for uncertainty and irregularity originating from technical limitations or

www.manaraa.com

physiological phenomena. So far, investigations on the prediction of respiratory motion have been limited to estimates of respiratory motion, probably due to immature development of medical systems. This leads to further investigations for adequate and sophisticated radiotherapy technology. Radiation therapy is one of the most advanced treatment techniques for macroscopic cancers. For the accurate and precise delivery of radiation therapy, the prediction of respiratory motion is important. Collaborative research activities with various disciplines including biomedical, engineering, and medical physics are required.

www.manaraa.com

APPENDIX A

A.1 ACRONYMS DEFINITIONS

ANFIS	Adaptive neuro-fuzzy interference system
IMM	Interacting multiple model
CT	Computed tomography
EBRT	External beam radiotherapy
CAT	Computed axial tomography
CBCT	Cone Beam CT
MRI	Magnetic Resonance Imaging
MLC	Multileaf collimator
DMLC	Dynamic MLC
RTRT	Real-time tumor-tracking
RPM	Real-time Position Management
IMRT	Intensity-modulated radiotherapy
RMSE	Root mean squared error
MSE	Mean square error
KF	Kalman filter
CV	Constant velocity
CA	Constant acceleration
EKF	Extended Kalman filter
SHL	Signal history length
FSM	Finite state model
EOE	End-to-exhale
IN	Inhale
EX	exhale
ARMA	Autoregressive moving average
SVM	Support vector machines
HMM	Hidden Markov model
ANN	Artificial neural network
NN	Neural network
LMS	Least mean squares
RLS	Recursive least squares
HEKF	Hybrid Extended Kalman filter
MC-IMME	Multi-channel interacting multiple model estimator

www.manaraa.com

IMME	Interacting multiple model estimator
GMM	Gaussian mixture model
EM	Expectation-maximization
MM	Multiple model
AMM	Autonomous multiple models
CMM	Cooperating multiple models
VSMM	Variable structure multi-models
HEKF	Hybrid implementation based on EKF
NRMSE	Normalized root mean squared error
RNN	Recurrent neural network
BPTT	Back-propagation-through-time
RTRL	Real-time recurrent learning
MLP	Multilayer perceptron
RMLP	Recurrent multilayer perceptron
CNN	Customized prediction with multiple patient interaction using NN
AMV	Autocorrelation MAX value
ADT	Autocorrelation delay time
ACC	Acceleration variance value
VEL	Velocity variance value
BRF	Breathing Frequency
FTP	Max Power of Fourier transform
PCA	Principal Component Analysis Coefficient
MLR	Multiple Linear Regression Coefficient
STD	Standard deviation of time series data
MLE	Maximum Likelihood Estimates
KDE	Kernel Density Estimation
O-ANN	Optimized Adaptive Neural Network
ALP	Adaptive Linear Prediction
ROC	Receiver operating characteristics
AUC	Area under the ROC curve
AIC	Akaike information criterion
BIC	Bayesian information criterion
ITV	Internal target volume
3D-CRT	Three-dimensional conformal radiotherapy
ECG	Electrocardiogram

www.manaraa.com

TPR	True positive rate
TP	True positive
FPR	False positive rate
FP	False positive
TN	True negative
FN	False negative

www.manaraa.com

A.2 SYMBOL DEFINITIONS

$x(k)$	n-dimensional state vector (data vector)
$z(k)$	Measurement vector
$u(k)$	n-dimensional known vector
$v(k)$	Process noise with the property of the zero-mean white Gaussian noise with covariance $Q(k)$
$w(k)$	Measurement noise with the property of the zero-mean white Gaussian noise with covariance $R(k)$
$Q(k)$	Covariance value of process noise $v(k)$
$R(k)$	Covariance value of measurement noise $w(k)$
$F(k)$	State transition model matrix which is applied to the previous state $x(k–1)$
$G(k)$	Control-input model matrix which is applied to the control vector $u(k)$
$H(k)$	Observation model matrix which maps the true state space into the observed space
$\hat{x}(k)$	Predicted state vector
$t(k)$	Recording time at time k
$P(k)$	State prediction covariance vector
$W(k)$	Filter gain value
$S(k)$	Measurement prediction covariance value
μ	Weighting coefficients.
F_{CV}	System matrix for CV filter
Γ_{CV}	Process noise gain matrix for CV filter
F_{CA}	System matrix for CA filter
Γ_{CA}	Process noise gain matrix for CA filter
μ_{ij}	Mixing probability given that the target is in state j that the transition occurred from state i
$\hat{x}^{0j}$	Mixed initial condition matrix
P^{0j}	Mixed initial Kalman filter covariance matrix
Λ_r	Likelihood function corresponding to filter r
μ_j	Mode probability update value for filter r, ($j = 1,\ldots, r$)
$\hat{x}(k\|k)$	Combination of the model-conditioned estimate
$P(k\|k)$	Combination of the model-conditioned estimates and covariance
L	Observations or measurement number that is corresponding to the number of sensor
G	Group number to partition L measurements into G sets
α_y	Prior probability value for the group y ($y \in G$)
m_y	Mean value of group y to be the centroid of the observations in the cluster ($y \in G$)
Σ_y	Covariance value of group y that describes the configurations of clusters ($y \in G$)
$\phi(\cdot)$	General multivariate Gaussian density function

www.manaraa.com

Θ	Set of finite mixture model parameter vectors *i.e.* $\Theta \equiv \{\alpha_y, m_y, \sum_y\}_{y=1}^{G}$
$p(z;\Theta)$	Joint probability density that consists of the mixture of Gaussians
$p(y\|z_j)$	Posterior probability value of group y with z_j
$\delta(G)$	Log-likelihood function with G components
$\Delta(G)$	Difference of the consecutive log-likelihood functions
β_y	Hyper-parameter that presents some background knowledge as a hypothetical prior probability
$\beta_y(k)$	Adaptive hyper-parameter
$\Delta\mu^y$	Difference between the current channel selection probability and the previous one in the group y
δ_{ADT}	Log-likelihood function with the adaptive posterior probability
μ_{ab}	Channel selection probability that represents the conditional transition probability from channel a to channel b.
$T(k)$	Asymptotic Lower Bound of recursive computation based on time k
$T(L)$	Lower bound of iteration execution time for k-means clustering based on L points
u	Input vector with external and feedback inputs
w	Weights
v	Internal activation function of a neuron
Φ	Nonlinear activation function
y_i	Output of the ith neuron
$x(k)$	External input of a system model at time k
$y(k)$	Output of a system model at time k
$(\cdot)^T$	Vector transpose operator
$w(k)$	Weight vector of the entire network at time k
$D(k)$	Desired (teaching) signal at time k
s	Number of weights in the entire network
p	Number of output nodes
$v(k)$	Recurrent activities inside the network at time k
$u(k)$	Input signal applied to the network at time k
$Q(k)$	Process noise with the property of a multivariate zero-mean white noise
$r(k)$	Measurement noise with the property of a multivariate zero-mean white noise
$b(\cdot,\cdot,\cdot)$	Measurement function that accounts for the overall nonlinearity of the multilayer perceptron from the input to the output layer.
$B(k)$	p×s measurement matrix of the linearized model
$\alpha(k)$	p×1 matrix denoting the difference between the desired response d(k) and its estimation
$\hat{w}(k\|k-1)$	s×1 vector denoting the estimate of the weight vector w(k) at time k given the observed data up to time k–1.

www.manaraa.com

$\hat{w}(k\|k)$	(=$\hat{w}(k+1\|k)$) Filtered updated estimate of w(k) on receipt of the observable d(k)
$G(k)$	s×p matrix denoting the Kalman gain that is an integral part of the EKF algorithm
$\Gamma(k)$	p×p matrix denoting the global conversion factor for the entire network
$P(k\|k\text{-}1)$	s×s prediction-error covariance matrix
$P(k\|k)$	s×s filtering-error covariance matrix
G	Designated number of mutually exclusive disjoint weight groups
$\hat{w}_i(k\|k)$	Filtered weight vector for the group i, where i = 1, 2,…, g.
$P_i(k\|k)$	Subset of the filtering-error covariance matrix for the group i, where i = 1, 2,…, g.
$Gi(k)$	Kalman gain matrix for the group i, where i = 1, 2,…, g.
c	Designated number of mutually exclusive channel
$\hat{w}_i^{CP}(k\|k)$	Filtered weight vector
$G_i^{CP}(k)$	Kalman gain matrix for the channel i
$P_i^{CP}(k\|k\text{-}1)$	Prediction-error covariance matrix for the channel i
$P_i^{CP}(k\|k)$	Filtering-error covariance matrix for the channel i
$\Gamma^{CP}(k)$	global conversion factor for the coupled entire network
$d_i(k)$	Desired response for the linearized system
μ_{ij}	Coupling degree to which component (i) depend on one another (j)
Π	Coupling matrix
$\alpha_i^{CP}(k)$	Difference between di(k) and coupled estimations for the channel number i
$\Pi(k)$	adaptive coupling matrix
$H(k)$	Error-gain matrix
$\Delta(k)$	Difference of the consecutive global error gain values
m_i	d-dimensional sample mean for group i
n_i	Component number of group i
S_W	within-class scatter value in the given samples
S_B	between-class scatter value in the given samples
$J(\cdot)$	Objective function to get the optimized group number (g).
θ	Set of neural network true weights and biases
$\hat{\theta}$	Least square estimation of θ
γ	Marginal value to judge prediction overshoot
$\hat{\sigma}$	Standard deviation estimator
I	Candidate feature combination vector
$\hat{I}$	Estimated feature combination vector
c	Class number
$\hat{c}$	Estimated class number to get the minimum of the objective function value (J(c)).

www.manaraa.com

μ_m	Mean vector of the m^{th} component
Σ_m	Covariance matrix of the m^{th} component
N	Neuron number of the input layer
H	Neuron number of the hidden layer
$\bar{x}$	Feature combination vector
$\hat{x}$	Estimated feature metrics
Λ	Total number of feature combination vector
z	Element number of feature extraction metrics
$C(10, z)$	Combination function for the number of selecting z objects from ten feature metrics
G	Total number of class in the given datasets
$p(x, \Theta)$	Joint probability density with $\Theta \equiv \{\alpha_m, \mu_m, \Sigma_m\}_{m=1}^{M}$
α_m	Prior probability
M	Number of finite mixture model (Cluster number)
K	Number of patient datasets
$L(\cdot)$	Objective function to maximize the log-likelihood function
$\hat{e}^m$	Classified feature vectors of class m
o^m	Reconstructed feature vectors with NN
δ^m	Reconstruction error
β_m	Probability of class m
ν_m	Means of class m
Σ_m	Covariance of class m
M_m	Mean value of the classified feature vectors ($\hat{x}^m$) in class m
$I(m\|\hat{x}_i)$	Generalized function depending on $\hat{x}_i$, where $I(m\|\hat{x}_i)=1$ if $\hat{x}_i$ is classified into class m; otherwise $I(m\|\hat{x}_i)=0$
$\bar{V}$	Averaged class mean with the probability for each class
$\bar{\Sigma}$	Averaged covariance with the probability for each class
ξ_m	Threshold value to detect the irregular breathing pattern
L_m	Total number of breathing data in class m
P_m	Subset of the patient whose score is within ξ_m in class m
T_i	Observation period of the patient i
BC_i	Breathing cycle range of the patient i
ψ_i	Number of irregular breathing pattern region of the patient i
R_i^{TP}	True positive range within the observation period (T_i)
R_i^{TN}	True negative range within the observation period (T_i)
γ_i	Ratio of the R_i^{TN} to the T_i for the patient i ($0 \le \gamma_i \le 1$)
Ψ_{th}	Regular threshold to decide whether patient i is regular or not

www.manaraa.com

APPENDIX B

This source codes are implemented in the platform of MATLAB 7.10.0(R2010a).

The estimation source codes for respiratory motion are as follows:

B.1	Neural Network
B.2	Adaptive Neural Network
B.3	Kalman Filter
B.4	Decoupled Extended Kalman Filter

The classification source codes for respiratory motions are as follows:

B.5	Feature Extraction
B.6	Reconstruction Error
B.7	Irregular Detection
B.8	Detection of True Positive and True Negative

B.1 MATLAB CODES FOR NEURAL NETWORK

```
% This is an example of nonlinear autoregressive network with exogenous
% inputs (NARX).
% In this exampel we will use a series-parallel architecture instead of
% feeding back the estimated output.
% This has two advantages.
% First - the input to the feedforward network is more accurate
% Second - the resulting network has a purely feedforward architecture
% magdata - compose of u and y. Each has 1*4001 double data set.
% This file load Cyberknife Data to implement Neural Network.
clear;
fid = fopen('Markers_DB10_Clear.mes');  % Load input data
if fid == -1
    disp('File open not successful');
else
    Cyberknife_Data = textscan(fid,'%f %f %f %f %f %f %f %f %f %f');
    len = length(Cyberknife_Data{1});
end
closeresult = fclose(fid);
if closeresult == 0
    disp('File close successful');
else
    disp('File close not successful');
end

for i = 1:len                          % Store input data in the array
    Time_Stamp(i) = Cyberknife_Data{1}(i);
    x_1(i) = Cyberknife_Data{2}(i);    y_1(i) = Cyberknife_Data{3}(i);
```


www.manaraa.com

```
    z_1(i) = Cyberknife_Data{4}(i);    p_1(i) = -sqrt(x_1(i)^2 +
y_1(i)^2 + z_1(i)^2);
    x_2(i) = Cyberknife_Data{5}(i);    y_2(i) = Cyberknife_Data{6}(i);
    z_2(i) = Cyberknife_Data{7}(i);    p_2(i) = -sqrt(x_2(i)^2 +
y_2(i)^2 + z_2(i)^2);
    x_3(i) = Cyberknife_Data{8}(i);    y_3(i) = Cyberknife_Data{9}(i);
    z_3(i) = Cyberknife_Data{10}(i);   p_3(i) = -sqrt(x_3(i)^2 +
y_3(i)^2 + z_3(i)^2);
end

u = x_1; y = p_1;       % x_1 : Input values, p_1 : target values
% load magdata
[u,us] = mapminmax(u);
[y,ys] = mapminmax(y);
y = con2seq(y); u = con2seq(u);
p = [u(3:end);y(3:end)]; t = y(3:end);

% Create the series-parallel NARX network using the function newnarxsp.
% Use 10 neurons in the hidden layer and use trainbr for the training
function.
d1 = [1:2];
d2 = [1:2];
narx_net = newnarxsp({[-1 1], [-1 1]},d1,d2,[10
1],{'logsig','purelin'});
% logsig : Log-Sigmoid Transfer Function
narx_net.trainFcn = 'trainbr';
narx_net.trainParam.show = 10;
narx_net.trainparam.epochs = 600;

% Now ready to train the network
for k=1:2,
    Pi{1,k}=u{k};
end
for k=1:2,
    Pi{2,k}=y{k};
end
narx_net = train(narx_net,p,t,Pi);

% simulates the network and plots the resulting errors
yp = sim(narx_net,p,Pi);
e = cell2mat(yp) - cell2mat(t);
plot(e);
figure;

% There is a toolbox function (sp2narx) for converting NARX networks
from
% the series-parallel configuration to the parallel configuration
narx_net2 = sp2narx(narx_net);
y1 = y(3:end); u1 = u(3:end);
p1 = u1(3:end); t1 = y1(3:end);
for k=1:2,
    Ai1{1,k}=zeros(10,1);
    Ai1{2,k}=y1{k};
end
for k=1:2,
    Pi1{1,k} = u1{k};
end
```


www.manaraa.com

```
yp1 = sim(narx_net2,p1,Pi1,Ai1);
yp1_1 = cell2mat(yp1); t1_1 = cell2mat(t1);
plot(Time_Stamp(5:end),yp1_1,'b', Time_Stamp(5:end),t1_1,'r')
legend('Neural Network Estimation','Measurement');
xlabel('Data Time Index (ms)');
ylabel('Normalized Position Values');
figure;
%----------------------------< Error Value >--------------------------%
e_2 = yp1_1 - t1_1;
plot(e_2)
title('Error Value');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%--------------------------------< END >-------------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


www.manaraa.com

B.2 MATLAB CODES FOR ADAPTIVE NEURAL NETWORK

```
% This code shows respiratory prediction method with
% Adaptive Neural Network
clear; close all;
fid = fopen('Markers_DB10_Clear.mes'); % Load input data
if fid == -1
    disp('File open not successful');
else
    Cyberknife_Data = textscan(fid,'%f %f %f %f %f %f %f %f %f %f');
    len = length(Cyberknife_Data{1});
end
closeresult = fclose(fid);
if closeresult == 0
    disp('File close successful');
else
    disp('File close not successful');
end
for i = 1:len           % Store input data in the array
    Time_Stamp(i) = Cyberknife_Data{1}(i);
    x_1(i) = Cyberknife_Data{2}(i);     y_1(i) = Cyberknife_Data{3}(i);
    z_1(i) = Cyberknife_Data{4}(i);     p_1(i) = -sqrt(x_1(i)^2 +
y_1(i)^2 + z_1(i)^2);
    x_2(i) = Cyberknife_Data{5}(i);     y_2(i) = Cyberknife_Data{6}(i);
    z_2(i) = Cyberknife_Data{7}(i);     p_2(i) = -sqrt(x_2(i)^2 +
y_2(i)^2 + z_2(i)^2);
    x_3(i) = Cyberknife_Data{8}(i);     y_3(i) = Cyberknife_Data{9}(i);
    z_3(i) = Cyberknife_Data{10}(i);    p_3(i) = -sqrt(x_3(i)^2 +
y_3(i)^2 + z_3(i)^2);
    p_0(i) = (p_1(i)+p_2(i)+p_3(i))/3;
end
I_data_ANN(:,1) = p_1; T_data_ANN(:,1) = p_0; % Read data array
[I_data_ANN,PS] = mapminmax(I_data_ANN(:,:)'); % Normalized the inputs
[T_data_ANN,PS2] = mapminmax(T_data_ANN(:,1)'); % Normalized the target
net_ANN = newlin([-1,1],1);                % generate neural network
net_ANN.inputWeights{1,1}.delays = [0 1 2];% initialize network delays
net_ANN.IW{1,1} = [7 8 9];     % initialize network weight
net_ANN.b{1} = [0];            % initialize network bias
pi ={1 2};          % the initial values of the outputs of the delays
I_data_ANN=num2cell(I_data_ANN); %
T_data_ANN=num2cell(T_data_ANN);
[a,pf] = sim(net_ANN,I_data_ANN,pi); % simulate network with input
net_ANN.adaptParam.passes = 2;
t = cputime;                   % Training the network
[net_ANN,y,E pf,af] = adapt(net_ANN,I_data_ANN,T_data_ANN,pi);
e = cputime - t;
fprintf('CPU time used (ANN): %f\n', e);
y=cell2mat(y);
T_data_ANN=cell2mat(T_data_ANN);
%--------------------------< Normalized RMSE >----------------------%
ANN_nRMSE=sqrt(sum((y-T_data_ANN).^2)/sum((T_data_ANN-
mean(T_data_ANN)).^2));
fprintf('Normalized RMSE(38ms) for ANN is %f\n', ANN_nRMSE);
%-------------------------------------------------------------------%
plot(Time_Stamp(:),y(:),'b',Time_Stamp(:),T_data_ANN(:),'r');
%-------------------------------< END >-----------------------------%
```


www.manaraa.com

B.3 MATLAB CODES FOR KALMAN FILTER

```
% This code shows prediction algorithm with Kalman Filter

close all;
%clear all;
clc;

dt=1/15;

measnoise = 10; % position measurement noise
accelnoise = .2; % acceleration noise

a = [1 dt; 0 1]; % transition matrix
b = [dt^2/2; dt]; % input matrix
c = [1 0]; % measurement matrix
x = [0; 0]; % initial state vector

xhat = x; % initial state estimate
Sz = measnoise^2; % measurement error covariance
Sw = accelnoise^2  * [dt^4/4 dt^3/2; dt^3/2 dt^2]; % process noise cov
P = Sw; % initial estimation covariance

temp = xlsread('..\excel\motionMed.xls', 'A1:C400'); % Load input
t=cputime;
p=length(temp);
xls_row=1;
duration=dt*(p-1); %see .xls file then use one/two less than total no
of row.

% Initialize arrays for later plotting.
pos = zeros(1, p); % true position array
poshat = zeros(1, p); % estimated position array
posmeas = zeros(1, p); % measured position array
vel = zeros(1, p); % true velocity array
velhat = zeros(1, p); % estimated velocity array
times = zeros(1,p); %initialize time variable
image_width = 640;
f_eqi = 5*58;

for i=1:p
        tic;


    if i < 3
        u=0;
    else
        v1=(temp(i-1,1)- temp(i-2,1))/dt ;%here we are using last 2
position for calculating instataneous accleration
        v2=(temp(i,1)- temp(i-1,1))/dt;
        u=(v2-v1)/dt;
    end

    % Simulate the linear system.
```


www.manaraa.com

```
    ProcessNoise = accelnoise * [(dt^2/2)*randn; dt*randn];
    x = a * x + b * u + ProcessNoise;
    % Simulate the noisy measurement
    MeasNoise = measnoise * randn;
    y = c * x + MeasNoise;
    % Extrapolate the most recent state estimate to the present time.
    xhat = a * xhat + b * u;
    % Form the Innovation vector.
    Inn = y - c * xhat;
    % Compute the covariance of the Innovation.
    s = c * P * c' + Sz;
    % Form the Kalman Gain matrix.
    K = a * P * c' * inv(s);
    % Update the state estimate.
    xhat = xhat + K * Inn;
    % Compute the covariance of the estimation error.

    P = a * P * a' - a * P * c' * inv(s) * c * P * a' + Sw;
    pos(i) = x(1);
    posmeas(i) = y;
    poshat(i) = xhat(1);
    vel(i) = x(2);
    velhat(i) = xhat(2);
times(i) = toc;

end

CPUTime=cputime-t;
disp('CPU time used: '); disp(CPUTime);

k=1:p;
k2=1:p-1;
actual = temp(1:p, xls_row)';

KF_angleError = actual - poshat;
%calculate mean error
meanError = mean(KF_angleError);
disp('Average/Mean Error: '); disp(meanError);

%calculate SD error
sdError = sum((KF_angleError-meanError).^2)/p;
disp('Standard Deviation of Error: '); disp(sdError);

%mean time
disp('Average/Mean Time: '); disp(mean(times));
```


www.manaraa.com

B.4 MATLAB CODES FOR DECOUPLED EXTENDED KALMAN FILTER

```
% This code shows respiratory prediction method with Decoupled EKF
% DEKF use 3 RMLPs for the prediction part.
clear; clc; close all;
NUM_EH = 3;  NUM_SS = 50; % number of Epoch and group
Training = 1000; % length of sequence for training
net = rmlp(9,6,6,1);    % Generate network
A_Neuron = net.AllNum; % to get the parameters from rmlp
IN = net.InNum; OUT = net.OutNum; H_1 = net.H1Num;  H_2 = net.H2Num;
WNum = net.WNum;   GpNum = A_Neuron;
len_subset = IN + OUT; % length of subset
W_All = [net.W.val]; % get Weight value
W_Gp = [net.W.dest]; % get All Weight value
for i = (1:GpNum),
W(i).val = W_All(min(find(W_Gp == i)):max(find(W_Gp == i)));
W(i).length = length(find(W_Gp == i));
end;
num_eh = NUM_EH; % number of Epoch
num_subset = NUM_SS;  % number of group
len_seq = Training;
R = annealing(100,5,num_eh); % initialize R value
Q = annealing(1E-2,1E-6,num_eh); % initialize Q value
learning_rate = annealing(1,1E-5,num_eh); % learning_rate
n = 1; m = 1;
timeflag = cputime; % Timer
start_point = ceil((len_seq-num_subset-len_subset+2)*rand(1,num_eh));
%------------------< End of training initialization >-----------------%
%------------< Import input data: input and target values >-----------%
load_CyberknifeData_TEST;     % Load input data
I_data(:,1) = x_1;  I_data(:,2) = y_1;  I_data(:,3) = z_1;
I_data(:,4) = x_2;  I_data(:,5) = y_2;  I_data(:,6) = z_2;
I_data(:,7) = x_3;  I_data(:,8) = y_3;  I_data(:,9) = z_3;
T_data(:,1) = p_0;
[I_data,PS] = mapminmax(I_data(:,:)'); % Normalized the input value
[T_data,PS2] = mapminmax(T_data(:,1)'); % Normalized the target value
I_data = I_data';   T_data = T_data';
[inpSize, inpNum] = size(I_data');[tarSize, tarNum] = size(T_data');
t = cputime;
%------------< Main loop - Decoupled Extended Kalman Filter >---------%
for k = (1:num_eh),
X1_0 = zeros(1,H_1);
for i = (1:GpNum),     % GpNum : A_Neuron = net.AllNum
K(i).val = 0.01^(-1)*eye(W(i).length);
end;
W0 = zeros(H_1,H_1+IN);
%------------------------< Initialization >--------------------------%
[X1_1 X2 out(1)] = rmlp_run(net,I_data(1,:),X1_0); % first running
[X1_2 X2 out(2)] = rmlp_run(net,I_data(2,:),X1_1); % second running
for j = (3:inpNum),
temp1 = 0; % temporary
AA = []; % temporary variable
W1 = []; % input --> first layer
W2 = []; % first --> second layer
W3 = []; % second --> output layer
for i = (1:H_1),
```


www.manaraa.com

```
W1 = [W1; W(i).val];
end;
for i = (H_1+1:H_1+H_2),
W2 = [W2; W(i).val];    % Weight matrix : first --> second layer
end;
for i = (H_1+H_2+1:A_Neuron), % Weight matrix :second --> output layer
W3 = [W3; W(i).val];
end;
[X1_3 X2 out(j)] = rmlp_run(net,I_data(j,:),X1_2); % Network running
for i = (H_1+H_2+1 : A_Neuron),
C(i).val = X2;
end;
D1 = (W3*diag(d_hyperb(W2*X1_3')))'*X1_3; % Network weight update
for i = (H_1+1 : H_1+H_2),
C(i).val = D1(i-H_1,:);
end;
D2 = (W3*diag(d_hyperb(W2*X1_3'))*...           % Network weight update
W2*diag(d_hyperb(W1*[X1_2 I_data(j,:)]')))'*...
[X1_2 I_data(j,:)];
D2 = D2 + (W3*diag(d_hyperb(W2*X1_3')) * ...
W2*diag(d_hyperb(W1*[X1_2 I_data(j,:)]'))* ...
W1(:,1:H_1)*diag(d_hyperb(W0*...
[X1_1 I_data(j-1,:)])))'*[X1_1 I_data(j-1,:)];
for i = (1 : H_1),
C(i).val = D2(i,:);
end;
%----------< Decoupled Extended Kalman Filter >---------%
alpha = T_data(j) - out(j); % Innovation process
for m = (1:GpNum),
temp1 = C(m).val*K(m).val*C(m).val' + temp1;
end;
Gamma = inv(temp1+R(1));
for i = (1:GpNum), % number of Group
G(i).val = K(i).val*C(i).val'*Gamma;

if abs(alpha) > 5E-2,
W(i).val = W(i).val + learning_rate(1)*(G(i).val*alpha)';
end;
K(i).val = K(i).val - G(i).val*C(i).val*K(i).val + Q(1);
end;
%-------------< Update the RMLP net Weight >-----------------%
for i = (1:GpNum),
AA = [AA, W(i).val];
end;
for i = (1:WNum),
net.W(i).val = AA(i); % update Weight of RMLP
end;
%---------------------< End of RMLP net Weight >-----------------%
X1_1 = X1_2;    % states replacement
X1_2 = X1_3;
W0 = W1;        % First layer Weight replacement
end;
end;
%----------------------------< End of DEKF >----------------------%
```


www.manaraa.com

B.5 MATLAB CODES FOR FEATURE EXTRACTION

```
% This code extracts feature from breathing datasets.
% The extracted feature metrics composed of 10 components as follow:
% 1: Autocorrelation MAX
% 2: Autocorrelation Delay time
% 3: Acceleration variance value
% 4: velocity variance value
% 5: Breath Frequency
% 6: Max Power of Fourier transform
% 7: Priciple Component Analysis
% 8: Multiple Linear Regression
% 9: Standard deviation
% 10: Maximum Likelihood Estimates
% Autocorrelation MAX and Delay time
input_data = (data_1(:,4)+data_2(:,4)+data_3(:,4))/3;
figure;
subplot(3,1,1);
plot(Time_Stamp(:),input_data(:));
xlabel('Time index (sec)');
ylabel('Amplitude');
subplot(3,1,2);
MAXLAG = 50000;
[Rxx,Lag]=xcorr(input_data,input_data,MAXLAG);
for i=1:MAXLAG
    if Rxx(i)==0 && Rxx(i+1)>0
        Min_lag = i+1;
    end
end
Decay_Time = (MAXLAG - Min_lag);
plot(Rxx);
xlabel('Lags');
ylabel('Autocorrelation Function');
%fprintf('Autocorrelation MAX value =    %f\n',max(Rxx));
%fprintf('Autocorrelation delay time=    %f\n',max(Decay_Time));
%% Acceleration and velocity variance value
Acceleration = zeros(len,1);
Velocity = zeros(len,1);
for j=2:len
    Acceleration(j) = (input_data(j) - input_data(j-1))/(Time_Stamp(j)-
Time_Stamp(j-1))^2;
    Velocity(j) = (input_data(j) - input_data(j-1))/(Time_Stamp(j)-
Time_Stamp(j-1));
end
%fprintf('Acceleration variance value =  %f\n',var(Acceleration));
%fprintf('Velocity variance value =      %f\n',var(Velocity));
%% Breath Frequency
Time_Fs = zeros(len,1);
Index_Fs = 0;
for j=2:len-1
    if Acceleration(j) < 0 && Acceleration(j+1) > 0
        Time_Fs(j)= Time_Stamp(j);       % Time stamp
        Index_Fs = Index_Fs + 1;         % Time stamp index
    end
end
Time_For_Frequency = zeros(Index_Fs,1);
```


www.manaraa.com

```
k=0;
for j=1:len
    if Time_Fs(j) ~= 0
        k = k+1;
        Time_For_Frequency(k)= Time_Fs(j);
    end
end
Freq = zeros(Index_Fs-1,1);
for j=1:Index_Fs-1
    Freq(j)=1/(Time_For_Frequency(j+1)-Time_For_Frequency(j));
end
%fprintf('Breath Frequency =            %f\n',mean(Freq));
%% Max Power of Fourier transform
% Reference: http://www.mathworks.com/help/techdoc/math/brentm1-1.html
fs = 26;
m = length(input_data);      % Window length
n = pow2(nextpow2(m));  % Transform length
Result_DFT = fft(input_data,n);  % DFT     y --> Result_DFT
f = (0:n-1)*(fs/n);     % Frequency range
power = Result_DFT.*conj(Result_DFT)/n;   % Power of the DFT
Result_DFT_1 = fftshift(Result_DFT);        % Rearrange y values
f0 = (-n/2:n/2-1)*(fs/n);                   % 0-centered frequency range
power0 = Result_DFT_1.*conj(Result_DFT_1)/n;% 0-centered power
subplot(3,1,3)
plot(f0,power0);
xlabel('Frequency (Hz)');
ylabel('Power');
title('{\bf 0-Centered Periodogram}');
%fprintf('Max Power of Fourier Tranaform=    %f\n',max(power0));
%% Principal Component Coefficients
p_Total = [data_1(:,4) data_2(:,4) data_3(:,4)];
Coeff = princomp(p_Total);
%fprintf('PCA Coefficient =          %f\n',...
%    sqrt(Coeff(1,1)^2+Coeff(2,2)^2+Coeff(3,3)));
%% Multiple Linear Regression Coefficient
Total_Data = [data_1(:,1:3) data_2(:,1:3) data_3(:,1:3)];
Reg = regress(input_data,Total_Data);
%fprintf('Multiple Linear Regression =       %f\n',sum(Reg));
%% Standard deviation of time series data
%fprintf('Standard deviation of data =       %f\n',std(input_data));
%% Maximum Likelihood Estimates
%PN = mapminmax(input_data);
[phat,pci]=mle(input_data,'distribution','normal','alpha',.05);
%fprintf('Maximum Likelihood Estimates =     %f\n\n',phat(1,1));
%% Summary
BFM(INDEX,1) = max(Rxx); %AMV
BFM(INDEX,2) = max(Decay_Time);%ADT
BFM(INDEX,3) = var(Acceleration);%ACC
BFM(INDEX,4) = var(Velocity);%VEL
BFM(INDEX,5) = mean(Freq);%BRF
BFM(INDEX,6) = max(power0);%FTP
BFM(INDEX,7) = sqrt(Coeff(1,1)^2+Coeff(2,2)^2+Coeff(3,3));%PCA
BFM(INDEX,8) = sum(Reg);%MLR
BFM(INDEX,9) = std(input_data);%STD
BFM(INDEX,10) = phat(1,1);%MLE
INDEX = INDEX+1;
```


www.manaraa.com

B.6 MATLAB CODES FOR RECONSTRUCTION ERROR

```
% This code produce Reconstruction Error for irregular detection
% The reconstruction error can be used to decide whether the
% breathing pattern is regular or irregular in the next step.
% Before executing the file, need cluster index (c) derived from
% 'Clustering basedonEM' file.
% Global CN (Cluster Number) can be decided 'Clustering basedonEM' file.
global M;           % The element number of cluster [c1,c2,...]
global CN;          % Cluster Number
%% Set up inputs for the neural network and get Delta value from Neural
networks
n=zeros(CN,1);d=zeros(CN,1);
for i=1:CN
    [INPUT]=ReadClusterIndexbased(i,M(i));
    switch i
        case 1          % The number of feature extraction metrics: 1
            INPUT_1=INPUT;
            [n(i),d(i)]=size(INPUT_1);
            Delta_1=NeuralNetworkReconstruct(INPUT_1,M(i));
        case 2          % The number of feature extraction metrics: 2
            INPUT_2=INPUT;
            [n(i),d(i)]=size(INPUT_2);
            Delta_2=NeuralNetworkReconstruct(INPUT_2,M(i));
        case 3          % The number of feature extraction metrics: 3
            INPUT_3=INPUT;
            [n(i),d(i)]=size(INPUT_3);
            Delta_3=NeuralNetworkReconstruct(INPUT_3,M(i));
        case 4          % The number of feature extraction metrics: 4
            INPUT_4=INPUT;
            [n(i),d(i)]=size(INPUT_4);
            Delta_4=NeuralNetworkReconstruct(INPUT_4,M(i));
        case 5          % The number of feature extraction metrics: 5
            INPUT_5=INPUT;
            [n(i),d(i)]=size(INPUT_5);
            Delta_5=NeuralNetworkReconstruct(INPUT_5,M(i));
        case 6          % The number of feature extraction metrics: 6
            INPUT_6=INPUT;
            [n(i),d(i)]=size(INPUT_6);
            Delta_6=NeuralNetworkReconstruct(INPUT_6,M(i));
        case 7          % The number of feature extraction metrics: 7
            INPUT_7=INPUT;
            [n(i),d(i)]=size(INPUT_7);
            Delta_7=NeuralNetworkReconstruct(INPUT_7,M(i));
        case 8          % The number of feature extraction metrics: 8
            INPUT_8=INPUT;
            [n(i),d(i)]=size(INPUT_8);
            Delta_8=NeuralNetworkReconstruct(INPUT_8,M(i));
        case 9          % The number of feature extraction metrics: 9
            INPUT_9=INPUT;
            [n(i),d(i)]=size(INPUT_9);
            Delta_9=NeuralNetworkReconstruct(INPUT_9,M(i));
    end
end
%% Neural Network
Beta = n./sum(n);       % Probability
```


www.manaraa.com

```
Vmean=zeros(CN,1);          % Mean
var_1=zeros(c1,1);var_2=zeros(c2,1);
var_3=zeros(c3,1);var_4=zeros(c4,1);var_5=zeros(c5,1);
Covar_m=zeros(CN,1);       % Covariance
for i=1:CN
    switch i
        case 1    % The number of class : 1
            Vmean(i)=sum(Delta_1)/(Beta(i)*sum(n));
            Temp=mean(INPUT_1);
            for j=1:c1
                var_1(j,1)=(INPUT_1(j,:)-Temp)*(INPUT_1(j,:)-Temp)';
            end
            Covar_m(i)=sum(var_1)/(Beta(i)*sum(n));
        case 2    % The number of class : 2
            Vmean(i)=sum(Delta_2)/(Beta(i)*sum(n));
            Temp=mean(INPUT_2);
            for j=1:c2
                var_2(j,1)=(INPUT_2(j,:)-Temp)*(INPUT_2(j,:)-Temp)';
            end
            Covar_m(i)=sum(var_2)/(Beta(i)*sum(n));
        case 3    % The number of class : 3
            Vmean(i)=sum(Delta_3)/(Beta(i)*sum(n));
            Temp=mean(INPUT_3);
            for j=1:c3
                var_3(j,1)=(INPUT_3(j,:)-Temp)*(INPUT_3(j,:)-Temp)';
            end
            Covar_m(i)=sum(var_3)/(Beta(i)*sum(n));
        case 4    % The number of class : 4
            Vmean(i)=sum(Delta_4)/(Beta(i)*sum(n));
            Temp=mean(INPUT_4);
            for j=1:c4
                var_4(j,1)=(INPUT_4(j,:)-Temp)*(INPUT_4(j,:)-Temp)';
            end
            Covar_m(i)=sum(var_4)/(Beta(i)*sum(n));
        case 5    % The number of class : 5
            Vmean(i)=sum(Delta_5)/(Beta(i)*sum(n));
            Temp=mean(INPUT_5);
            for j=1:c5
                var_5(j,1)=(INPUT_5(j,:)-Temp)*(INPUT_5(j,:)-Temp)';
            end
            Covar_m(i)=sum(var_5)/(Beta(i)*sum(n));
    end
end
Bar_Mean=sum(Beta.*Vmean)/CN;
Bar_Covar=sum(Beta.*Covar_m)/CN;
Zeta=zeros(CN,1);
for i=1:CN
    Zeta(i)=(1-1/18)*(Covar_m(i)-Bar_Mean)*sqrt(Bar_Covar)/n(i);
    % 7800 = 26 Hz * 5 mimute * 60 seconds
end
% Class 1
REGB_1=zeros(n(1),1);
for i=1:n(1)
    if Delta_1(i,1)<=Zeta(1)         % Regular = 2
        REGB_1(i,1)=2;
    elseif Delta_1(i,1)>Zeta(1)      % Irregular = 1
        REGB_1(i,1)=1;
```


www.manaraa.com

```
    end
end
RESULT_1=[Delta_1 REGB_1];
% Class 2
REGB_2=zeros(n(2),1);
for i=1:n(2)
    if Delta_2(i,1)<=Zeta(2)        % Regular = 2
        REGB_2(i,1)=2;
    elseif Delta_2(i,1)>Zeta(2)     % Irregular = 1
        REGB_2(i,1)=1;
    end
end
RESULT_2=[Delta_2 REGB_2];
% Class 3
REGB_3=zeros(n(3),1);
for i=1:n(3)
    if Delta_3(i,1)<=Zeta(3)        % Regular = 2
        REGB_3(i,1)=2;
    elseif Delta_3(i,1)>Zeta(3)     % Irregular = 1
        REGB_3(i,1)=1;
    end
end
RESULT_3=[Delta_3 REGB_3];
% Class 4
REGB_4=zeros(n(4),1);
for i=1:n(4)
    if Delta_4(i,1)<=Zeta(4)        % Regular = 2
        REGB_4(i,1)=2;
    elseif Delta_4(i,1)>Zeta(4)     % Irregular = 1
        REGB_4(i,1)=1;
    end
end
RESULT_4=[Delta_4 REGB_4];
% Class 5
REGB_5=zeros(n(5),1);
for i=1:n(5)
    if Delta_5(i,1)<=Zeta(5)        % Regular = 2
        REGB_5(i,1)=2;
    elseif Delta_5(i,1)>Zeta(5)     % Irregular = 1
        REGB_5(i,1)=1;
    end
end
RESULT_5=[Delta_5 REGB_5];
RESULT=zeros(MAX,1);
c1_index=0;c2_index=0;c3_index=0;c4_index=0;c5_index=0;
for i=1:MAX
    switch C(i)
        case 1
            c1_index=c1_index+1;
            RESULT(i,1)=REGB_1(c1_index);
        case 2
            c2_index=c2_index+1;
            RESULT(i,1)=REGB_2(c2_index);
        case 3
            c3_index=c3_index+1;
            RESULT(i,1)=REGB_3(c3_index);
        case 4
```


www.manaraa.com

```
            c4_index=c4_index+1;
            RESULT(i,1)=REGB_4(c4_index);
        case 5
            c5_index=c5_index+1;
            RESULT(i,1)=REGB_5(c5_index);

    end
end
RESULT_C=[C RESULT];        % First Column : Cluter number
                            % Second Column : Regular = 2, Irregular =
1
%% Save RESULT_C as External file 'Classifier_Results'
fid = fopen('Classifier_Results.mes','w');
[i,j]=size(RESULT_C);
for i=1:i
    fprintf(fid,'%d    %d\n', RESULT_C(i,1),RESULT_C(i,2));
end
fclose(fid);
fprintf('End of File (Classifier_Results.mes) generation~! ^^\n');
```


www.manaraa.com

B.7 MATLAB CODES FOR IRREGULAR DETECTION

```
% This function calculate true positive range, true negatiave range,
% and regular ration with breathing datasets.

% BC: Breathing Cycle
% Psi: regular threshold to decide whether the patterns is regular
% or irregular
% Range_TP: True positive range within observation period
% Rnage_TN: True negetaive range within observation period
% Ratio: regular ratio Range_TP over observation period

function [BC Psi Range_TP Range_TN Ratio]=
IrregularDetection(data_1,data_2,data_3,Time_Stamp,len)
%% Combine three channel signals into one input data
input_data = (data_1(:,4)+data_2(:,4)+data_3(:,4))/3;
%% Breathing Frequency
Min_Index=1;
MAXMIN=[zeros(len,2) NaN(len,1)];        % First Column:MAXMIN, Second
Column:Amplitude
Range = 3.5*26;  % Range(s*Hz) : searching range to detect max and min
while Min_Index~=len
    if Min_Index>len-Range  % Exit the loop if the remain is short
        break;
    end
    MAX=max(input_data(Min_Index:Min_Index+Range));
    for j=Min_Index:Min_Index+Range
        if MAX==max(input_data(j))
            MAXMIN(j,1)=2;  % Assign MAX = 2
            MAXMIN(j,2)=MAX;% Assing amplitude
            MAXMIN(j,3)=MAX;
            Max_Index=j;
        end
    end
    if Max_Index>len-Range  % Exit the loop if the remain is short
        break;
    end
    MIN=min(input_data(Max_Index:Max_Index+Range));
    for j=Max_Index:Max_Index+Range
        if MIN==min(input_data(j))
            MAXMIN(j,1)=1;  % Assign MIN = 1
            MAXMIN(j,2)=MIN;% Assign amplitude
            MAXMIN(j,3)=MIN;
            Min_Index=j;
        end
    end
end
%% Count the number of MAX and MIN
numMIN=0;numMAX=0;
Time_Fs = zeros(len,1);
for i=1:len
    if MAXMIN(i,1)==1
        numMIN = numMIN + 1;
        Time_Fs(i)= Time_Stamp(i);
    end
    if MAXMIN(i,1)==2
```


www.manaraa.com

```
        numMAX = numMAX + 1;
    end
end
%% Assign the position value to MAX and MIN (Get Psi(Irregular) number)
ExtrPos=zeros(numMIN+numMAX,6); % ExtrPos = [time, position,|MAX-MIN|]
k=0;
for i=1:len
    if MAXMIN(i,1) ~= 0
        k=k+1;
        ExtrPos(k,1)=Time_Stamp(i,1);    % Time
        ExtrPos(k,2)=MAXMIN(i,1);        % MAX or MIN
        ExtrPos(k,3)=MAXMIN(i,2);        % Position
    end
end
for i=1:numMIN+numMAX-1
    ExtrPos(i,4)=ExtrPos(i+1,3)-ExtrPos(i,3);
    ExtrPos(i,5)=sqrt((ExtrPos(i,4))^2);
end
DIthreld=0.5*mean(ExtrPos(:,5));
for i=1:numMIN+numMAX-1
    if ExtrPos(i,5)<=DIthreld
        ExtrPos(i,6)=1;
    end
end
Psi=sum(ExtrPos(:,6));
%% Detect Breathing Cycle (BC)
Time_For_Frequency = zeros(numMIN,1);
k=0;
for j=1:len
    if Time_Fs(j) ~= 0
        k = k+1;
        Time_For_Frequency(k)= Time_Fs(j);
    end
end
BreathCycle = zeros(numMIN-1,1);
for j=1:numMIN-1
    BreathCycle(j)=Time_For_Frequency(j+1)-Time_For_Frequency(j);
end
BC=mean(BreathCycle);
fprintf('Breathing Cycle(mean) =   %f\n',BC);
%% True Positives Range & True Negatives Range for a patient
Range_TP=mean(BreathCycle)*Psi/2;
Range_TN=(Time_Stamp(len)-Time_Stamp(1))-Range_TP;
fprintf('The Range of True Positive (Irregul) = %f\n',Range_TP);
fprintf('The Range of True Negative (Regular) = %f\n',Range_TN);
Ratio=Range_TN/(Range_TN+Range_TP);
if (Range_TN/(Range_TN+Range_TP))>=0.75
    fprintf('The Patient is Regular:Regular Percent = %.2f\n',...
        100*Range_TN/(Range_TN+Range_TP));
end
if (Range_TN/(Range_TN+Range_TP))<=0.5
    fprintf('The Patient is Irregular:Regular Percent = %.2f\n',...
        100*Range_TN/(Range_TN+Range_TP));
end
if 0.5<(Range_TN/(Range_TN+Range_TP)) &&
(Range_TN/(Range_TN+Range_TP))<0.75
    fprintf('The Patient is NaN case:Regular Percent = %.2f\n',...
```


www.manaraa.com

```
        100*Range_TN/(Range_TN+Range_TP));
end
%% Define the irregular Point Variable
Irr_Line=NaN(len,1);
temp=zeros(numMIN+numMAX,2);     % temp = [ time position ]
for i=1:numMIN+numMAX
    if ExtrPos(i,6)==1
        temp(i,1)=ExtrPos(i,1); % time
        temp(i,2)=ExtrPos(i,3); % position
    end
end
for i=1:numMIN+numMAX
    for j=1:len
        if temp(i,1)==Time_Stamp(j,1)
            Irr_Line(j,1)=temp(i,2);
        end
    end
end
```


www.manaraa.com

B.8 MATLAB CODES FOR DETECTION OF TRUE POSITIVE AND TRUE NEGATIVE

```
% This code detects ture positive and ture negative
% within observation periods. This code can be used as a golden
% standard in our approach for the classification.

clear;
clc
initpath;
% Patient DB17_2 for the figure 'True Positive vs. True Negative'
%% Read the external file
TOT_READ_DB94_1; % Patient i = 317

%% Combine three channel signal into one input data
input_data = (data_1(:,4)+data_2(:,4)+data_3(:,4))/3;
%% Breathing Frequency
Min_Index=1;
MAXMIN=[zeros(len,2) NaN(len,1)];        % First Column:MAXMIN, Second
Column:Amplitude
%Range = 3.5*26;  % Range(s*Hz) : searching range to detect max and min
Range = 3*26;
%Range = 4*26;
while Min_Index~=len
    if Min_Index>len-Range  % Exit the loop if the remain is short
        break;
    end
    MAX=max(input_data(Min_Index:Min_Index+Range));
    for j=Min_Index:Min_Index+Range
        if MAX==max(input_data(j))
            MAXMIN(j,1)=2;  % Assign MAX = 2
            MAXMIN(j,2)=MAX;% Assing amplitude
            MAXMIN(j,3)=MAX;
            Max_Index=j;
        end
    end
    if Max_Index>len-Range  % Exit the loop if the remain is short
        break;
    end
    MIN=min(input_data(Max_Index:Max_Index+Range));
    for j=Max_Index:Max_Index+Range
        if MIN==min(input_data(j))
            MAXMIN(j,1)=1;  % Assign MIN = 1
            MAXMIN(j,2)=MIN;% Assign amplitude
            MAXMIN(j,3)=MIN;
            Min_Index=j;
        end
    end
end
%% Count the number of MAX and MIN
numMIN=0;numMAX=0;
Time_Fs = zeros(len,1);
for i=1:len
    if MAXMIN(i,1)==1
        numMIN = numMIN + 1;
        Time_Fs(i)= Time_Stamp(i);
    end
```


www.manaraa.com

```
    if MAXMIN(i,1)==2
        numMAX = numMAX + 1;
    end
end
%% Assign the position value to MAX and MIN (Get Psi(Irregular) number)
ExtrPos=zeros(numMIN+numMAX,6); % ExtrPos = [time, position,|MAX-MIN|]
k=0;
for i=1:len
    if MAXMIN(i,1) ~= 0
        k=k+1;
        ExtrPos(k,1)=Time_Stamp(i,1);    % Time
        ExtrPos(k,2)=MAXMIN(i,1);        % MAX or MIN
        ExtrPos(k,3)=MAXMIN(i,2);        % Position
    end
end
for i=1:numMIN+numMAX-1
    ExtrPos(i,4)=ExtrPos(i+1,3)-ExtrPos(i,3);
    ExtrPos(i,5)=sqrt((ExtrPos(i,4))^2);
end
DIthreld=0.5*mean(ExtrPos(:,5));
for i=1:numMIN+numMAX-1
    if ExtrPos(i,5)<=DIthreld
        ExtrPos(i,6)=1;
    end
end
Psi=sum(ExtrPos(:,6));
%% Detect Breathing Cycle (BC)
Time_For_Frequency = zeros(numMIN,1);
k=0;
for j=1:len
    if Time_Fs(j) ~= 0
        k = k+1;
        Time_For_Frequency(k)= Time_Fs(j);
    end
end
BreathCycle = zeros(numMIN-1,1);
for j=1:numMIN-1
    BreathCycle(j)=Time_For_Frequency(j+1)-Time_For_Frequency(j);
end
fprintf('Breathing Cycle(mean) =   %f\n',mean(BreathCycle));
fprintf('Total number of Psi = %d\n',Psi);
%% True Positives Range & True Negatives Range for a patient
Range_TP=mean(BreathCycle)*Psi/2;
Range_TN=(Time_Stamp(len)-Time_Stamp(1))-Range_TP;
fprintf('The Range of True Positive (Irregul) = %.2f\n',Range_TP);
fprintf('The Range of True Negative (Regular) = %.2f\n',Range_TN);
fprintf('Ratio = %.2f\n',Range_TN/(Range_TP+Range_TN))
if (Range_TN/(Range_TN+Range_TP))>=0.75
    fprintf('The Patient is Regular:Regular Percent = %.2f\n',...
        100*Range_TN/(Range_TN+Range_TP));
end
if (Range_TN/(Range_TN+Range_TP))<=0.5
    fprintf('The Patient is Irregular:Regular Percent = %.2f\n',...
        100*Range_TN/(Range_TN+Range_TP));
end
if 0.5<(Range_TN/(Range_TN+Range_TP)) &&
(Range_TN/(Range_TN+Range_TP))<0.75
```


www.manaraa.com

```
    fprintf('The Patient is NaN case:Regular Percent = %.2f\n',...
        100*Range_TN/(Range_TN+Range_TP));
end
%% Define the irregular Point Variable
Irr_Line=NaN(len,1);
temp=zeros(numMIN+numMAX,2);     % temp = [ time position ]
for i=1:numMIN+numMAX
    if ExtrPos(i,6)==1
        temp(i,1)=ExtrPos(i,1); % time
        temp(i,2)=ExtrPos(i,3); % position
    end
end
for i=1:numMIN+numMAX
    for j=1:len
        if temp(i,1)==Time_Stamp(j,1)
            Irr_Line(j,1)=temp(i,2);
        end
    end
end
%% Draw the figures
figure;
plot(Time_Stamp(:),input_data(:),'b',Time_Stamp(:),MAXMIN(:,3),'rd',...
    Time_Stamp(:),Irr_Line(:),'go');
xlabel('Data Time Index(Second)','FontSize',18,'FontName','Arial');
ylabel('Breathing Position(cm)','FontSize',18,'FontName','Arial');
legend('Breathing curve','Extrema','Irregular point');
%%
fprintf('%.2f   ',mean(BreathCycle)); % Breathing Cycle(mean)
fprintf('%d     ',Psi); % Total number of Psi
fprintf('%.2f   ',Range_TP); % Range of True Positive (Irregul)
fprintf('%.2f   ',Range_TN); % Range of True Negative (Regular)
fprintf('%.2f\n',Range_TN/(Range_TP+Range_TN)) % Ratio
```


www.manaraa.com

B.9 MATLAB CODES FOR ROC CURVES

```
% This code generate ROC curves with different threshold Psi_th=0.8,
% 0.85, and 0.9

%clear;
%clc;
%close all;
%% Load 5m TP TN
Load_15m_TPTN_ALL;
%% Load Range TPTN
Load_Range_TPTN_ALL;    % Get all the range of regular and irregular
for all the patient.

%% Threshold
Threshold = (0.01:0.01:1.0);

Threshold=Threshold';
[n,d]=size(Threshold);
%%
Psi = 0.92;      % Psi 0.8, 0.85, 0.9, 0.92(mean)

TP_FP_Rate_92_15m = zeros(n,2); % generate TP and FP with 15 minute
for i=1:n
    Threshold_th = Threshold(i,1);
    [TP_FP_Rate_92_15m(i,1)
TP_FP_Rate_92_15m(i,2)]=TPFPRate_5m(Threshold_th,TPTN_15m,Psi,Range_TPT
N);
end
%% Draw figure
figure;
plot(TP_FP_Rate_92_15m(:,2),TP_FP_Rate_92_15m(:,1),'b:');
xlabel('FP Rate','FontSize',18,'FontName','Arial');
ylabel('TP Rate','FontSize',18,'FontName','Arial');
axis([0, 1, 0, 1]);
%%
Psi = 0.9;      % Psi 0.8, 0.85, 0.9, 0.92

TP_FP_Rate_90 = zeros(n,2);
for i=1:n
    Threshold_th = Threshold(i,1);
    [TP_FP_Rate_90(i,1)
TP_FP_Rate_90(i,2)]=TPFPRate_5m(Threshold_th,TPTN_15m,Psi,Range_TPTN);
end
%% Draw figure
figure;
plot(TP_FP_Rate_90(:,2),TP_FP_Rate_90(:,1),'b:');
xlabel('FP Rate','FontSize',18,'FontName','Arial');
ylabel('TP Rate','FontSize',18,'FontName','Arial');
axis([0, 1, 0, 1]);
%%
Psi = 0.85;      % Psi 0.8, 0.85, 0.9, 0.92

TP_FP_Rate_85 = zeros(n,2);
for i=1:n
```


www.manaraa.com

```
    Threshold_th = Threshold(i,1);
    [TP_FP_Rate_85(i,1)
TP_FP_Rate_85(i,2)]=TPFPRate_5m(Threshold_th,TPTN_15m,Psi,Range_TPTN);
end
%% Draw figure
figure;
plot(TP_FP_Rate_85(:,2),TP_FP_Rate_85(:,1),'b:');
xlabel('FP Rate','FontSize',18,'FontName','Arial');
ylabel('TP Rate','FontSize',18,'FontName','Arial');
axis([0, 1, 0, 1]);
%%
Psi = 0.8;      % Psi 0.8, 0.85, 0.9, 0.92

TP_FP_Rate_80 = zeros(n,2);
for i=1:n
    Threshold_th = Threshold(i,1);
    [TP_FP_Rate_80(i,1)
TP_FP_Rate_80(i,2)]=TPFPRate_5m(Threshold_th,TPTN_15m,Psi,Range_TPTN);
end
%% Draw figure
figure;
plot(TP_FP_Rate_80(:,2),TP_FP_Rate_80(:,1),'b:');
xlabel('FP Rate','FontSize',18,'FontName','Arial');
ylabel('TP Rate','FontSize',18,'FontName','Arial');
axis([0, 1, 0, 1]);
%%
Psi = 0.758;      % Psi 0.8, 0.85, 0.9, 0.92

TP_FP_Rate_75 = zeros(n,2);
for i=1:n
    Threshold_th = Threshold(i,1);
    [TP_FP_Rate_75(i,1)
TP_FP_Rate_75(i,2)]=TPFPRate_5m(Threshold_th,TPTN_15m,Psi,Range_TPTN);
end
%% Draw figure
figure;
plot(TP_FP_Rate_75(:,2),TP_FP_Rate_75(:,1),'b:');
xlabel('FP Rate','FontSize',18,'FontName','Arial');
ylabel('TP Rate','FontSize',18,'FontName','Arial');
axis([0, 1, 0, 1]);
%% Total Draw figure
figure
plot(TP_FP_Rate_80(:,2),TP_FP_Rate_80(:,1),'r',...
     TP_FP_Rate_85(:,2),TP_FP_Rate_85(:,1),'b-.',...
     TP_FP_Rate_90(:,2),TP_FP_Rate_90(:,1),'k:');
     %TP_FP_Rate_75(:,2),TP_FP_Rate_75(:,1),'g');
xlabel('FP Rate','FontSize',18,'FontName','Arial');
ylabel('TP Rate','FontSize',18,'FontName','Arial');
axis([0, 1, 0, 1]);
legend('\Psi_t_h = 0.8','\Psi_t_h = 0.85','\Psi_t_h = 0.9');
%% Calculate Area Under Curve (AUC)
AUC_75 = 0;
for i=2:n
    AUC_75 = AUC_75 + TP_FP_Rate_75(i,1)*((TP_FP_Rate_75(i,2))-
(TP_FP_Rate_75(i-1,2)));
end
```


www.manaraa.com

```
fprintf('AUC of TP_FP_Rate_75 = %f\n',AUC_75);
%-------
AUC_80 = 0;
for i=2:n
    AUC_80 = AUC_80 + TP_FP_Rate_80(i,1)*((TP_FP_Rate_80(i,2))-
(TP_FP_Rate_80(i-1,2)));
end
fprintf('AUC of TP_FP_Rate_80 = %f\n',AUC_80);
%-------
AUC_85 = 0;
for i=2:n
    AUC_85 = AUC_85 + TP_FP_Rate_85(i,1)*((TP_FP_Rate_85(i,2))-
(TP_FP_Rate_85(i-1,2)));
end
fprintf('AUC of TP_FP_Rate_85 = %f\n',AUC_85);
%-------
AUC_90 = 0;
for i=2:n
    AUC_90 = AUC_90 + TP_FP_Rate_90(i,1)*((TP_FP_Rate_90(i,2))-
(TP_FP_Rate_90(i-1,2)));
end
fprintf('AUC of TP_FP_Rate_90 = %f\n',AUC_90);
```


www.manaraa.com

REFERENCES

[1] P. J. Keall, G. S. Mageras, J. M. Balter, R. S. Emery, K. M. Forster, S. B. Jiang, J. M. Kapatoes, D. A. Low, M. J. Murphy, B. R. Murray, C. R. Ramsey, M. B. Van Herk, S. S. Vedam, J. W. Wong, and E. Yorke, "The management of respiratory motion in radiation oncology report of AAPM Task Group 76," *Med. Phys.*, vol. 33, no. 10, pp. 3874–3900, 2006.

[2] R. I. Berbeco, S. Nishioka, H. Shirato, G. T. Y. Chen and S. B. Jiang, "Residual motion of lung tumours in gated radiotherapy with external respiratory surrogates," *Phys. Med. Biol.*, vol. 50, no. 16, pp. 3655–3667, 2005.

[3] Y. D. Mutaf, C. J. Scicutella, D. Michalski, K. Fallon, E. D. Brandner, G. Bednarz and M. S. Huq, "A simulation study of irregular respiratory motion and its dosimetric impact on lung tumors," *Phys. Med. Biol.*, vol. 56, no. 3, pp. 845–859, 2011.

[4] D. Putra, O. C. L. Haas, J. A. Mills and K. J. Burnham, "A multiple model approach to respiratory motion prediction for real-time IGRT," *Phys. Med. Biol.*, vol. 53, no. 6, pp. 1651–1663, 2008.

[5] D. Ruan, "Kernel density estimation-based real-time prediction for respiratory motion," *Phys. Med. Biol.*, vol. 55, no. 5, pp. 1311–1326, 2010.

[6] X. Tang, G. C. Sharp and S. B. Jiang, "Fluoroscopic tracking of multiple implanted fiducial markers using multiple object tracking," *Phys. Med. Biol.*, vol. 52, no. 14, pp. 4081–4098, 2007.

[7] S. J. Lee, Y. Motai and M. Murphy, "Respiratory Motion Estimation with Hybrid Implementation of Extended Kalman Filter," *IEEE Trans. Ind. Electron.*, vol. PP, no. 99, 2011.

[8] M. Isaksson, J. Jalden, M. J. Murphy, "On using an adaptive neural network to predict lung tumor motion during respiration for radiotherapy applications," *Med. phys.*, vol. 32, no. 12, pp. 3801–3809, 2005.

[9] H. D. Kubo, P. Len, S. Minohara and H. Mostafavi, "Breathing synchronized radiotherapy program at the University of California Davis Cancer Center," *Med. Phys.*, vol. 27, no. 2, pp. 346–353, 2000.

[10] A. Schweikard, G. Glosser, M. Bodduluri, M. J. Murphy, J. R. Adler, "Robotic motion compensation for respiratory movement during radiosurgery," *Comput. Aided Surg.*, vol. 5, no. 4, pp. 263–277, 2000.

[11] L. I. Cervino, Y. Jiang, A. Sandhu and S. B. Jiang, "Tumor motion prediction with the diaphragm as a surrogate: a feasibility study," *Phys. Med. Biol.*, vol. 55, no. 9, pp. 221–229, 2010.

[12] S-M Hong, B-H Jung and D Ruan, "Real-time prediction of respiratory motion based on a local dynamic model in an augmented space," *Phys. Med. Biol.*, vol. 56, no. 6, pp. 1775–89, 2011.

[13] K. Malinowski, T. J. McAvoy, R. George, S. Dietrich and W. D. D'Souza, "Incidence of Changes in Respiration-Induced Tumor Motion and Its Relationship with Respiratory Surrogates during Individual Treatment Fractions," *Int. J. Radiat. Oncol. Biol. Phys.*, pp. 1–9, 2011.

[14] R. Zeng, J. A. Fessler and J. M. Balter, "Estimating 3-D Respiratory Motion From Orbiting Views by Tomographic Image Registration," *IEEE Trans. Med. Imag.*, vol. 26, no. 2, pp. 153–163, 2007.

[15] W. Bai and S. M. Brady, "Motion Correction and Attenuation Correction for Respiratory Gated PET Images," *IEEE Trans. Med. Imag.*, vol. 30, no. 2, pp. 351–365, 2011.

[16] D. Sarrut, B. Delhay, P. Villard, V. Boldea, M. Beuve and P. Clarysse, "A Comparison Framework for Breathing Motion Estimation Methods From 4-D Imaging," *IEEE Trans. Med. Imag.*, vol. 26, no. 12, 2007.

[17] J. Ehrhardt, R. Werner, A. Schmidt-Richberg, and H. Handels, "Statistical Modeling of 4D Respiratory Lung Motion Using Diffeomorphic Image Registration," *IEEE Trans. Med. Imag.*, vol. 30, no. 2, pp. 251–265, 2011.

[18] Q. Zhang, A. Pevsner, A. Hertanto, Y. Hu, K. E. Rosenzweig, C. C. Ling and G. S. Mageras, "A patient-specific respiratory model of anatomical motion for radiation treatment planning," *Med. Phys.*, vol. 34, no. 12, pp. 4772–4781, 2007.

www.manaraa.com

[19] K. Nakagawa, K. Yoda, Y. Masutani, K. Sasaki and K. Ohtomo, "A Rod Matrix Compensator for Small-Field Intensity Modulated Radiation Therapy: A Preliminary Phantom Study," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 5, pp. 943–946, 2007.

[20] R. Lu, R. J. Radke, L. Hong, C. Chui, J. Xiong, E. Yorke and A. Jackson, "Learning the Relationship Between Patient Geometry and Beam Intensity in Breast Intensity-Modulated Radiotherapy," *IEEE Trans. Biomed. Eng.*, vol. 53, no, 5, pp. 908–920, 2006.

[21] Y. Li and J. Lei, "A Feasible Solution to the Beam-Angle-Optimization Problem in Radiotherapy Planning With a DNA-Based Genetic Algorithm," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 3, pp. 499–508, 2010.

[22] V. Agostini, M. Knaflitz and F. Molinari, "Motion Artifact Reduction in Breast Dynamic Infrared Imaging," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 903–906, 2009.

[23] K. Bush, I. M. Gagne, S. Zavgorodni, W. Ansbacher and W. Beckham, "Dosimetric validation of Acuros XB with Monte Carlo methods for photon dose calculations," *Med. Phys.*, vol. 38, no. 4, pp. 2208–2221, 2011.

[24] L. I. Cervino, J. Du and S. B. Jiang, "MRI-guided tumor tracking in lung cancer radiotherapy," *Phy. Med. Biol.*, vol. 56, no. 13, pp. 3773–3785, 2011.

[25] E. W. Pepina, H. Wu and H. Shirato, "Dynamic gating window for compensation of baseline shift in respiratory-gated radiation therapy," *Med. Phys.*, vol. 38, no. 4, pp. 1912–1918, 2011.

[26] P. R. Poulsenb, B. Cho , A. Sawant, D. Ruan and P. J. Keall, "Detailed analysis of latencies in image-based dynamic MLC tracking," *Med. Phys.*, vol. 37, no. 9, pp. 4998–5005, 2010.

[27] T. Roland, P. Mavroidis, C. Shi and N. Papanikolaou, "Incorporating system latency associated with real-time target tracking radiotherapy in the dose prediction step," *Phys. Med. Biol.*, vol. 55, no. 9, pp. 2651–2668, 2010.

[28] J. Jacq, C. Schwartz, V. Burdin, R. Gérard, C. Lefévre, C. Roux and O. Rémy-Néris, "Building and Tracking Root Shapes," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 3, pp. 696–707, 2010.

[29] H. Paganetti, "The Use of Computational Patient Models to Assess the Risk of Developing Radiation-Induced Cancers From Radiation Therapy of the Primary Cancer," *Proceedings of the IEEE*, vol. 97, no. 12, pp. 1977–1987, 2009.

[30] A. L. Maitre, W. P. Segars, S. Marache, A. Reilhac, M. Hatt, S. Tomei, C. Lartizien and D. Visvikis, "Incorporating Patient-Specific Variability in the Simulation of Realistic Whole-Body 18F-FDG Distributions for Oncology Applications," *Proceedings of the IEEE*, vol. 97, no. 12, pp. 2026–2038, 2009.

[31] A. Kalet, G. Sandison, H. Wu and R. Schmitz, "A state-based probabilistic model for tumor respiratory motion prediction," *Phys. Med. Biol.*, vol. 55, no. 24, pp. 7615–7631, 2010.

[32] D. Ruan and P. Keall, "Online prediction of respiratory motion: multidimensional processing with low-dimensional feature learning," *Phys. Med. Biol.*, vol. 55, no. 11, pp. 3011–3025, 2010.

[33] N. Riaz, P. Shanker, R. Wiersma, O. Gudmundsson, W. Mao, B. Widrow and L. Xing, "Predicting respiratory tumor motion with multi-dimensional adaptive filters and support vector regression," *Phys. Med. Biol.*, vol. 54, no. 19, pp. 5735–5748, 2009.

[34] R. Wernera, J. Ehrhardt, R. Schmidt and H. Handels, "Patient-specific finite element modeling of respiratory lung motion using 4D CT image data," *Med. Phys.*, vol. 36, no. 5, pp. 1500–1510, 2009.

[35] M. J. Murphy and D. Pokhrel, "Optimization of an adaptive neural network to predict breathing," *Med. Phys.*, vol. 36, no. 1, pp. 40–47, 2009.

[36] M. J. Murphy and S. Dieterich, "Comparative performance of linear and nonlinear neural networks to predict irregular breathing," *Phys. Med. Biol.*, vol. 51, no. 22, pp. 5903–5914, 2006.

[37] S. S. Vedam, P. J. Keall, A. Docef, D. A. Todor, V. R. Kini and R. Mohan, "Predicting respiratory motion for four-dimensional radiotherapy," *Med. Phys.*, vol. 31, no. 8, pp. 2274–2283, 2004.

[38] D. Yang, W. Lu, D. A. Low, J. O. Deasy, A. J. Hope and I. El Naqa, "4D-CT motion estimation using deformable image registration and 5D respiratory motion modeling," *Med. Phys.*, vol. 35, no. 10, pp. 4577–4590, 2008.

www.manaraa.com

[39] M. Schwarz, J. V. D. Geer, M. V. Herk, J. V. Lebesque, B. J. Mijnheer and E. M. F. Damen, "Impact of geometrical uncertainties on 3D CRT and IMRT dose distributions for lung cancer treatment," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 65, no. 4, pp. 1260–1269, 2006.

[40] M. Kakar, H. Nystr¨om, L. R. Aarup, T. J. Nøttrup and D. R. Olsen, "Respiratory motion prediction by using the adaptive neuro fuzzy inference system (ANFIS)," *Phys. Med. Biol.*, vol. 50, no. 19, pp. 4721–4728, 2005.

[41] J. Wilbert, J. Meyer, K. Baier, M. Guckenberger , C. Herrmann , R. Hess , C. Janka , L. Ma , T. Mersebach , A. Richter , M. Roth , K. Schilling , M. Flentje, "Tumor tracking and motion compensation with an adaptive tumor tracking system (ATTS) System description and prototype testing," *Med. Phys.*, vol. 35, no. 9, pp. 3911–3921, 2008.

[42] I. Buzurovic, T. K. Podder, K. Huang and Y. Yu, "Tumor Motion Prediction and Tracking in Adaptive Radiotherapy," *IEEE Int. Conf. on Bioinformatics and Bioengineering*, pp. 273–278, 2010.

[43] I. Buzurovic, K. Huang, Y. Yu and T. K. Podder, "A robotic approach to 4D real-time tumor tracking for radiotherapy," *Phys. Med. Biol.*, vol. 56, no. 5, pp. 1299–1318, 2011.

[44] P. S. Verma, H. Wu, M. P. Langer, I. J. Das and G. Sandison, "Survey: Real-time tumor motion prediction for Image Guided Radiation Treatment," *Computing in Science & Engineering*, vol. 13, no. 5, pp. no. 24–35, 2011.

[45] D. Putra, O. C. L. Haas, J. A. Mills and K. J. Bumham, "Prediction of Tumour Motion using Interacting Multiple Model Filter," *Int. Conf. Advances in Medical, Signal and Information Processing*, pp. 1–4, 2006.

[46] G. C. Sharp, S. B. Jiang, S. Shimizu, and H. Shirato, "Prediction of respiratory tumour motion for real-time image-guided radiotherapy," *Phys. Med. Biol.*, vol. 49, no. pp. 425–440, 2004.

[47] Q. Ren, S. Nishioka, H. Shirato and R. I. Berbeco, "Adaptive prediction of respiratory motion for motion compensation radiotherapy," *Phys. Med. Biol.*, vol. 52, no. 22, pp. 6651–6661, 2007.

[48] K. C. McCall and R. Jeraj, "Dual-component model of respiratory motion based on the periodic autoregressive moving average (periodic ARMA) method," *Phys. Med. Biol.*, vol. 52, no. 12, pp. 3455–66, 2007.

[49] S. B. Jiang, "Radiotherapy of Mobile Tumors," *Semin. Radiat. Oncol.*, vol. 16, no. 4, pp. 239–248, 2006.

[50] C. Ozhasoglu and M. J. Murphy, "Issues in respiratory motion compensation during external-beam radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 52, no. 5, pp. 1389–1399, 2002.

[51] H. Shirato, S. Shimizu, T. Kunieda, K. Kitamura, M. van Herk, K. Kagei, T. Nishioka, S. Hashimoto, K. Fujita, H. Aoyama, K. Tsuchiya, K. Kudo and K. Miyasaka, "Physical aspects of a real-time tumor-tracking system for gated radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 48, no. 4, pp. 1187–1195, 2000.

[52] H. Shirato, S. Shimizu, K. Kitamura, T. Nishioka, K. Kagei, S. Hashimoto, H. Aoyama, T. Kunieda, N. Shinohara, H. Dosaka-Akita and K. Miyasaka, "Four-dimensional treatment planning and fluoroscopic real-time tumor tracking radiotherapy for moving tumor," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 48, no. 2, pp. 435–442, 2000.

[53] H. Wu, G. C. Sharp, B. Salzberg, D. Kaeli, H. Shirato and S. B. Jiang, "A finite state model for respiratory motion analysis in image guided radiation therapy," *Phys. Med. Biol.*, vol. 49, no. 23, pp. 5357–5372, 2004.

[54] M. J. Murphy, "Tracking moving organs in real time," *Semin. Radiat. Oncol.*, vol. 14, no. 1, pp. 91–100, 2004.

[55] http://www.cancer.gov/cancertopics/coping/radiation-therapy-and-you/

[56] http://www.morsecyberknife.com/CyberKnife-System

[57] http://www.varian.com/us/oncology/radiation_oncology/

[58] K. M. Langen and D. T. L. Jones, "Organ motion and its management," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 50, no. 1, pp. 265–278, 2001.

[59] H. Himberg and Y. Motai, "Head Orientation Prediction: Delta Quaternions Versus Quaternions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1382–1392, 2009.

www.manaraa.com

[60] P. Keall, S. Vedam, R. George, C. Bartee, J. Siebers, F. Lerma, E. Weiss and T. Chung, "The Clinical Implementation of Respiratory-Gated Intensity-Modulated Radiotherapy," *Medical Dosimetry*, vol. 31, no. 2, pp. 152–162, 2006.

[61] E. Weiss, K. Wijesooriya, S. V. DILL and P. J. Keall, "Tumor and normal tissue motion in the thorax during respiration: Analysis of volumetric and positional variations using 4D CT," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 67, no. 1, pp. 296–307, 2007.

[62] E. Weiss, K. Wijesooriya, V. Ramakrishnan and P. J. Keall, "Comparison of intensity-modulated radiotherapy planning based on manual and automatically generated contours using deformable image registration in four-dimensional computed tomography of lung cancer patients," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 70, no. 2, pp. 572–581, 2008.

[63] E. Weissa, K. Wijesooriya and P. Keall, "Esophagus and spinal cord motion relative to GTV motion in four-dimensional CTs of lung cancer patients," *Radiotherapy and Oncology*, vol. 87, no. 1, pp. 44–48, 2008.

[64] Y. Suh, E. Weiss, H. Zhong, M. Fatyga, J. V. Siebers, and P. J. Keall, "A deliverable four-dimensional intensity-modulated radiation therapy-planning method for dynamic multileaf collimator tumor tracking delivery," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 71, no. 5, pp. 1526–1536, 2008.

[65] G. D. Hugo, E. Weiss, A. Badawi and M. Orton, "Localization Accuracy of the Clinical Target Volume during Image-Guided Radiotherapy of Lung Cancer," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 81, no. 2, pp. 560–567, 2011.

[66] http://www.medical.siemens.com

[67] J. E. Bayouth, "Siemens Multileaf Collimator Characterization and Quality Assurance Approaches for Intensity-Modulated Radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 71, no. 1, pp. S93–S97, 2008.

[68] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis and M. Welsh, "Wireless Sensor Networks for Healthcare," *Proceeding of the IEEE*, vol. 98, no. 11, pp. 1947–1960, 2010.

[69] S. T. Shivappa, M. M. Trivedi and B. D. Rao, "Audiovisual Information Fusion in Human–Computer Interfaces and Intelligent Environments: A Survey," *Proceeding of the IEEE*, vol. 98, no. 10, pp. 1692–1715, 2010.

[70] G. S. Stamatakos, D. D. Dionysiou, E. I. Zacharaki, N. A. Mouravliansky, K. S. Nikita and N. K. Uzunoglu, "In Silico Radiation Oncology: Combining Novel Simulation Algorithms With Current Visualization Techniques," *Proceeding of the IEEE*, vol. 90, no. 11, pp. 1764–1777, 2002.

[71] S. J. McQuaid, T. Lambrou, V. J. Cunningham, V. Bettinardi, M. C. Gilardi and B. F. Hutton, "The Application of a Statistical Shape Model to Diaphragm Tracking in Respiratory-Gated Cardiac PET Images," *Proceedings of the IEEE*, vol. 97, no. 12, pp. 2039–2052, 2009.

[72] H. Zaidi and B. M. W. Tsui, "Review of Computational Anthropomorphic Anatomical and Physiological Models," *Proceedings of the IEEE*, vol. 97, no. 12, pp. 1938–1953, 2009.

[73] M. Niedre and V. Ntziachristos, "Elucidating Structure and Function In Vivo With Hybrid Fluorescence and Magnetic Resonance Imaging," *Proceedings of the IEEE*, vol. 96, no. 3, pp. 382–396, 2008.

[74] T. Harada, H. Shirato, S. Ogura, S. Oizumi, K. Yamazaki, S. Shimizu, R. Onimaru, K. Miyasaka, M. Nishimura and H. Dosaka-Akita, "Real-time tumor-tracking radiation therapy for lung carcinoma by the aid of insertion of a gold marker using bronchofiberscopy," *Cancer*, vol. 95, no. 8, pp. 1720–1727, 2002.

[75] S. S. Vedam, V. R. Kini, P. J. Keall, V. Ramakrishnan, H. Mostafavi and R. Mohan, "Quantifying the predictability of diaphragm motion during respiration with a noninvasive external marker," *Med. Phys.*, vol. 30, no. 4, pp. 505–513, 2003.

[76] J. He, G. J. O'Keefe, S. J. Gong, G. Jones, T. Saunder, A. M. Scott and M. Geso, "A Novel Method for Respiratory Motion Gated With Geometric Sensitivity of the Scanner in 3D PET," *IEEE Trans. Nuclear Science*, vol. 55, no. 5, pp. 2557–2565, 2008.

[77] http://www.elekta.com/healthcare_international_elekta_oncology.php

www.manaraa.com

[78] H. Tadayyon, A. Lasso, A. Kaushal, P. Guion and G. Fichtinger, "Target Motion Tracking in MRI-guided Transrectal Robotic Prostate Biopsy," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 11, pp. 3135–3142, 2011.

[79] J. R. Wong, L. Grimm, M. Uematsu, R. Oren, C. W. Cheng, S. Merrick and P. Schiff, "Image-guided radiotherapy for prostate cancer by CT-linear accelerator combination: prostate movements and dosimetric considerations," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 61, no. 2, pp. 561–569, 2005.

[80] M. J. Fitzpatrick, G. Starkschall, J. A. Antolak, J. Fu, H. Shukla, P. J. Keall, P. Klahr and R. Mohan, "Displacement-based binning of time-dependent computed tomography image data sets," *Med. Phys.*, vol. 33, no. 1, pp. 235–246, 2006.

[81] J. Ehrhardt, R. Werner, D. Säring, T. Frenzel, W. Lu, D. Low and H. Handels, "An optical flow based method for improved reconstruction of 4D CT data sets acquired during free breathing," *Med. Phys.*, vol. 34, no. 2, pp. 711–721, 2007.

[82] L. I. Cerviño, A. K. Y. Chao, A. Sandhu and S. B. Jiang, "The diaphragm as an anatomic surrogate for lung tumor motion," *Phys. Med. Biol.*, vol. 54, no. 11, pp. 3529–3541, 2009.

[83] F. O. Spoelstra , J. R. van Sörnsen de Koste, A. Vincent, J. P. Cuijpers, B. J. Slotman and S. Senan, "An evaluation of two internal surrogates for determining the three-dimensional position of peripheral lung tumors," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 74, no. 2, pp 623–629, 2009.

[84] D. Ruan, "Prospective detection of large prediction errors: a hypothesis testing approach," *Phys. Med. Biol.*, vol. 55, no. 13, pp 3885–3904, 2010.

[85] P. R. Poulsen , B. Cho, D. Ruan, A. Sawant and P. J. Keall, "Dynamic multileaf collimator tracking of respiratory target motion based on a single kilovoltage imager during arc radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 77, no. 2, pp. 600–607, 2010.

[86] A. Sawant, R. L. Smith, R. B. Venkat, L. Santanam, B. Cho, P. Poulsen, H. Cattell, L. J. Newell, P. Parikh and P. J. Keall, "Toward submillimeter accuracy in the management of intrafraction motion: the integration of real-time internal position monitoring and multileaf collimator target tracking," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 74, no. 2, pp. 575–582, 2009.

[87] D. Ruan, J. A. Fessler and J. M. Balter, "Real-time prediction of respiratory motion based on local regression methods," *Phys. Med. Biol.*, vol. 52, no. 23, pp. 7137–7152, 2007.

[88] A. D. Vandermeer, H. Alasti, Y. B. Cho and B. Norrlinger, "Investigation of the dosimetric effect of respiratory motion using four-dimensional weighted radiotherapy," *Phys. Med. Biol.*, vol. 52, no. 15, pp. 4427–4448, 2007.

[89] E. Chin and K. Otto, "Investigation of a novel algorithm for true 4D-VMAT planning with comparison to tracked, gated and static delivery," *Med. Phys.*, vol. 38, no. 5, pp. 2698–2707, 2011.

[90] A. A. Patel, J. A. Wolfgang, A. Niemierko, T. S. Hong, T. Yock, N. C. Choi, "Implications of respiratory motion as measured by four-dimensional computed tomography for radiation treatment planning of esophageal cancer," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 74, no. 1, pp. 290–296, 2009.

[91] Q. J. Wu, D. Thongphiew, Z. Wang, V. Chankong and F. F. Yin, "The impact of respiratory motion and treatment technique on stereotactic body radiation therapy for liver cancer," *Med. phys.*, vol. 5, no. 4, pp. 1440–1451, 2008.

[92] T. Depuydt, D. Verellen, O. Haas, T. Gevaert, N. Linthout, M. Duchateau, K. Tournel, T. Reynders, K. Leysen, M. Hoogeman, G. Storme, M. De Ridder, "Geometric accuracy of a novel gimbals based radiation therapy tumor tracking system," *Radiother. Oncol.*, vol. 98, no. 3, pp. 365–372, 2011.

[93] C. Shi and N. Papanikolaou, "Tracking versus Gating in the Treatment of Moving Targets," *European Oncological Disease*, pp. 83–86, 2007.

[94] D. Ramanan, D. A. Forsyth and A. Zisserman, "Tracking People by Learning Their Appearance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 65–81, 2007.

[95] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Büla and P. Robert, "Ambulatory System for Human Motion Analysis Using a Kinematic Sensor: Monitoring of Daily Physical Activity in the Elderly," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 6, pp. 711–723, 2003.

[96] Y.-M. Kuo, J.-S. Lee and P.-C. Chung, "A Visual Context-Awareness-Based Sleeping-Respiration Measurement System," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 255–265, 2010.

www.manaraa.com

[97] K. Punithakumar, I. B. Ayed, A. Islam, I. G. Ross and S. Li, "Tracking Endocardial Motion Via Multiple Model Filtering," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 8, pp. 2001–2010, 2010.

[98] H. Ghasemzadeh, V. Loseu and R. Jafari, "Structural Action Recognition in Body Sensor Networks: Distributed Classification Based on String Matching," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 425–435, 2010.

[99] K. Punithakumar, I. B. Ayed, I. G. Ross, A. Islam, J. Chong and S. Li, "Detection of Left Ventricular Motion Abnormality Via Information Measures and Bayesian Filtering," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 4, pp. 1106–1113, 2010.

[100] A. P. King, K. S. Rhode, R. S. Razavi and T. R. Schaeffter, "An Adaptive and Predictive Respiratory Motion Model for Image-Guided Interventions: Theory and First Clinical Application," *IEEE Trans. Med. Imag.*, vol. 28, no. 12, pp. 2020–2032 , 2009.

[101] N. A. Ablitt, J. Gao, J. Keegan, L. Stegger, D. N. Firmin and G.-Z. Yang, "Predictive Cardiac Motion Modeling and Correction With Partial Least Squares Regression," *IEEE Trans. Med. Imag.*, vol. 23, no. 10, pp. 1315–1324, 2004.

[102] A. S. Naini, T.-Y. Lee, R. V. Patel and A. Samani, "Estimation of Lung's Air Volume and Its Variations Throughout Respiratory CT Image Sequences," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 1, pp. 152–158, 2011.

[103] Y. K. Park , S. Kim, H. Kim, I. H. Kim, K. Lee and S. J. Ye, "Quasi-breath-hold technique using personalized audio-visual biofeedback for respiratory motion management in radiotherapy," *Med. Phys.*, vol. 38, no. 6, pp. 3114–3124, 2011.

[104] X. Wang , R. A. Amos, X. Zhang, P. J. Taddei, W. A. Woodward, K. E. Hoffman, T. K. Yu, W. Tereffe, J. Oh, G. H. Perkins, M. Salehpour, S. X. Zhang, T. L. Sun, M. Gillin, T. A. Buchholz, E. A. Strom, "External-beam accelerated partial breast irradiation using multiple proton beam configurations," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 80, no. 5, pp. 1464–1472, 2011.

[105] E. Johnston , M. Diehn, J. D. Murphy, B. W. Jr Loo, P. G. Maxim, "Reducing 4D CT artifacts using optimized sorting based on anatomic similarity," *Med. Phys.*, vol. 38, no. 5, pp. 2424–2429, 2011.

[106] H. Liu , Q. Wu, "Dosimetric and geometric evaluation of a hybrid strategy of offline adaptive planning and online image guidance for prostate cancer radiotherapy," *Phys. Med. Biol.*, vol. 56, no. 15, pp. 5045–5062, 2011.

[107] C. Ling, P. Zhang, T. Etmektzoglou, J. Star-Lack, M. Sun, E. Shapiro and M. Hunt, "Acquisition of MV-scatter-free kilovoltage CBCT images during RapidArc™ or VMAT," *Radiother. Oncol.*, vol. 100, no. 1, pp. 145–149, 2011.

[108] R. D. Wiersma, W. Mao, L. Xing, "Combined kV and MV imaging for real-time tracking of implanted fiducial markers," *Med. Phys.*, vol. 35, no. 4, pp. 1191–1198, 2008.

[109] P. J. Keall, A. D. Todor, S. S. Vedam, C. L. Bartee, J. V. Siebers, V. R. Kini, R. Mohan, "On the use of EPID-based implanted marker tracking for 4D radiotherapy," *Med. Phys.*, vol. 31, no. 12, pp. 3492 –3499, 2004.

[110] J. H. Lewis, R. Li, X. Jia, W. T. Watkins, Y. Lou, W. Y. Song and S. B. Jiang, "Mitigation of motion artifacts in CBCT of lung tumors based on tracked tumor motion during CBCT acquisition," *Phys. Med. Biol.*, vol. 56, no. 17, pp. 5485–5502, 2011.

[111] Q. Zhang, Y. C. Hu, F. Liu, K. Goodman, K. E. Rosenzweig and G. S. Mageras, "Correction of motion artifacts in cone-beam CT using a patient-specific respiratory motion model," *Med. Phys.*, vol. 37, no. 6, pp. 2901–2909, 2010.

[112] P. R. Poulsen, W. Fledelius, Paul J. Keall, E. Weiss, J. Lu, E. Brackbill and G. D. Hugo, "A method for robust segmentation of arbitrarily shaped radiopaque structures in cone-beam CT projections," *Med. Phys.*, vol. 38, no. 4, pp. 2151–2156, 2011.

[113] J. Dey, W. P. Segars, P. H. Pretorius, R. P. Walvick, P. P. Bruyant, S. Dahlberg, M. A. King, "Estimation and correction of cardiac respiratory motion in SPECT in the presence of limited-angle effects due to irregular respiration," *Med. Phys.*, vol. 37, no. 12, pp. 6453–6465, 2010.

[114] P. J. Roach, D. J. Gradinscak, G. P. Schembri, E. A. Bailey, K. P. Willowson and D. L. Bailey, "SPECT/CT in V/Q Scanning," *Semin. Nucl. Med.*, vol. 40, no. 6, pp 455–466, 2010.

www.manaraa.com

[115] J. G. Parker, B. A. Mair, D. R. Gilland, "Respiratory motion correction in gated cardiac SPECT using quaternion-based, rigid-body registration," *Med. Phys.*, vol. 36, no. 10, pp. 4742–4754, 2009.

[116] H. Ue, H. Haneishi, H. Iwanaga and K. Suga, "Nonlinear motion correction of respiratory-gated lung SPECT images," *IEEE Trans. Med. Imag.*, vol. 25, no. 4, pp. 486–495, 2006.

[117] T. O. J. Fuchs, M. Kachelriess and W. A. Kalender, "Fast Volume Scanning Approaches by X-Ray-Computed Tomography," *Proceedings of the IEEE*, vol. 91, no. 10, pp. 1492–1502, 2003.

[118] S. R. Cherry, A. Y. Louie and R. E. Jacobs, "The Integration of Positron Emission Tomography With Magnetic Resonance Imaging," *Proceedings of the IEEE*, vol. 96, no. 3, pp. 416–438, 2008.

[119] R. M. Lewitt and S. Matej, "Overview of methods for image reconstruction from projections in emission computed tomography," *Proceedings of the IEEE*, vol. 91, no. 10, pp. 1588–1611, 2003.

[120] M. Dawood, N. Lang, X. Jiang and K. P. Schäfers, "Lung Motion Correction on Respiratory Gated 3-D PET/CT Images," *IEEE Trans. Med. Imag.*, vol. 25, no. 4, pp. 476–485, 2006.

[121] J. A. van Dalen, W. V. Vogel, F. H. M. Corstens, and W. J.G. Oyen, "Multi-modality nuclear medicine imaging: artefacts, pitfalls and recommendations," *Cancer Imaging*, vol. 7, no. 1, pp. 77–83, 2007.

[122] L. Walsh, M. Morgia, A. Fyles and M. Milosevic, "Technological advances in radiotherapy for cervical cancer," *Curr. Opin. Oncol.*, vol. 23, no. 5, pp. 512–518, 2011.

[123] J. T. Hepel and D. E. Wazer, "A comparison of brachytherapy techniques for partial breast irradiation," *Brachytherapy*, In Press, 2011.

[124] E. W. Pepin, H. Wu, Y. Zhang and B. Lord, "Correlation and prediction uncertainties in the CyberKnife Synchrony respiratory tracking system," *Med. Phys.*, vol. 38, no. 7, pp. 4036–4044, 2011.

[125] http://www.biostatistics.vcu.edu/people/faculty/sun.html

[126] D. A. Low , P. J. Parikh, W. Lu, J. F. Dempsey, S. H. Wahab, J. P. Hubenschmidt, M. M. Nystrom, M. Handoko, J. D. Bradley, "Novel breathing motion model for radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 63, no. 3, pp. 921–929, 2005.

[127] H. Yan, F. F. Yin, G. P. Zhu, M. Ajlouni and J. H. Kim, "Adaptive prediction of internal target motion using external marker motion: a technical study," *Phys. Med. Biol.*, vol. 51, no. 1, pp. 31–44, 2006.

[128] F. Wang and V. Balakrishnan, "Robust Steady-State Filtering for Systems With Deterministic and Stochastic Uncertainties," *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2550–2558, 2003.

[129] S. Vedam, A. Docef, M. Fix, M. Murphy and P. Keall, "Dosimetric impact of geometric errors due to respiratory motion prediction on dynamic multileaf collimator-based four-dimensional radiation delivery," *Med. Phys.*, vol. 32, no. 6, pp. 1607–1620, 2005.

[130] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 3rd ed., 1996.

[131] G. D. Hugo, J. Liang, J. Campbell and D. Yan, "Online target position localization in the presence of respiration: a comparison of two methods," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 69, no. 5. pp. 1634–1641, 2007.

[132] B. P. Thompson, G. D. Hugo, "Quality and accuracy of cone beam computed tomography gated by active breathing control," *Med. Phys.*, vol. 35, no. 12, pp. 5595–5608, 2008.

[133] G. D. Hugo, J. Campbell, T. Zhang, D. Yan, "Cumulative Lung Dose for Several Motion Management Strategies as a Function of Pretreatment Patient Parameters," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 74, no. 2, pp. 593–601, 2009.

[134] P. Zhang, G. D. Hugo, D. Yan, "Planning study comparison of real-time target tracking and four-dimensional inverse planning for managing patient respiratory motion," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 72, no. 4, pp. 1221–1227, 2008.

[135] R. S. Brock, A. Docef, M. J. Murphy, "Reconstruction of a cone-beam CT image via forward iterative projection matching," *Med. Phys.*, vol. 37, no. 12, pp. 6212–6220, 2010.

[136] Guy Tchoupo and Alen Docef, "Nonlinear Set Membership Time series Prediction of Breathing," *European Signal Processing Conf.*, 2008.

www.manaraa.com

[137] F. Ernst and A. Schweikard, "Forecasting respiratory motion with accurate online support vector regression (SVRpred)," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 4, no. 5, pp. 439–447, 2009.

[138] W. D. D'Souza, K. Malinowski, H. H. Zhang, "Machine Learning for Intra-Fraction Tumor Motion Modeling with Respiratory Surrogates," *Int. Conf. Machine Learning and Applications*, pp. 463–467, 2009.

[139] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support Vector Regression Machines," *Advances in Neural Information Processing Systems* 9, pp. 155–161, 1996.

[140] S. Chen, S. Zhou, F. F. Yin, L. B. Marks and S. K. Das, "Investigation of the support vector machine algorithm to predict lung radiation-induced pneumonitis," *Med. Phys.*, vol. 34, no. 10, pp. 3808–3814, 2007.

[141] L. Ma, C. Herrmann and K. Schilling, "Modeling and prediction of lung tumor motion for robotic assisted radiotherapy," *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 189–194, 2007.

[142] L. Brewster, R. Mohan, G. Mageras, C. Burman, S. Leibel and Z. Fuks, "Three dimensional conformal treatment planning with multileaf collimators," *Int. J. Radia. Oncol. Biol. Phys.*, vol. 33, no. 5, pp. 1081–1089, 1995.

[143] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller," *IEEE Trans. Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 404–418, 1990.

[144] http://medicalphysicsweb.org/cws/companies/category/165

[145] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks," *IEEE Trans. Neural Netw.*, vol. 5, no 2, pp 279–297, 1994.

[146] S. Haykin, *Neural Networks and Learning Machines*, 3rd edn, Pearson, 2009.

[147] J. Meyer, A. Richter, K. Baier, J. Wilbert, M. Guckenberger and M. Flentje, "Tracking moving objects with megavoltage portal imaging: A feasibility study," *Med. Phys.*, vol. 33, no. 5, pp. 1275–1280, 2006.

[148] W. Lu, M. M. Nystrom, P. J. Parikh, D. R. Fooshee, J. P. Hubenschmidt, J. D. Bradley, and D. A. Low, "A semi-automatic method for peak and valley detection in free-breathing respiratory waveforms," *Med. Phys.*, vol. 33, no. 10, pp. 3634–3636, 2006.

[149] L. Huang, K. Park, T. Boike, P. Lee, L. Papiez, T. Solberg, C. Ding and R. D. Timmerman, "A study on the dosimetric accuracy of treatment planning for stereotactic body radiation therapy of lung cancer using average and maximum intensity projection images," *Radiotherapy and Oncology*, vol. 96, no. 1, pp. 48–54, 2010.

[150] B. Guo, X. G. Xu and C. Shi, "Real time 4D IMRT treatment planning based on a dynamic virtual patient model: proof of concept," *Med. phys.*, vol. 38, no, 5, pp. 2639–2650, 2011.

[151] A. Filler, "The History, Development and Impact of Computed Imaging in Neurological Diagnosis and Neurosurgery_CT, MRI, and DTI," *Internet J. Neurosurgery*, vol. 7, no. 1, pp. 1–69, 2009.

[152] https://rpop.iaea.org.

[153] A. P. Dhawan, B. D'Alessandro and X. Fu, "Optical Imaging Modalities for Biomedical Applications," *IEEE Reviews in Biomed. Eng.*, vol. 3, pp. 69–92, 2010.

[154] A. P. Gibson, J. C. Hebden and S. R. Arridge, "Recent advances in diffuse optical imaging," *Phys. Med. Biol.*, vol. 50, no. 4, pp. R1–R43, 2005.

[155] W. D. D'Souza, S. A. Naqvi and C. X. Yu, "Real-time intra-fraction-motion tracking using the treatment couch: a feasibility study," *Phys. Med. Biol.*, vol. 50, no. 17, pp. 4021–4033, 2005.

[156] S. Han-Oh, B. Y. Yi, F. Lerma, B. L. Berman, M. Gui and C. Yu, "Verification of MLC based real-time tumor tracking using an electronic portal imaging device," *Med. Phys.*, vol. 37, no. 6, pp. 2435–2440, 2010.

[157] B. Y. Yi, S. Han-Oh, F. Lerma, B. L. Berman and C. Yu, "Real-time tumor tracking with preprogrammed dynamic multileaf-collimator motion and adaptive dose-rate regulation," *Med. Phys.*, vol. 35, no. 9, pp. 3955–3962, 2008.

www.manaraa.com

[158] T. Gevaert , D. Verellen , B. Engels , T. Depuydt , K. Heuninckx , K. Tournel , M. Duchateau , T. Reynders and M. De Ridder, "Clinical Evaluation of a Robotic 6-Degree of Freedom Treatment Couch for Frameless Radiosurgery," *Int. J. Radiat. Oncol.*, In Press, 2011.
[159] A. Schweikard, H. Shiomi and J. Adler, "Respiration tracking in radiosurgery," *Med. Phys.*, vol. 31, no. 10, pp. 2738–2741, 2004.
[160] H. Ghasemzadeh and R. Jafari, "Physical Movement Monitoring Using Body Sensor Networks: A Phonological Approach to Construct Spatial Decision Trees," *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 66–77, 2011.
[161] H. Chen and Yo. Li, "Enhanced Particles with Pseudolikelihoods for Three-Dimensional Tracking," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 2992–2997, 2009.
[162] D. Smith and S. Singh, "Approaches to Multisensor Data Fusion in Target Tracking: A Survey," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1696–1710, 2006.
[163] D. Naso, B. Turchiano and P. Pantaleo, "A Fuzzy-Logic Based Optical Sensor for Online Weld Defect-Detection," *IEEE Trans. Ind. Informat*., vol. 1, no. 4, pp. 259–273, 2005.
[164] T. Mukai and M. Ishikawa, "An Active Sensing Method Using Estimated Errors for Multisensor Fusion Systems," *IEEE Trans. Ind. Electron.*, vol. 43, no 3, pp. 380–386, 1996.
[165] J. Liu, J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed State Representation for Tracking Problems in Sensor Networks," *Proc. Information Processing in Sensor Networks*, pp. 234–242, 2004.
[166] K. Zhou and S. I. Roumeliotis, "Optimal Motion Strategies for Range-Only Constrained Multisensor Target Tracking," *IEEE Trans. Robotics*, vol. 24, no 5, pp. 1168–1185, 2008.
[167] J. G. García, J. G. Ortega, A. S. García, and S. S. Martínez, "Robotic Software Architecture for Multisensor Fusion System," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 766–777, 2009.
[168] N. Bellotto, and Huosheng Hu, "Multisensor-Based Human Detection and Tracking for Mobile Service Robots," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 1, pp. 167–181, 2009.
[169] T. Kirubarajan, H. Wang, Y. Bar-Shalom, and K. R. Pattipati, "Efficient multisensor fusion using multidimensional data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 386–400, 2001.
[170] Z. Khan, T. Balch, and F. Dellaert, "MCMC Data Association and Sparse Factorization Updating for Real Time Multitarget Tracking with Merged and Multiple Measurements," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1960–1972, 2006.
[171] Lang Hong, Shan Cong, and D. Wicker, "Distributed multirate interacting multiple model fusion (DMRIMMF) with application to out-of-sequence GMTI data," *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 102–107, 2004.
[172] Samuel Blackman and Robert Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
[173] X. R. Li, V. P. Jilkov, "Survey of Maneuvering Target Tracking – Part V: Multiple Model Methods", *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp.1255–1321, 2005.
[174] E. Mazor, A. Averbuch, Y. Bar-Shalom and J. Dayan "Interacting multiple model methods in target tracking: a survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp.103–123, 1998.
[175] A. K. Jana, "A Hybrid FLC-EKF Scheme for Temperature Control of a Refinery Debutanizer Column," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 25–35, 2010.
[176] L. C. Yang, J. H. Yang, E. M. Feron, "Multiple Model Estimation for Improving Conflict Detection algorithms," *IEEE Conf. Systems, Man and Cybernetecis*, vol. 1, pp.242–249, 2004.
[177] H. Bom, Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp.780–783, 1988.
[178] L. Campo, P. Mookerjee, Y. Bar-Shalom, "State estimation for systems with sojourn-time-dependent markov model switching," *IEEE Trans. Autom. Control*, vol. 36, no. 2, pp. 238–243, 1991.
[179] X. R. Li, Y. Zhang, "Numerically robust implementation of multiple-model algorithms," *IEEE Trans.Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 266–278, 2000.
[180] X. R. Li, Z. Zhao and X. Li, "General Model-Set Design Methods for Multiple-Model Approach," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1260–1276, 2005.
[181] L. Hong, "Multirate interacting multiple model filtering for target tracking using multirate models," *IEEE Trans. Autom. Control*, vol. 44, no. 7, pp. 1326–1340, 1999.
[182] W. Farrell, "Interacting multiple model filter for tactical ballistic missile tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 418–426, 2008.

www.manaraa.com

[183] X. R. Li, and Y. Bar-Shalom, "Performance prediction of the interacting multiple model algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 3, pp. 755–771, 1993.
[184] J. G. Ramírez, "Statistical Intervals: Confidence, Prediction, Enclosure," *SAS Institute Inc.*, white paper, 2009.
[185] P. Drineas, A. Frieze, R. Kannan, S. Vempala and V. Vinay, "Clustering Large Graphs via the Singular Value Decomposition," *Machine Learning*, 56, pp. 9–33, 2004.
[186] L. Xu, "How many clusters?: A YING–YANG machine based theory for a classical open problem in pattern recognition," *in Proc. IEEE Int. Conf. Neural Networks*, vol. 3, pp. 1546–1551, 1996.
[187] Lei Xu, "Bayesian Ying–Yang machine, clustering and number of clusters," *Pattern Recognition Letters*, vol. 18, pp. 1167–1178, 1997.
[188] P. Guo, C.L.P. and M. R. Lyu, "Cluster number selection for a small set of samples using the Bayesian Ying-Yang model," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 757–763, 2002.
[189] Nikos Vlassis and Aristidis Likas, "A Greedy EM Algorithm for Gaussian Mixture Learning," *Neural Processing Letters*, vol. 15, no. 1, pp. 77–87, 2002.
[190] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1344–1348, 2005.
[191] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.
[192] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou, "Robust Sequential Data Modeling Using an Outlier Tolerant Hidden Markov Model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1657–1669, 2009.
[193] S. Har-Peled and B. Sadri, "How fast is the k-means method?," *Algorithmica*, vol. 41, no. 3, pp. 185–202, 2005.
[194] R. Nock and F. Nielsen, "On Weighting Clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1223–1235, 2006.
[195] A. Y. Ng, M. I. Jordan and Y. Weiss, "On spectral clustering: analysis and an algorithm," *Advances in Neural Information Processing*, vol. 14, pp. 849–856, 2002.
[196] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp 395–416, 2007.
[197] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian inference for dynamic models with Dirichlet process mixtures," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 71–84, 2008.
[198] S. Kim, P. Smyth and H. Stern, "A Bayesian Mixture Approach to Modeling Spatial Activation Patterns in Multisite fMRI Data," *IEEE Trans. Med. Imag.*, vol. 29, no. 6, pp. 1260–1274, 2010.
[199] M. Ramoni, P. Sebastiani and P. Cohen, "Bayesian Clustering by Dynamics," *Machine Learning*, vol. 47, no. 1, pp. 91–121, 2001.
[200] R. L. Streit and P. K. Willett, "Detection of Random Transient Signals via Hyperparameter Estimation," *IEEE Trans. Signal Process.*, vol. 47, no. 7, pp. 1823–1834, 1999.
[201] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and navigation*, Wiley and Sons, 2001.
[202] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity Analysis of the Marginalized Particle Filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, 2005.
[203] H. Himberg, Y. Motai, "Head orientation prediction: delta quaternions versus quaternions," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1382–1392, 2009.
[204] M. H. Kim, S. Lee and K. C. Lee, "Kalman Predictive Redundancy System for Fault Tolerance of Safety-Critical Systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 46–53, 2010.
[205] V. P. Jilkov, X. R. Li, "Bayesian estimation of transition probabilities for markovian jump systems by stochastic simulation," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1620–1630, 2004.
[206] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, John Wiley & Sons, 2001.
[207] G. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, 2000.
[208] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp 1–38, 1977.
[209] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and applications to clustering*, Marcel Dekker, 1988.
[210] J. Kleinberg, É. Tardos, *Algorithm Design*, Pearson Education, ch. 2, 2006.

www.manaraa.com

[211] X.-R. Li and b.-S. Yaakov, "Multiple-Model Estimation with Variable Structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493 1996.
[212] X. R. Li, X. R. Zhi and Y. M. Zhang, "Multiple-model estimation with variable structure-part III: Model-Group Switching Algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 1, pp. 225–241, 1999.
[213] http://www.polhemus.com/.
[214] P. Orbanz and Y. W. Teh, "Bayesian Nonparametric Models," *In Encyclopedia of Machine Learning*, Springer, 2010.
[215] K. P. Burnham and D. R. Anderson, *Model selection and multi-model inference: a practical information-theoretic approach*, Springer-Verlag New York, 2002.
[216] J. Tan and N. Kyriakopoulos, "Implementation of a tracking Kalman filter on a digital signal processor," *IEEE Trans. Ind. Electron.*, vol. 35, no. 1, pp 126–134, 1988.
[217] Heui-Wook Kim and Seung-Ki Sul, "A new motor speed estimator using Kalman filter in low-speed range," *IEEE Trans. Ind. Electron.*, vol. 43, no. 4, pp. 498–504, 1996.
[218] B. Terzic and M. Jadric, "Design and implementation of the extended Kalman filter for the speed and rotor position estimation of brushless DC motor," *IEEE Trans. Ind. Electron.*, vol. 48, no. 6, pp. 1065–1073, 2001.
[219] Murat Barut, Seta Bogosyan and Metin Gokasan, "Speed-Sensorless Estimation for Induction Motors Using Extended Kalman Filters," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 272–280, 2007.
[220] S.-h. P. Won, W. W. Melek and F. Golnaraghi, "A Kalman/Particle Filter-Based Position and Orientation Estimation Method Using a Position Sensor/Inertial Measurement Unit Hybrid System," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1787–1798, 2010.
[221] M. Chueh, Y. L. W. Au Yeung, K. -P. C. Lei and S. S. Joshi, "Following Controller for Autonomous Mobile Robots Using Behavioral Cues," *IEEE Trans. Ind. Electron.*, vol. 55, no. 8, pp. 3124–3132, 2008.
[222] Y. Motai and A. Kosaka, "Hand–Eye Calibration Applied to Viewpoint Selection for Robotic Vision," *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3731–3741, 2008.
[223] Won-Sang Ra, Hye-Jin Lee, Jin Bae Park and Tae-Sung Yoon, "Practical Pinch Detection Algorithm for Smart Automotive Power Window Control Systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1376–1384, 2008
[224] K. Szabat and T. Orlowska-Kowalska, "Performance Improvement of Industrial Drives With Mechanical Elasticity Using Nonlinear Adaptive Kalman Filter," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1075–1084, 2008.
[225] A. G. Beccuti, S. Mariethoz, S. Cliquennois, Shu Wang and M. Morari, "Explicit Model Predictive Control of DC–DC Switched-Mode Power Supplies With Extended Kalman Filtering," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1864–1874, 2009.
[226] K. Szabat, T. Orlowska-Kowalska and M. Dybkowski, "Indirect Adaptive Control of Induction Motor Drive System With an Elastic Coupling," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4038–4042, 2009.
[227] C. Mitsantisuk, S. Katsura, K. Ohishi, "Kalman-Filter-Based Sensor Integration of Variable Power Assist Control Based on Human Stiffness Estimation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3897–3905, 2009.
[228] N. Salvatore, A. Caponio, F. Neri, S. Stasi, G. L. Cascella, "Optimization of Delayed-State Kalman-Filter-Based Algorithm via Differential Evolution for Sensorless Control of Induction Motors," *IEEE Trans. Ind. Electron.*, vol. 57, no. 1, pp. 385–394, 2010.
[229] M. Charkhgard, M. Farrokhi, "State of Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF," *IEEE Trans. Ind. Electron.*, vol. 57, no 12, pp. 4178–4187, 2010.
[230] R. J. Williams, "Training recurrent networks using the extended Kalman filter," *Int. Joint Conf. Neural Networks*, vol. 4, pp. 241–246 , 1992.
[231] D. W. Ruck, S. K. Rogers, M. Kabrisky, P. S. Maybeck and M. E. Oxley, "Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 6, pp. 686–691, 1992.
[232] S. Murtuza and S. F. Chorian, "Node decoupled extended Kalman filter based learning algorithm for neural network," *IEEE Int. Symposium on Intelligent Control*, pp. 364–369, 1994.

www.manaraa.com

[233] S. Li, D. C. Wunsch, E. O'Hair and M. G. Giesselmann, "Extended Kalman Filter Training of Neural Networks on a SIMD Parallel Machine," *J. Parallel and Distributed Computing*, vol. 62, no. 4, pp. 544–562, 2002.
[234] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma and Y. Uchikawa, "Trajectory control of robotic manipulators using neural networks," *IEEE Trans. Ind. Electron.*, vol. 38, no. 3, pp. 195–202, 1991.
[235] H. Tai, J. Wang and K. Ashenayi, "A neural network-based tracking control system," *IEEE Trans. Ind. Electron.*, vol. 39, no. 6, pp 504–510, 1992.
[236] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Trans. Ind. Electron.*, vol. 39, no. 6, pp 472–489, 1992.
[237] M. Saad, P. Bigras, L.-A. Dessaint and K. Al-Haddad, "Adaptive robot control using neural networks," *IEEE Trans. Ind. Electron.*, vol. 41, no. 2, pp. 173–181, 1994.
[238] T. W. S. Chow and Yong Fang, "A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics," *IEEE Trans. Ind. Electron.*, vol. 45, no. 1, pp. 151–161, 1998.
[239] P. Payeur, Hoang Le-Huy and C. M. Gosselin, "Trajectory prediction for moving objects using artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 42, no. 2, pp. 147–158, 1995.
[240] Chi-Huang Lu, Ching-Chih Tsai, "Adaptive Predictive Control With Recurrent Neural Network for Industrial Processes: An Application to Temperature Control of a Variable-Frequency Oil-Cooling Machine," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp 1366–1375, 2008.
[241] B. M. Wilamowski, N. J. Cotton, O. Kaynak and G. Dundar, "Computing Gradient Vector and Jacobian Matrix in Arbitrarily Connected Neural Networks," *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3784–3790, 2008.
[242] M. Wlas, Z. Krzemiriski and H. A. Toliyat, "Neural-Network-Based Parameter Estimations of Induction Motors," *IEEE Trans. Ind. Electron.*, vol. 55, no. 4, pp. 1783–1794, 2008.
[243] J. Mazumdar and R. G. Harley, "Recurrent Neural Networks Trained With Backpropagation Through Time Algorithm to Estimate Nonlinear Load Harmonic Currents," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3484–3491, 2008.
[244] S. Cong and Y. Liang, "PID-Like Neural Network Nonlinear Adaptive Control for Uncertain Multivariable Motion Control Systems," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3872–3879, 2009.
[245] T. Orlowska-Kowalska and M. Kaminski, "Effectiveness of Saliency-Based Methods in Optimization of Neural State Estimators of the Drive System with Elastic Couplings," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4043–4051, 2009.
[246] Chow Yin Lai, F. L. Lewis, V. Venkataramanan, Xuemei Ren, Shuzhi Sam Ge and T. Liew, "Disturbance and Friction Compensations in Hard Disk Drives Using Neural Networks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 2, pp. 784–792, 2010.
[247] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems Part I: deterministic non-linear systems," *Int J Control*, vol. 41, no. 2, pp. 303–328, 1985.
[248] S. Chen and S. A. Billings, "Representations of non-linear systems: the NARMAX model," *Int. J. Control*, vol. 49, no. 3, pp. 1013–1032, 1989.
[249] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, 1990.
[250] O. Nerrand, P. Roussel-Ragot, L. Personnaz, and G. Dreyfus, "Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms," *Neural Computation*, vol. 5, no. 2, pp. 165–199, 1993.
[251] F. Ernst, A. Schlaefer, S. Dieterich and A. Schweikard, "A fast lane approach to LMS prediction of respiratory motion signals," *Biomedical Signal Processing and Control* 3, pp 291–299, 2008.
[252] J. H. Goodband, O. C. L. Haas and J. A. Mills, "A comparison of neural network approaches for on-line prediction in IGRT," *Med. Phys.*, vol. 35, no. 3, pp. 1113–1122, 2008.
[253] F. Ernst and A. Schweikard, "Predicting respiratory motion signals for image-guided radiotherapy using multi-step linear methods (MULIN)," *Int. J. CARS* 3, pp. 85–90, 2008.
[254] M. J. Murphy and D. Pokhrel, "Optimization of adaptive neural network to predict breathing," *Med Phys*, vol. 36, no. 1, pp. 40–47, 2009.
[255] Danilo Mandic and Jonathon Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures, and Stability*, John Wiley & Sons, 2001.

www.manaraa.com

[256] F. J. Pineda, "Generalization of backpropagation to recurrent neural networks," *Physical Rev. Lett.*, vol. 59, no. 19, pp. 2229–2232, 1987.
[257] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
[258] B. A. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks: a survey," *IEEE Trans. Neural Netw.*, vol. 6, no. 5, pp. 1212–1228, 1995.
[259] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1 , no 2, pp. 270–280, 1989.
[260] R. J. Williams and D. Zipser, "Experimental Analysis for the Real-time Recurrent Learning Algorithm," *Connection Science*, vol. 1, no. 1, pp. 87–111, 1989.
[261] G. Kechriotis and E. S. Manolakos, "Training fully recurrent neural networks with complex weights," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 3, pp. 235–238, 1994.
[262] G. Chryssolouris, M. Lee and A. Ramsey, "Confidence interval prediction for neural network models," *IEEE Trans. Neural Netw.*, vol. 7, no 1, pp. 229–232, 1996.
[263] J. T. Gene Hwang and A. Adam Ding, "Prediction Intervals for Artificial Neural Networks," *J. American Statistical Association*, vol. 92, no. 438, pp. 748–757, 1997.
[264] Tom Heskes, "Practical Confidence and Prediction Intervals," *Advances in Neural Information Processing Systems* 9, pp. 176–182, 1997.
[265] Durga L. Shrestha and Dimitri P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Networks*, vol. 19, no. 2, pp 225–235, 2006.
[266] David J. Olive, "Prediction intervals for regression models," *Computational Statistics & Data Analysis*, vol. 51, no. 6, pp. 3115–3122, 2007.
[267] A. Khosravi, S. Nahavandi and D. Creighton, "Constructing prediction intervals for neural network metamodels of complex systems," *Int. Joint Conf. Neural Networks*, pp. 1576–1582, 2009.
[268] A. Bhattacharya, C. Chakraborty, "A Shunt Active Power Filter With Enhanced Performance Using ANN-Based Predictive and Adaptive Controllers," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 421–428, 2011.
[269] G. Benchetrit, "Breathing pattern in humans: diversity and individuality," *Respiration Physiology*, vol. 122, no. 2–3, pp. 123–129, 2000.
[270] I. G. Buliev, C. T. Badea, Z. Kolitsi and N. Pallikarakis, "Estimation of the heart respiratory motion with applications for cone beam computed tomography imaging: a simulation study," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 404–411, 2003.
[271] T. Mu, T. C. Pataky, A. H. Findlow, M. S. H. Aung and J. Y. Goulermas, "Automated Nonlinear Feature Generation and Classification of Foot Pressure Lesions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 418–424, 2010.
[272] H. Atoui, J. Fayn and P. Rubel, "A Novel Neural-Network Model for Deriving Standard 12-Lead ECGs From Serial Three-Lead ECGs: Application to Self-Care," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 883–890, 2010.
[273] H. Murshed , H. H. Liu , Z. Liao, J. L. Barker, X. Wang, S. L. Tucker, A. Chandra, T. Guerrero, C. Stevens, J. Y. Chang, M. Jeter, J. D. Cox, R. Komaki and R. Mohan, "Dose and volume reduction for normal lung using intensity-modulated radiotherapy for advanced-stage non-small-cell lung cancer," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 58, no. 4, pp. 1258–1267, 2004.
[274] A. Lanatà, E. P. Scilingo, E. Nardini, G. Loriga, R. Paradiso and D. De-Rossi, "Comparative Evaluation of Susceptibility to Motion Artifact in Different Wearable Systems for Monitoring Respiratory Rate," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 378–386, 2010.
[275] A. Cohen and D. Landsberg, "Analysis and Automatic Classification of Breath Sounds," *IEEE Trans. Biomed. Eng.*, vol. 31, no. 9, pp. 585–590, 1984.
[276] M. J. Baemani, A. Monadjemi, and P. Moallem, "Detection of Respiratory Abnormalities Using Artificial Neural Networks," *J Computer Science*, vol. 4, no. 8, pp. 663–667, 2008.
[277] S. Jafari, H. Arabalibeik and K. Agin, "Classification of normal and abnormal respiration patterns using flow volume curve and neural network," *Int. Symposium on Health Informatics and Bioinformatics*, pp. 110–113, 2010.
[278] I. Ayappa, R. G. Norman, D. Whiting, A. H. W. Tsai, F. Anderson, E. Donnely, D. J. Silberstein, and D. M. Rapoport, "Irregular Respiration as a Marker of Wakefulness during Titration of CPAP," Sleep, vol. 32, no. 1, pp. 99–104, 2009.

www.manaraa.com

[279] P. Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic Classification of Heartbeats Using ECG Morphology and Heartbeat Interval Features," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 7, pp. 1196–1206, 2004.
[280] P. Chazal, and R. B. Reilly, "A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2535–2543, 2006.
[281] M. Hoogeman, JB Prévost, J. Nuyttens, J. Pöll, P. Levendag P, B. Heijmen, "Clinical accuracy of the respiratory tumor tracking system of the cyberknife: assessment by analysis of log files," *Int J Radiat Oncol Biol Phys*., vol. 74, no. 1, pp. 297–303, 2009.
[282] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.
[283] M. A. Kupinski, D. C. Edwards, M. L. Giger, and C. E. Metz, "Ideal Observer Approximation Using Bayesian Classification Neural Networks," *IEEE Trans. Med. Imag.*, vol. 20, no. 9, pp 886–899, 2001.
[284] Z. Li, D. Lin and X. Tang, "Nonparametric Discriminant Analysis for Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 755–761, 2009.
[285] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
[286] C. I. Christodoulou, C. S. Pattichis, M. Pantziaris and A. Nicolaides, "Texture-Based Classification of Atherosclerotic Carotid Plaques," *IEEE Trans. Med. Imag.*, vol. 22, no. 7, pp. 902–912 2003.
[287] W. Chen, C. E. Metz, M. L. Giger, and K. Drukker, "A Novel Hybrid Linear/Nonlinear Classifier for Two-Class Classification: Theory, Algorithm, and Applications," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp 428–441, 2010.
[288] V. Srinivasan, C. Eswaran, and N. Sriraam, "Approximate Entropy-Based Epileptic EEG Detection Using Artificial Neural Networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 3, pp. 288–295, 2007.
[289] A. H. Khandoker, J. Gubbi and M. Palaniswami, "Automated Scoring of Obstructive Sleep Apnea and Hypopnea Events Using Short-Term Electrocardiogram Recordings," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 1057–1067, 2009.
[290] J. Zhao and P. L. H. Yu, "Fast ML Estimation for the Mixture of Factor Analyzers via an ECM Algorithm," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1956–1961, 2008.

www.manaraa.com