USC-SIPI Report #456

Advanced Learning Systems for Highly Uncertain Environments

By
Daoyuan Zhai

May 2012

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
USC Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

ADVANCED LEARNING SYSTEMS FOR HIGHLY UNCERTAIN ENVIRONMENTS

by

Daoyuan Zhai

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

May 2012

Dedication

To my parents, Tiejun Zhai and Xia Duan

and all my family and friends

for their unconditional love and support.

Acknowledgements

First of all, I would like to express my deepest gratitude to my academic advisor and dissertation committee chair, Prof. Jerry M. Mendel, for his guidance and support throughout my entire Ph.D. study. His knowledge, experience and encouragement have led me through many difficulties I encountered during this long and tough process. And without a kind and patient advisor like him, my five years here at USC would be much harder.

And I would also like to thank all my fellow students here in Prof. Mendel's team for our fruitful discussions and friendships, they are Dr. Feilong Liu, Dr. Dongrui Wu, Minshen Hao, Mohammad Reza Rajati, and Mohammad Mehdi Korjani. I'd especially like to thank Dr. Dongrui Wu for introducing me to Prof. Mendel and his team.

I also want to extend my deepest thanks to the entire staff at the Center for Interactive Smart Oilfield Technologies (CiSoft) at USC, especially Director Dr. Iraj Ershaghi. Their hard work have created a stable source of financial and intellectual support for our projects, which allows me to whole-heartedly focus on the research issues in a worry-free condition.

I am also grateful to all my qualification and defense committee members: Prof. Iraj Ershaghi, Prof. Antonio Ortega, Prof. Keith Jenkins, and Prof. Jay C.-C. Kuo. I want to thank them for sharing their valuable time with me and making a lot of insightful suggestions.

Last but not least, my sincerest gratitude goes to all my family and friends. In particular, I want to thank my uncle, Dr. Michael Chang. If it wasn't due to his support, I probably wouldn't be able to come to the United States to pursue my dream.

I will always remember and be grateful.

Table of Contents

List of Tables

List of Figures

Abstract

Data quality, which broadly refers to both data frequency and accuracy, has been the key factor that determines the performances of most of today's machine learning systems. However, making frequent and accurate measurements for certain applications can be extremely costly, or simply impossible in some cases.

Given a seriously limited number of measurements that are subject to heavy noises, the efforts to ensure the reliability of the outcomes can usually be spent on both the data side and the system side. More specifically, one can try to interpolate the measurements that one fails to collect, and, at the same time, find ways to suppress noises in the actual measurements; on the other hand, one can also design the system in such a way that it becomes more robust even under a great amount of uncertainties.

To this end, the work presented in this dissertation is aimed at addressing this issue from both of these two perspectives, separately. Using the subsurface flow trend detection problem, an increasingly popular topic in the domain of petroleum engineering in recent years, as our background application, the first part of the dissertation introduces a novel iterated state-estimation-based interpolation approach that fills in virtual measurements for an extremely sparsely sampled dataset, and demonstrates that, with the help of these interpolated measurements, the original extended Kalman filter technique, that was developed for this estimation problem, could obtain significantly improved results, almost as if the full set of measurements were given.

Then, from a system design point of view, the second part of this dissertation introduces a paradigm for building certain types of advanced fuzzy logic systems, e.g., non-singleton interval type-2 fuzzy logic systems that operate in extremely uncertain environments. Such systems have not been widely studied before due to their very complex analytical forms that make conventional gradient-based learning processes impractical; however, with the introduction and advancement of population-based random optimization techniques, such as the one this dissertation mainly focuses on, quantum-based particle swarm, the learning process of these systems can be significantly simplified and made more efficient. This dissertation uses a universal image noise removal problem and a forest-fire-size prediction problem as examples to demonstrate general design frameworks. Under such frameworks, these systems can be easily modified and applied to other applications.

The third and final part of this dissertation presents theoretical studies that were performed for the most advanced fuzzy logic sets that are currently under research — general type-2 fuzzy sets. Our results include 1) a unified theorem that demonstrates how previously uncomputable uncertainty measures for such type-2 fuzzy sets, centroid, cardinality, fuzziness, variance, and skewness, can be obtained through a novel α-plane representation theorem; and, 2) a fast *centroid-flow algorithm* for computing the centroid of a general type-2 fuzzy set.

Chapter 1 Introduction

A widely quoted definition of machine learning, an increasingly popular and challenging branch of artificial intelligence, is provided by Mitchell [132] as follows: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". To put it more understandably, it is a scientific discipline that studies past examples (training data that are collected through sensors or database systems) to capture characteristics of their complex underlying patterns or probability distribution, which enable computers to evolve behaviors that make intelligent decisions in favor of certain evaluating criteria.

Since it is the purpose of machine learning systems to reveal hidden relations between observable variables, it is obvious that data quality is of key importance to system performance, just as the phrase "garbage in, garbage out" tries to draw attention to the fact that even the most intelligently designed algorithm will still unquestioningly process the most nonsensical of input data and produce nonsensical output. Among a wide range of applications that machine learning algorithms focus on solving, such as bioinformatics [144], computational finances [131] and natural language processing [83], there are those applications, for which data are fully collectable and easily accessible, e.g., when developing a movie recommendation system [225], a massive amount of user ratings can be freely downloaded from the online movie database (www.imdb.com).

Nevertheless, there are many other applications where easy data is not the case, namely, it could be that collecting enough data is extremely costly or even impractical, e.g., measuring individual production rate of an oil producer well in an onshore oilfield that has aging infrastructure; or, it could be that, even though a full set of data is given, it suffers from overwhelming uncertainties, e.g., an image that is badly distorted during acquisition stage and is then passed through a highly noisy channel; or, it could be a combination of both.

This dissertation focuses on developing machine learning systems that are suited for working under the antagonistic situations described above. The study philosophy employed herein has two separate pieces: 1) Reconstruct the missing measurements to create a virtually full set of data such that the original learning algorithm, which was designed to perform well only when a full set of data is given, is able to regain equivalently good performance; and, 2) Introduce a paradigm that guides the development of advanced fuzzy logic learning algorithms that are far more robust to heavy noise, but was previously understudied due to designing difficulties. The dissertation also presents several important theoretical results that were obtained for the most advanced type of fuzzy set that is currently under study, type-2 fuzzy set. These results will become key foundations for other researchers who have been working on developing type-2 fuzzy logic systems, which are theoretically the most capable members in the fuzzy logic family for handling uncertainties.

This dissertation presents these research outcomes is mainly by means of application examples. Therefore, the rest of this chapter briefly goes through the background motivation and objectives of three applications, through which the different algorithms for our learning systems can be studied and examined.

1.1 Signal Processing in Waterflood Management

Currently (as of the 4th quarter, 2011), crude oil is the most important form of energy our world relies on. An International Energy Agency (IEA) study shows that the world's current oil demand amounts to approximately 88.3 million barrels per day (mb/d) (see Fig. 1.1). This number has risen about 2% compared to 2010 and will continue to grow in 2012,

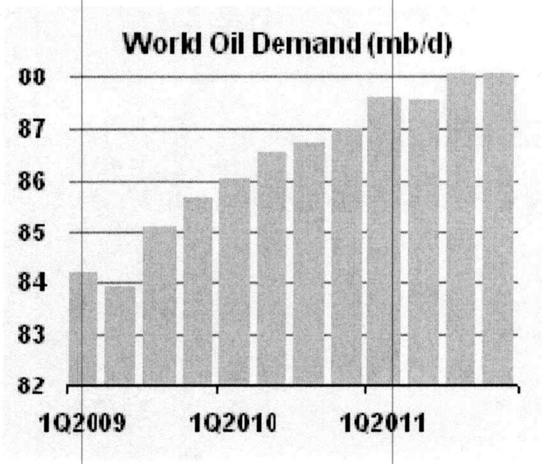


Figure 1.1 World Oil Demand from 2009 to 2011.

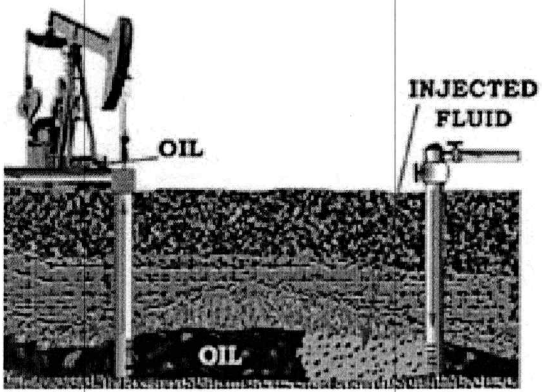


Figure 1.2 Mechanism of Secondary Recovery Stage

due to improving economic conditions in Europe and developing countries. Also, it is steadily becoming more and more difficult to produce crude oil from existing oilfields and to find new oilfields to fill in the increasing gap between demands and actual productions. Thus, our ability to perform oil extraction and recovery more efficiently and economically becomes more and more critical.

The oil extraction and recovery of a field is typically divided into three stages. The first of which is called the *Primary Recovery Stage* (PRS). In this stage, reservoir drive comes from a number of natural mechanisms, e.g., natural water displacing oil downward into the well, expansion of the natural gas at the top of the reservoir, expansion of gas initially dissolved in the crude oil, gravity drainage resulting from the movement of oil within the reservoir from the upper to the lower parts where the wells are located, etc. The *Recovery Factor* (RF) of this stage is generally 5-15%, which means more than 80% of the *Original Oil In Place* (OOIP) is left behind.

Over the course of PRS, the natural underground pressure gradually decreases and eventually becomes insufficient to push oil to the surface. This leads to a *Secondary Recovery Stage* (SRS). The methods applied during this stage rely on supplying external forces into the reservoir in the form of injecting fluids to increase reservoir pressure, hence increasing or even replacing the natural reservoir drive with an artificial drive. Sometimes, pumps, such as beam pumps and electrical submersible pumps, are used. Other techniques include water injection, natural gas reinjection and gas lift (See Fig. 1.2). Typical RF from water-flood operations in this stage is about 30%, depending on the properties of oil and the characteristics of the reservoir rock. On average, the RF after the PRS and SRS is between 30% and 50%.

Eventually, a *Tertiary Recovery Stage* (TRS) is launched when the SRS can no longer provide enough supply for adequate production, but only when the oil can still be recovered profitably. Such a decision is determined by the cost of the extraction method and the current price of crude oil. The techniques of TRS, which includes steam injection, carbon dioxide flooding, etc., try to reduce the oil viscosity, mobilize the oil and make oil extraction easier. During this stage, another 5% to 15% of the reservoir oil is recovered.

Throughout this complex multi-stage process, SRS is where the most significant number of RF can be achieved; and, waterflood management is the most widely employed solution for SRS [175], because water is the most easily accessible and economical injectant compared to steam, carbon dioxide, hydrocarbon and other chemicals.

The success of a waterflood operation is generally decided by an understanding of inter-well connectivities, namely, one's ability to detect preferential subsurface flow trends, between water injection wells (injectors) and oil production wells (producers). Such knowledge can be obtained by studying the reservoirs modeled by commercial simulators [24, 54, 212]. However, in order to simulate a reservoir that correctly reflects the real world situation, knowledge of various reservoir characterizations, such as fluid properties, capillary pressures, fluid contacts, porosities and so on is required, but often it is not easy to measure them as they are subject to a significant amount of uncertainty. Most of all, this simulator approach is very expensive and time-consuming, and, therefore, is only specifically used for large investment decisions.

Tracer testing [1, 2, 5, 20, 137] is another approach to map high permeability channels during waterfloods, and to help estimate many useful reservoir features, such as reservoir swept volume, fluid velocities, and flow geometry. However, carrying out frequent tracer tests is also expensive and time-costly. And, above all, tracer tests cannot provide us with a dynamic view of the system.

In recent years, applying signal-processing-based techniques that use the most commonly available data in an oilfield (injection and production rates) has become increasingly popular in waterflood management because they are relatively inexpensive. Many researchers have made contributions in this area [6, 7, 84, 102, 110, 140, 145, 151, 158, 168, 207, 228, 230], e.g., Albertoni and Lake [6] modeled the production rate as a weighted sum of delayed injection rates, Lee et al. [102] proposed to characterize the reservoir between multiple producers and injectors as a standard linear Finite Impulse Response-Multiple Input Multiple Output system, Liu et al. [110] and Zhai et al. [228, 230] introduced an Extended Kalman Filter (EKF) [73, 121] strategy based on a two-parameter Auto-Regressive (AR) model, Yousef et al. [207] developed a Multiple Linear Regression method based on a Capacitance Model, and Sayarpour et al. [158] improved this method by introducing a Capacitance-Resistance Model.

The application prototype that this dissertation uses to introduce the novel data interpolation technique developed to enhance learning system performance is to solve the waterflood management problem via the Extended Kalman Filter (EKF) [110]. Our EKF uses a relatively simple two-parameter model to characterize the reservoir as a continuous impulse response that converts input signals (injection rates) to output signals (production rates). If N injectors are assumed to influence a given producer, this model has exactly $2N$ parameters. Such a producer-centric model is used to construct an equivalent State-Variable-Model (SVM) which can then be used by an EKF to directly estimate and keep track of the Injector-Producer-Relationship (IPR) values.

For completeness, this dissertation reviews how one can modify and extend Liu and Mendel's work to resolve several practical issues that emerged when their original technique was applied to process real field data, namely: 1) Some of the estimated IPR values turned out to be negative; 2) There exists no explicit guideline for selecting possibly contributing injectors for a producer-centric model; and, 3) The sum of the IPR values between an injector and all of its influenced producers can sometimes be greater than one, which is physically impossible, because this implies that the injector is contributing more water than the total amount it has injected.

As mentioned at the beginning of this chapter, the performance of all signal-processing-based approaches to detect IPR values are dependent on the quality and frequency of the measured injection and production rates. Unfortunately, in practice, the production rates (well-test data) are only measured once every eight to ten days on average, which is significantly lower than the measuring frequency required by EKF approaches to provide reliable outcomes. This is where a reliable interpolation method comes into play, to help accurately reconstruct missing production rates, and thus allow the system to avoid certain highly negative impacts that such under-sampled data could bring about.

We have developed a reliable interpolation technique called an Iterative Extended Kalman Filter and Smoother (IEKFS) that dynamically interpolates missing data between two available measurements k_1 and k_2. The missing production rate at each time point is modeled as a state variable that is forward-estimated from k_1 to k_2 by the EKF (using measurements up to k_1), and is then backward-estimated by the *Extended Kalman Smoother* (*EKS*) (using the measurement at k_2). This process is carried out iteratively back and forth between k_1 and k_2 until accurate estimates of missing data are recovered. It will be shown, by using reservoir-simulator data, and assuming one measurement every eight days, that the IEKFS interpolated production rates lead to IPR estimates that are almost equivalent to those obtained when production rates are measured daily.

1.2 Universal Image Noise Removal

In this section, we focus on designing a novel filter to reduce noises introduced into digital images during acquisition and/or transmission stages, that are adequately modeled by either Additive Gaussian White Noise (AGWN), impulse noise, or Mixed Gaussian and Impulse Noise (MGIN) [146, 149].

AWGN, which is inadvertently introduced into an image during its acquisition stage, can be modeled as adding to each image pixel a value from a zero-mean Gaussian distribution. An ideal filter for removing AWGN would be able to smooth pixels within a distinct local region of an image without reducing the sharpness of the edges of that region. A Gaussian filter, which is a linear filter, can smooth noise out very efficiently; but, it does this at the price of significant edge blurring. To overcome this drawback, some nonlinear filters have been proposed [143, 176] that focus on using local measures of an image to detect the edges and smooth them less than other parts of the image.

Impulse noise, generally caused by transmission errors, can be modeled by randomly replacing a portion of the pixels with random pixels, while leaving the remaining pixels unchanged. The filters specifically developed for AWGN removal do not work well for impulse noise, because these filters consider the impulse noise pixels as edges, and preserve them. Different kinds of filters that aim at removing impulse noise have therefore been proposed, and have been summarized by Yildirim et al. [204] as follows: 1) standard median filter [59, 177], which replaces the center pixel of a filtering window with the median value of all pixels in that window, has good performance in terms of noise removal, but it also blurs image details like thin lines even at low noise levels; 2) modified versions of the median filter, e.g., weighted and center-weighted median filters [96, 205, 206], which give more weights to certain pixels in the filtering window, gain improved performance in terms of preserving image details at the cost of reduced noise removal capability; 3) impulse detectors, which aim at deciding whether the center pixel of the filtering window has been corrupted by noise or not, e.g., [4, 15, 29, 30, 33, 34, 35, 48, 53, 61, 90, 148, 159, 166, 172, 182, 201, 208, 211, 231]; 4) other mean-filter-based approaches [3, 71, 113] that exhibited good performance, but at the price of high computational expenses; and, 5) many different approaches based on soft computing methodologies [14, 36, 133, 154, 156, 157, 179, 204, 210], along with a number of nonlinear filters [99, 150, 155, 167, 184, 196, 209] that integrate the desired features of some of the aforementioned filters.

MGIN often occurs when an already corrupted image is transmitted over a noisy communication channel [61, 146], and, is modeled by first adding AWGN and then impulse noise into the images. The study of MGIN removal is considered to be more general and challenging than that of pure AWGN or impulse noise removal; therefore, relatively fewer works have been published in this area, e.g., 1) Ma et al. developed a Structure-Adaptive Hybrid Vector Filter (SAHVF) [115] that employs a quad-tree decomposition to assess activities in different regions of the image, in order to choose a noise removal filter among a set of *ad hoc* approaches such as Peer Group Filter (PGF) [89], Adaptive Nearest Neighbor Filter (ANNF) [147], and Structure Weighted Average Filter (SWAF); 2) Garnett et al. [61]

introduced a Rank Order Absolute Differences (ROAD) method that first focuses on impulse detection, and then uses a bilateral Gaussian filter to suppress AWGN; 3) Yang and Wu [203] first use an Impulse Proportion Adaptive Median Filter (IPAMF) to remove impulse noise, and then a Block-Matching 3-Dimensional Discrete Cosine Transformation (BM3D DCT) filter [50] to clean up AWGN.

Two types of criteria are often used to evaluate the performances of noise filters: 1) quantitative measures, such as Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), and 2) visual quality of the restored image. It is generally very difficult for a filter to produce both quantitatively excellent and visually satisfactory outputs; however, it has been observed, when focusing on impulse noise removal only, that the method of Yildirim et al. [204], which is based on a Singleton Interval Type-2 (S-IT2) TSK Fuzzy Logic System (FLS) [122], significantly outperforms all other filters in terms of quantitative measures, and, at the same time, provides very satisfactory visual results.

We are motivated to expand and enhance the idea of Yildirim et al. [204] to achieve a more robust universal MGIN filter that outperforms all other methods both quantitatively and visually. More specifically, based on the framework of Yildirim et al. [204], in this dissertation, we develop a Non-Singleton IT2 (NS-IT2) FLS [122] using a Quantum-behaved Particle Swarm Optimization (QPSO) algorithm, because: (1) When compared to S-IT2 FLSs, NS-IT2 FLSs takes into consideration the uncertainties associated with the input measurements; and, (2) a NS-IT2 FLS can be trained so that it simultaneously accounts for both impulse noise and AWGN, instead of just impulse noise, as in the case of Yildirim et al.. We integrate this NS-IT2 FLS with two other supplementary components, a BM3D DCT filter [50] and a contrast scaling filter [64], both of which further strengthen the system's performance. We will show that this systematic approach enables highly corrupted images in a MGIN environment to achieve much better visual and quantitative results than any other existing method.

1.3 Forest-Fire-Size Prediction

This next application is about predicting forest-fire-size. A forest fire differs from other fires by its extensive size, the speed at which it can spread out from its original source, its potential to change direction unexpectedly, and its ability to jump gaps such as roads, rivers and fire breaks, which altogether make its prevention, detection and suppression very difficult. Most forest fires have significant negative impacts on the ecosystem and human society, causing losses of forest, lives and infrastructures; and, despite increasing human efforts, large amount of forests still burn down each year worldwide, e.g., the statistics of forest fires in the United States from 1960 to 2009 are depicted in Fig. 1.3.

Early detection, which is essential to successful fire suppression, can be based on automatic data collected by satellites and infrared/smoke scanners. Many different data-mining techniques have been proposed to analyze these data and extract expert knowledge for fire/smoke detection [76, 171, 178] and false alarm reduction [3]; however, there are high operational and maintenance costs and localization delays associated with satellites and scanners.

On the other hand, spatial and meteorological data provided by local weather stations are often real-time, economical and constantly available; therefore, researchers have started to employ such data to construct expert systems for fire size prediction [41, 58]. An accurate

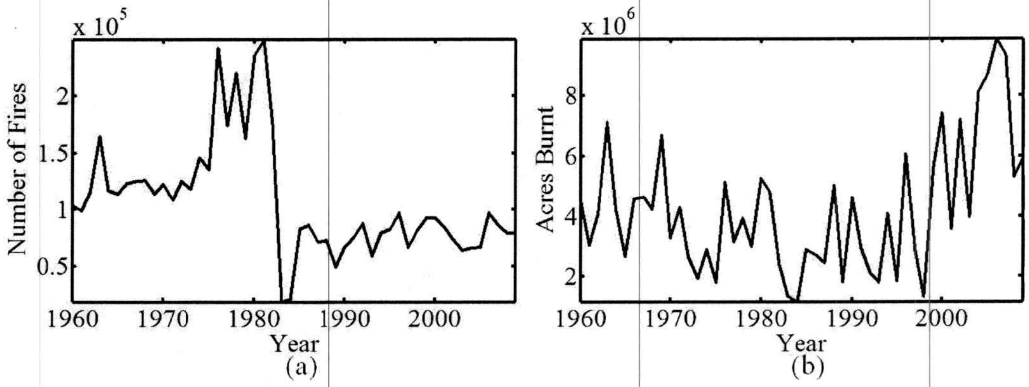


Figure 1.3 Statistics of forest fires in the United States from 1960 to 2009: (a) Total number of fires each year; (b) Acres of forests burnt each year.

prediction of burnt forest areas can enable fire-fighting agencies to appropriately allocate and deploy resources and man-power, which will help avoid wasteful overreaction to relatively small fires and under-reaction to large ones.

Cortez and Morais [41] made the first attempt to study the performances of different data-mining techniques designed for forest-fire-size prediction. Their comparative tests included a *Naïve Average Predictor* that always predicts the fire size to be the average value of the fire sizes in the training data set, a *Naïve Zero Predictor* that always predicts the fire size to be zero, a *Multiple Regression* approach [31], a *Decision Trees* approach [153], a *Random Forests* approach [12], a *Neural Network* approach [16], and a *Support Vector Machine* (SVM) approach [69]. They used two different measures for evaluation purposes, namely, *Root Mean Squared Error* (RMSE) and *Mean Absolute Deviation* (MAD).

Based on 30 Monte-Carlo simulations each with a ten-fold cross-validation (a total number of 300 tests), their average results showed that the only case where a non-Naïve approach could actually outperform the Naïve Zero or Average Predictor (only by a very arrow margin) is when the SVM was used, but then only with respect to MAD. Under other circumstances, either Naïve Zero or Average Predictor works much better.

Comment: It has long been a debateable issue [60] as to whether such Naïve methods should be used in comparative studies; because, even though they may produce smaller errors in terms of measures like RMSE and MAD, the *mutual information* [47] between their estimated and the actual measurements is always zero, which means that absolutely no information regarding the actual measurements can be inferred from these methods. Because other authors have used Naïve methods as benchmark techniques, we will also include them in this chapter.■

Recently, Fowler et al. [58] proposed an *Evolved Type-1* (T1) *FLS* which significantly improved prediction performance in terms of RMSE measure; however, in terms of MAD measure, this approach produced the second worst results among all techniques.

In nature, forest fires occur much more frequently than people normally think [see Fig. 1(a)]. The sizes of over 99% of the fires are distributed within a certain range, and we call such fires *regular events* (we don't mean to minimize the significance of these fires by calling them regular events; their sizes could still be very extensive; it is only a means to distinguish them from the concept of *extreme events* defined later.). The news media usually only cover less than 1% of the fires that are the most devastating and disastrous, and we call such fires *extreme events*. Note that not only are the extreme events greatly different from all regular events, they are also very far away from each other. In other words, the extreme events are also quite unique.

For this application, we propose a *Mixture FLS* that distinguishes between the two kinds of events, and uses them separately to obtain two sets of rules: 1) a fixed number of rules modeled by S-IT2 FSs, and tuned also by the QPSO algorithm using an adequate amount of regular events; and 2) rules modeled by T1 FSs, where one rule is specifically designed for each extreme event.

1.4 Computing Uncertainty Measures for Type-2 Fuzzy Sets

Previously, five uncertainty measures have previously been defined for Interval Type-2 Fuzzy Sets (IT2 FSs), namely centroid, cardinality, fuzziness, variance and skewness. Based on a recently developed α-plane representation for a general T2 FS, this thesis generalize these definitions to such T2 FSs and, more importantly, derive a unified strategy for computing all different uncertainty measures with low complexity. The uncertainty measures of T2 FSs with different shaped *Footprints of Uncertainty* and different kinds of secondary membership functions (MFs) are computed and are given as examples. Observations and summaries are made for these examples, and a *Summary Interval Uncertainty Measure* for a general T2 FS is proposed to simplify the interpretations. Comparative studies of uncertainty measures for Quasi-Type-2 (QT2), IT2 and T2 FSs are also performed to examine the feasibility of approximating T2 FSs using QT2 or even IT2 FSs.

Since centroid is probably the most important kind of uncertainty measures in practice, this thesis also studies ways to boost its computation efficiency. Note that Karnik-Mendel (KM) or the Enhanced Karnik-Mendel (EKM) algorithms are used for computing the centroid of each α-plane, whose iterative features can be time-consuming, especially when the algorithms have to be repeated for many α - planes. So we proposes a new method named *Centroid Flow* (*CF*) *algorithm* to compute the centroid of $\tilde{A}$ without having to apply KM/EKM algorithms for every α – plane. Extensive simulations have shown that the CF algorithm can reduce the computation time by 75% to 80% and 50% to 75% compared to KM and EKM algorithms, respectively, and still maintains satisfactory computation accuracy for various T2 FSs when the primary variable x and α - plane are discretized finely enough.

1.5 Thesis Outline

This dissertation is organized as follows: Chapter 2 introduces the parametric models that can be used to characterize the subsurface reservoir; reviews the general EKF strategy developed to estimate numerical IPR values; extends the technique by incorporating linear constraints, as done by Simon et al. [163, 164, 165] and Yang and Blasch [202], which, for the first time, allows all of the producer-centric models to be grouped together; demonstrates, by using simulation data, the negative impacts of not having data sampled frequently enough; and, finally, presents the technical details of the IEKFS interpolation algorithm and its power in improving the accuracy of the estimated IPR values.

Chapter 3 presents an overview of the NS-IT2 FLS, the detailed structure of its building blocks, namely, *neurofuzzy filters*, and how the training image set can be generated; introduces the QPSO algorithm that can be used to search for the "optimal" set of design parameters for each neurofuzzy filter; briefly reviews the BM3D DCT and contrast scaling filters, that bring about additional enhancements to the overall performance of the NS-IT2 system; and, demonstrates the improved performance of our proposed method by comparing it against several other techniques as well as its T1 and singleton IT2 counterparts.

Chapter 4 analyzes the benchmark forest-fire data set and introduces a numerical method to select data elements as system inputs; introduces the Mixture FLS and the launching criteria of the two sets of rules; and, presents the results of comparative simulations.

Chapter 5 provides necessary background material about general T2 FSs; defines uncertainty measures for a general T2 FS; shows how they can be computed using the α-Plane Representation Theorem (*RT*); provides a summary of existing algorithms for computing uncertainty measures for an IT2 FS (they are needed to compute the uncertainty measures for a general T2 FS); gives examples of uncertainty measures for general T2 FSs for different footprints of uncertainty (*FOU*s) and secondary MFs; provides quantitative interpretations and summaries for the results; and performs comparative studies of uncertainty measures for Quasi-Type-2 (QT2), IT2 and general T2 FSs.

Chapter 6 reviews the KM algorithms and their important properties; illustrates how to use structures of the secondary MFs to approximate neighboring α-planes based on the current α-plane; states the Centroid-Flow (*CF*) algorithm; explains a technical subtlety that allows the algorithm to be programmed more efficiently; and, examines the performance of CF algorithms as compared to KM (EKM) algorithms both in terms of accuracy and computation time.

Chapter 7 draws conclusions and discusses possible future research topics.

Chapter 2 Robust Interpolation based on an Iterated Extended Kalman Filter and Smoother

This chapter focuses on an interwell connectivity detection system, for which our advanced interpolation technique is developed. We first review the parametric models that can be used to characterize the subsurface reservoir, the general strategy to estimate the numerical interwell connectivity values in practice, and certain advanced extensions of the technique. We then go on to present the technical details of the interpolation algorithm, and demonstrate its power in improving the accuracy of our system.

2.1 Introduction

Albertoni and Lake [6] and Youself et al. [207] modeled the reservoir as a system that is characterized by a continuous impulse response and converts the input signal (i.e., injection rates) into an output signal (i.e., production rates). Specifically, Albertoni and Lake modeled the reservoir by a Resistive Model (RM) characterized by a collection of weights, whereas Yousef et al. modeled the reservoir by a Capacitance Model (CM) characterized by two parameters, λ and τ, which quantify the interwell connectivity and dissipation, respectively. In their methods, the Injector-Producer-Relationship (IPR) values are quantitatively measured in terms of the model parameters, namely, weights in the RM and τ in the CM. The parameters of these two models are estimated by the same method, Multiple Linear Regression (MLR). There are certain limitations associated with these two approaches: 1) the parameters are assumed to be stationary during the estimation process, i.e., constant over the window during which the data are analyzed; therefore, whenever changes occur in the IPR values, the analysis has to be repeated for the new situation, which may not be very practical, because the reservoir is dynamic and no change-detection has been developed for these techniques; 2) the CM model is relatively complex; and although it is only characterized by two parameters, one has to determine the primary production and bottom hole pressure impact in order to use the model.

To overcome these difficulties, Liu and Mendel [110] propose to use a two parameter Auto-Regressive (AR) model (also referred to as Liu-Mendel Model in this dissertation) to characterize the impulse response between a single injector and a single producer, and the area under the impulse response can be viewed as a measure of the IPR. This nonlinear model can be analyzed by an Extended Kalman Filter to dynamically estimate the IPR values.

However, a problem emerged as the EKF was tested using real oilfield data, that is, some estimated IPRs become negative during the estimation process. This is largely due to the overwhelming noises in the measured well rates. Although this problem can be resolved, in a relatively straightforward manner, by applying a modified SVM to estimate the square-root of the IPR values so that their squared values are guaranteed to be non-negative, there exists another problem that does not have such an easy fix: Some injector-centric sums [Injector-centric sum (ICS) is the sum of IPRs between an injector and all of its affected producers, whereas producer-centric sum (PCS) is the sum of IPRs between a producer and all of its contributing injectors] of the estimated IPRs can become greater than one, which means such injectors appear to be contributing more water to producers than the amount injected, which is physically impossible. It turns out that both problems can be addressed simultaneously if constraints can be imposed during the EKF processing.

Simon, et al [163-165] and Yang and Blasch [202] have shown how to incorporate linear and nonlinear constraints into the structure of a Kalman filter. Based on their works, this dissertation shows how to develop another modified SVM that allows linear inequality constraints to be imposed, and then generalizes the linearly constrained Kalman filter to a linearly constrained EKF using this new SVM.

It is intuitive that the performances of such signal-processing-based techniques are dependent on the quality and frequency of the measured injection and production rates. However, in practice, the production rates (well-test data) are only measured once every eight to ten days on average; this is significantly lower than the measuring frequency required by these approaches to provide reliable IPR estimations. In this chapter, we first use hypothetical data to show that, *if we directly use such low frequency data, it will take a very long time (usually two to three years) for our approach to reach correct results.* In real field, the simplest and often-used interpolation (data conditioning) method is the Zero-Order Hold approach, namely, to carry on the previously available measurement all the way until a new measurement becomes available. Unfortunately, it also can be shown, using the hypothetical data, that *the production rates interpolated by zero-order hold method will cause all the estimated IPR values to converge to one same average value*. To avoid these highly negative impacts, a more reliable interpolation method that can help us accurately reconstruct missing production rates is required.

As the core element of this chapter, we propose an Iterated Extended Kalman Filter and Smoother (IEKFS) approach to dynamically interpolate the missing data between two available measurements. The missing production rate at each time point is modeled as a state variable; and, it is forward-estimated by the EKF based on previously available measurements, and is also backward-estimated by the Extended Kalman Smoother (EKS) based on the available measurements after it; this process is carried out iteratively until good quality data has been obtained. We will show that, *assuming only one measurement every eight to ten days, the production rates interpolated by the IEKFS approach allow our technique to obtain almost equivalent results as if we were having one measurement every day*. Such interpolated data can also be utilized by other signal-processing-based techniques and help enhance their performances.

2.2 Liu-Mendel Model and an Extended Kalman Filter

2.2.1 Reservoir Model

In this dissertation, the reservoir is modeled as a collection of continuous-time impulse responses that convert injection rates into a production rate. A producer-centric reservoir model with one producer and N independently contributing injectors and is depicted in Fig. 2.1, where $i_1(t)$, $i_2(t)$,..., $i_N(t)$, $n_1(t)$, $n_2(t)$,..., $n_N(t)$ and $i_{m,1}(t)$, $i_{m,2}(t)$,..., $i_{m,N}(t)$ are the actual injection rates that flow into the reservoir, the corresponding injection rate measurement noise, and the measured injection rates, respectively; $p(t)$, $n_p(t)$ and $p_m(t)$ are the actual production rate, the corresponding production rate measurement noise, and the measured production rate, respectively; $p_j^c(t)$ (j = 1,..., N) represents the amount of production rate in $p(t)$ caused by the j^{th} injector ; and, $f(r_j,k_j)$ are scale functions which decides how much of the injection rate of the j^{th} injector flows towards the producer, where $f(r_j,k_j)$ can be viewed as a linear or non-linear scalar function of the distance, r_j, and the

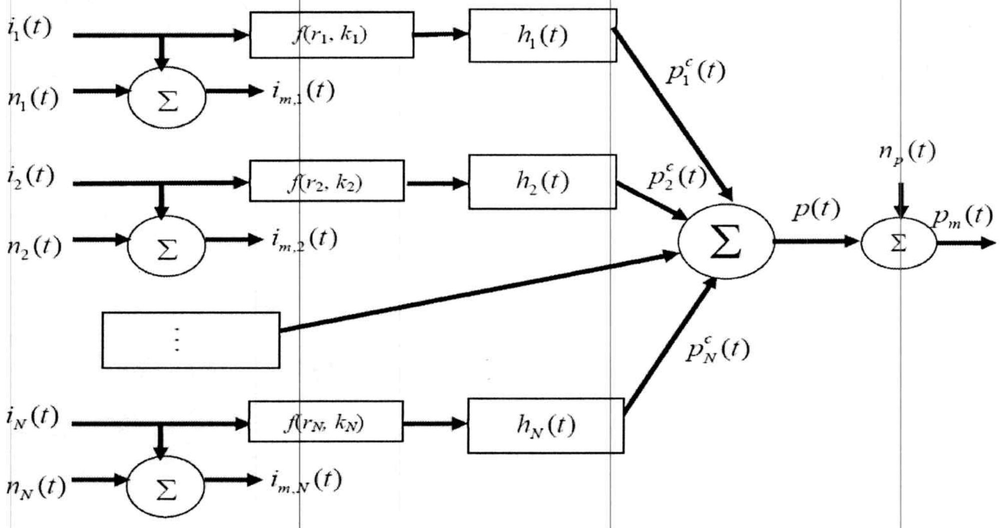


Figure 2.1 The Reservoir Model for a single producer and its N contributing injectors.

permeability, k_j, between the j^{th} injector and the producer. Because noise-free data, $i_1(t)$, $i_2(t)$,…, $i_N(t)$ and $p(t)$, are not directly available, we can only use their measured values, $i_{m,1}(t)$, $i_{m,2}(t)$,…, $i_{m,N}(t)$ and $p_m(t)$, for our data processing.

2.2.2 Subsystem Injector-Producer Model

The two parameter Auto-Regressive (AR) model that characterizes the impulse response between a single producer and a single injector has the following expression:

$$h(t) = bte^{-at} \tag{2-1}$$

Because only sampled injection and production rates are available for processing, a discrete-time version of (2-1) is used; its Z-transform is:

$$H(z)=\frac{\gamma z^{-1}}{(1-\alpha z^{-1})^2} \tag{2-2}$$

where $\alpha = e^{-\alpha T}$ and $\gamma = b\alpha T$ are two parameters that determine the model and T is the sampling period. The total production rate at a specific producer is the sum of the individual production rates contributed by injectors that influence this producer, namely (assuming there are N contributing injectors):

$$P(z)=\sum_{j=1}^{N} P_j^c(z)=\sum_{j=1}^{N} H_j(z)f(r_j,k_j)I_j(z)=\sum_{j=1}^{N}\frac{\gamma_j f(r_j,k_j)z^{-1}}{(1-\alpha_j z^{-1})^2}I_j(z) \tag{2-3}$$

where $P(z)$, $P_j^c(z)$ and $I_j(z)$ are the Z transforms of the total production rate, $p(k)$, the production rate contributed by the j^{th} injector, $p_j^c(k)$, and the injection rate of the j^{th} contributing injector, $i_j(k)$, respectively.

From (2-2), the subsystem between the producer and its j^{th} contributing injector can be modeled as ($j = 1, 2, \ldots, N$):

$$(1-\alpha_j z^{-1})^2 P_j^c(z)=\gamma_j f(r_j,k_j)z^{-1}I_j(z) \tag{2-4}$$

which, when transformed back into time domain, is:

$$p_j^c(k+1)-2\alpha_j p_j^c(k)+\alpha_j{}^2 p_j^c(k-1)=\gamma_j f(r_j,k_j)i_j(k) \tag{2-5}$$

Liu and Mendel [110] defined the numerical IPR value as the area under the sampled impulse response of each injector-producer pair, and showed that it can be calculated as ($j = 1, 2, \ldots, N$):

$$IPR_j = \frac{\gamma_j f(r_j, k_j)}{(1-\alpha_j)^2} \tag{2-6}$$

Observe that three parameters—α_j, γ_j and $f(r_j,k_j)$—need to be estimated to compute this IPR_j; however, because γ_j and $f(r_j,k_j)$ are multiplicative in (2-5) and (2-6), the number of parameters can be reduced from three to two by letting $\gamma_j' = \gamma_j f(r_j, k_j)$. Note, also, that only the measured injection rates, $i_{m,j}(k) = i_j(k) + n_j(k)$, are available for processing; hence, (2-5) and (2-6) can be re-expressed, as:

$$\begin{aligned} p_j^c(k+1) - 2\alpha_j p_j^c(k) + \alpha_j^2 p_j^c(k-1) &= \gamma_j'(i_{m,j}(k) - n_j(k)) \\ &= \gamma_j' i_{m,j}(k) + n_{p_j^c}(k) \end{aligned} \tag{2-7}$$

$$IPR_j = \frac{\gamma_j'}{(1-\alpha_j)^2} \tag{2-8}$$

Note that in (2-7), we set $n_{p_j^c}(k) = \gamma_j' n_j(k)$ so that we will be able to use $i_{m,j}(k)$ instead of $i_j(k)$ as our input, and to use $n_{p_j^c}(k)$ as the standard additive noise.

2.2.3 Original State Variable Model

To establish an Extended Kalman Filter and Smoother, the first step is to construct a State Variable Model (SVM) [73, 121] based on the second-order finite-difference equation (2.7). Liu and Mendel [110] chose to use $\alpha_j(k)$, $\gamma_j'(k)$, $p_j^c(k-1)$ and $p_j^c(k)$ as the state variables for the j^{th} contributing injector, which, together, form the following 4×1 state vector ($j = 1, 2, \ldots, N$):

$$\begin{aligned}\boldsymbol{x}_j(k) &= \left[x_{j1}(k), x_{j2}(k), x_{j3}(k), x_{j4}(k)\right]^T \\ &= \left[\alpha_j(k), \gamma_j'(k), p_j^c(k-1), p_j^c(k)\right]^T\end{aligned} \tag{2-9}$$

Aggregating the state vectors of all N contributing injectors, one obtains a complete $4N \times 1$ state vector for the entire producer-centric model, as:

$$\boldsymbol{x}(k) = \left[\boldsymbol{x}_1(k)^T, \boldsymbol{x}_2(k)^T, \cdots, \boldsymbol{x}_N(k)^T\right]^T \tag{2-10}$$

And, based on (2-7), (2-9) and (2-10), the SVM for the entire producer-centric system can be constructed as [73, 121]:

$$\begin{cases} \boldsymbol{x}(k+1) = \begin{bmatrix} x_{11}(k) \\ x_{12}(k) \\ x_{14}(k) \\ 2x_{11}(k)x_{14}(k) - x_{11}^2(k)x_{13}(k) + x_{12}(k)i_{m,1}(k) \\ x_{21}(k) \\ x_{22}(k) \\ x_{24}(k) \\ 2x_{21}(k)x_{24}(k) - x_{21}^2(k)x_{23}(k) + x_{22}(k)i_{m,2}(k) \\ \vdots \\ x_{N1}(k) \\ x_{N2}(k) \\ x_{N4}(k) \\ 2x_{N1}(k)x_{N4}(k) - x_{N1}^2(k)x_{N3}(k) + x_{N2}(k)i_{m,N}(k) \end{bmatrix} + \boldsymbol{n}_x(k) \\ p(k+1) = p_1^c(k+1) + p_2^c(k+1) + \cdots + p_N^c(k+1) + n_p(k+1) \\ \qquad = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(k+1) + n_p(k+1) \end{cases} \tag{2-11}$$

where the two equations are called *state equation* and *measurement equation*, respectively; and $\boldsymbol{n}_x(k) = \left[\boldsymbol{n}_{x_1}(k)^T, \boldsymbol{n}_{x_2}(k)^T, \ldots, \boldsymbol{n}_{x_N}(k)^T\right]^T$ and $n_p(k+1)$ are additive zero-mean white noises with covariance matrix $\boldsymbol{Q}_k$ and variance r_{k+1}, respectively. Each component $\boldsymbol{n}_{x_j}(k)$ $(j = 1, 2, \ldots, N)$ is $[n_{\alpha_j}(k)\ \ n_{\gamma_j'}(k)\ \ 0\ \ n_{p_j^c}(k+1)]^T$. The explicit form of $\boldsymbol{Q}_k$ is:

$$\boldsymbol{Q}_k = diag\left[r_{\alpha_1}, r_{\gamma_1'}, 0, r_{p_1^c}, r_{\alpha_2}, r_{\gamma_2'}, 0, r_{p_2^c}, \cdots, r_{\alpha_N}, r_{\gamma_N'}, 0, r_{p_N^c} \right] \tag{2-12}$$

2.2.4 Original State Variable Model

For practical use, the SVM in (2-11) has to be modified in two important ways: 1) IPR_j $(\ j = 1, 2, \ldots, N)$ is treated as a state variable, and is estimated directly. Its error (pseudo-) variance can be used to provide upper and lower bounds for the estimates; and 2) α_j and IPR_j $(\ j = 1, 2, \ldots, N)$ must be positive numbers, but sometimes their estimates become negative due to strong noises and other uncertainty factors; therefore, $\sqrt{\alpha_j}$ and $\sqrt{IPR_j}$ are used as state variables instead of α_j and γ_j'. Regardless of the sign of the estimated values of $\sqrt{\alpha_j}$ and $\sqrt{IPR_j}$, their squared values always give positive estimates for α_j and IPR_j.

Consequently, $\left[\sqrt{IPR_j(k)}\ \ \sqrt{\alpha_j(k)}\ \ p_j^c(k-1)\ \ p_j^c(k)\right]^T$ is used as the state vector, $\boldsymbol{x}_j(k)$, for the j^{th} contributing injector. As in the above section, the complete state vector is $\boldsymbol{x}(k) = [\boldsymbol{x}_1(k)^T, \boldsymbol{x}_2(k)^T, \ldots, \boldsymbol{x}_N(k)^T]^T$.

Based on (2-8), $\gamma_j'(k)$ can be expressed in terms of $\alpha_j(k)$ and $IPR_j(k)$ as:

$$\gamma_j'(k) = IPR_j(k)(1-\alpha_j(k))^2 \tag{2-13}$$

which allows us to re-express (2.7) in terms of the new state variables as:

$$p_j^c(k+1) - 2\sqrt{\alpha_j}^2 p_j^c(k) + \sqrt{\alpha_j}^4 p_j^c(k-1) = \sqrt{IPR_j}^2 (1-\sqrt{\alpha_j}^2)^2 i_{m,j}(k) + n_{p_j^c}(k) \qquad (2\text{-}14)$$

Consequently, our modified SVM is

$$\begin{cases} \boldsymbol{x}(k+1) = \begin{bmatrix} x_{11}(k) \\ x_{12}(k) \\ x_{14}(k) \\ 2x_{12}^2(k)x_{14}(k) - x_{12}^4(k)x_{13}(k) + x_{11}^2(k)(1+x_{12}^2(k))^2 i_{m,1}(k) \\ x_{21}(k) \\ x_{22}(k) \\ x_{24}(k) \\ 2x_{22}^2(k)x_{24}(k) - x_{22}^4(k)x_{23}(k) + x_{21}^2(k)(1+x_{22}^2(k))^2 i_{m,2}(k) \\ \vdots \\ x_{N1}(k) \\ x_{N2}(k) \\ x_{N4}(k) \\ 2x_{N2}^2(k)x_{N4}(k) - x_{N2}^4(k)x_{N3}(k) + x_{N1}^2(k)(1+x_{N2}^2(k))^2 i_{m,N}(k) \end{bmatrix} + \boldsymbol{n}_x(k) \\ p(k+1) = p_1^c(k+1) + p_2^c(k+1) + \cdots + p_N^c(k+1) + n_p(k+1) \\ \qquad = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; \cdots \;\; 0 \;\; 0 \;\; 0 \;\; 1]\boldsymbol{x}(k+1) + n_p(k+1) \end{cases} \qquad (2\text{-}15)$$

where, $\boldsymbol{n}_x(k)$ and $n_p(k+1)$ have been described in Section 2.2.3.

2.2.5 Extended Kalman Filter

Details about the EKF can be found in [73, 121]. The nonlinear (NL) SVM for an EKF has the following general form:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{f}[\boldsymbol{x}(k), k] + \boldsymbol{n}_x(k) \\ \boldsymbol{y}(k+1) = \boldsymbol{h}[\boldsymbol{x}(k+1), k+1] + \boldsymbol{n}_y(k) \end{cases} \qquad (2\text{-}16)$$

where, in general, $\boldsymbol{f}[\bullet]$ and $\boldsymbol{h}[\bullet]$ are nonlinear functions of the state vector, and $\boldsymbol{n}_x(k)$ and $\boldsymbol{n}_y(k)$ correspond to the additive zero-mean white noises for the state and measurement equations, respectively. In the EKF, $\boldsymbol{f}[\bullet]$ is linearized about $\hat{\boldsymbol{x}}(k|k)$, i.e.,

$$\boldsymbol{x}(k+1) \approx \boldsymbol{f}\left[\hat{\boldsymbol{x}}(k \mid k), k\right] + \boldsymbol{F}_x\left[\hat{\boldsymbol{x}}(k \mid k), k\right] \times \left[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k \mid k)\right] + \boldsymbol{n}_x(k) \tag{2-17}$$

where $\boldsymbol{F}_x = \partial \boldsymbol{f}\left[x(k), k\right] / \partial \boldsymbol{x}(k)$ is a $4N \times 4N$ Jacobian matrix, the explicit forms of which are given below for our NL state equation in (2-15):

$$\boldsymbol{F}_x = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{A}_N \end{pmatrix} \tag{2-18}$$

where the explicit form of $\boldsymbol{A}_j$ is:

$$\boldsymbol{A}_j = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 2x_{j1}(k)(1+x_{j2}^2(k))^2 i_{m,j}(k) & \begin{matrix} 4x_{j2}(k)x_{j4}(k) - 4x_{j2}^3(k)x_{j3}(k) + \\ 4x_{j1}^2(k)(1+x_{j2}^2(k))x_{j2}(k)i_{m,j}(k) \end{matrix} & -x_{j2}^4(k) & 2x_{j2}^2(k) \end{bmatrix} \tag{2-19}$$

What distinguishes our SVM in (2-15) from the more general SVM in (2-16) is that, in (2-15), $\boldsymbol{h}[\bullet]$ is linear, and can be expressed as:

$$y(k+1) = \boldsymbol{H}\boldsymbol{x}(k+1) + n_y(k+1) \tag{2-20}$$

where

$$\boldsymbol{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-21}$$

is a 1×4*N* vector.

The EKF has two stages, *Predictor* and *Corrector*, and is summarized as follows:

1. Initialize the EKF with $\hat{\boldsymbol{x}}(0|0)$, $\boldsymbol{P}(0|0)$, $\boldsymbol{Q}_k$ and $\boldsymbol{r}_k$.
2. Predictor (k = 0, 1,…):

$$\hat{\boldsymbol{x}}(k+1|k) = \boldsymbol{f}\left[\hat{\boldsymbol{x}}(k|k), k\right] \tag{2-22}$$

$$\boldsymbol{P}(k+1|k) = \boldsymbol{F}_x\left[\hat{\boldsymbol{x}}(k|k), k\right]\boldsymbol{P}(k|k)\boldsymbol{F}_x'\left[\hat{\boldsymbol{x}}(k|k), k\right] + \boldsymbol{Q}_k \tag{2-23}$$

3. Corrector (k = 0, 1,…):

$$\hat{\boldsymbol{x}}(k+1|k+1) = \hat{\boldsymbol{x}}(k+1|k) + \boldsymbol{K}(k+1)\times\left[y(k+1) - \boldsymbol{H}\hat{\boldsymbol{x}}(k+1|k)\right] \tag{2-24}$$

$$\boldsymbol{K}(k+1) = \frac{\boldsymbol{P}(k+1|k)\boldsymbol{H}'}{\boldsymbol{H}\boldsymbol{P}(k+1|k)\boldsymbol{H}' + \boldsymbol{r}_{k+1}} \tag{2-25}$$

$$\boldsymbol{P}(k+1|k+1) = \left[\boldsymbol{I} - \boldsymbol{K}(k+1)\boldsymbol{H}\right]\boldsymbol{P}(k+1|k) \tag{2-26}$$

2.2.6 Real Data Processing

The real data used in our study is from a section of an oil field. Most of the injectors in this section have two completions; and because the injection rates of different completions for the same injector are separate, each completion is treated independently, i.e., each completion is treated as an independent "injector" in our SVM. The total number of producers in this field is denoted by N_P and the total number of injector completions is denoted by N_I; and the well names have been removed and re-labeled.

The data starts from Jan. 1st, 2005, which is labeled as day 1 in our figures, and ends on Jul. 31st, 2008, which is labeled as day 1308. Both injection and production rates are well-test data. The sampling rate of injection rate is one measurement per day and the sampling rate of production rate is, on average, one measurement every eight to ten days. *For days when production rates were not available, we used the value from the last available data point all the way until the next available data point.* This so-called Zero-Order Hold method is the simplest and one of the most often used data-interpolation approach in practice. Although later we are going to propose a complex data-interpolation method, for the purpose of explaining the framework of our estimation strategy, we assume the Zero-Order-Held production rates meet our process requirements.

Our strategy for processing the entire section has been to apply our producer-centric EKF to each producer separately. We choose a producer labeled as P-130 to give an example of this process.

A very important first step is to choose an initial set of injectors that are possibly contributing to the producer. A relatively simple and commonly used way for doing this is to use expert knowledge provided by petroleum engineers that are familiar with this field. For this section, it is known that there are parallel fractures along each well that align 45° to NE. It is believed that injectors located along this fracture alignment are more likely to contribute to a producer. Knowing this, one strategy for choosing the injectors that contribute to a producer is to draw an ellipse centered about the producer whose major axis is along the fracture alignment, and to assume the injectors inside the ellipse are influencing that

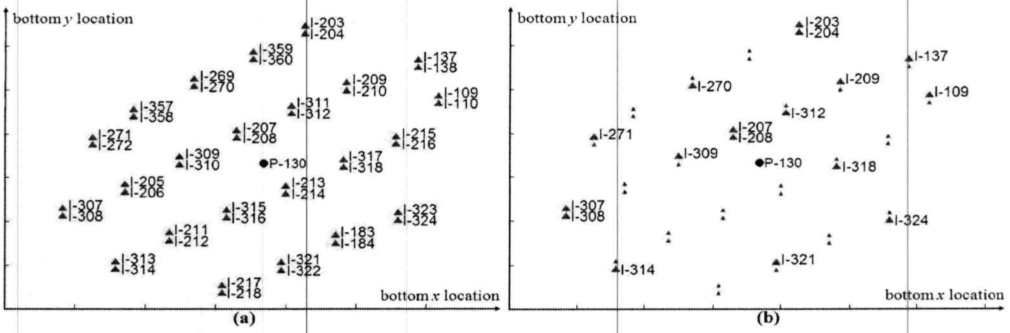


Figure 2.2 When s = 1, (a) P-130's local area and 46 initial completions included in its producer-centric model; and (b) the 17 completions that remained after the elimination process.

producer. Additional expert knowledge was provided about the size of the ellipse, i.e., in general the major axis of the ellipse should be 700 feet long and its minor axis should 500 feet long. Because these lengths are subjective, they may vary from producer to producer; hence, a strategy for choosing an optimal ellipse size is presented in Section 2.2.7.

We began by using a $700' \times 500'$ ellipse for a producer. Completions inside this ellipse are included in our EKF model. To give the reader a clearer picture of this, P-130 and injectors inside the ellipse are depicted in Fig. 2.2(a) by circle and triangles, respectively. Note that the upper and lower triangles at the same location represent short and long completion of an injector, respectively. And there are 46 completions in this model.

After applying our EKF to this single producer-46 completions model, we obtained the IPR curves that are depicted in Fig. 2.3(a).

46 completions is not a small number, so it is very likely that some of these completions may be irrelevant to P-130. Observe, from Fig. 2.3(a), that there are many IPR curves that only have very small values. We assumed, therefore, that the completions that have very low IPR values do not influence P-130 and should be eliminated from the model.

More specifically, we computed the mean of the IPR values for the most recent month (i.e., days 1278 ~ day1308) for each completion and compared it with a chosen threshold. A completion was kept in the model only if its IPR value was greater than or equal to the threshold; otherwise, that completion was eliminated from the model.

Our strategy for choosing a good threshold is dependent on the number of completions

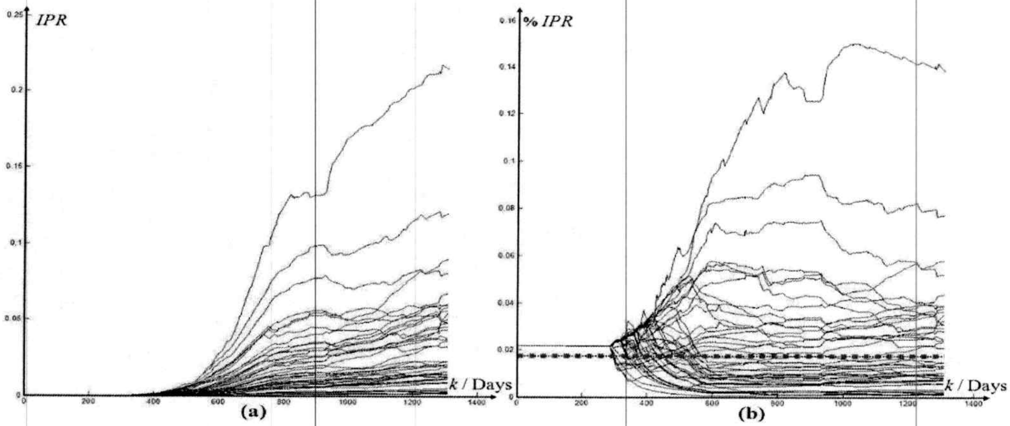


Figure 2.3 When s = 1, (a) IPR curves for the 46 initial completions; and (b) normalized IPR curves (solid lines) and the elimination threshold (dotted line).

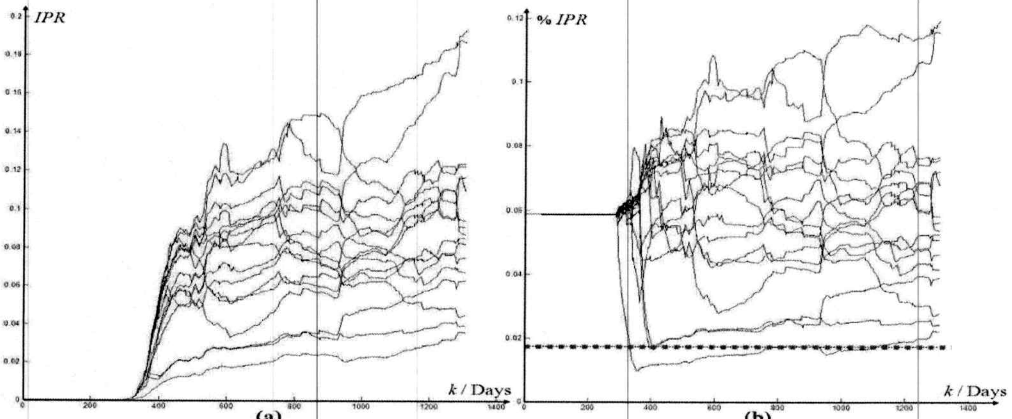


Figure 2.4 When s = 1, (a) IPR curves for the 17 remaining completions after the elimination process; and (b) normalized IPR curves (solid lines) and the elimination threshold (dotted line).

inside the ellipse, and was inspired by the fact that an existing common, simple and practical way for deciding the impacts of injectors in water-flood management is to assign equal weights to them. For example, if *N* injectors are thought to be impacting a producer, then one assumes each of them has an impact weight of 1/*N*. We modified this by using ρ/N as the threshold, where we found, by trial and error, that 80% of 1/*N* does a good job as a threshold.

The IPR curves in Fig. 2.3(a) are in absolute values, but, to use a percentage threshold they need to be normalized. This is done by dividing each IPR curve by the sum of all the IPR curves. These IPR curves in percentages are depicted in Fig. 2.3(b). Also shown on that figure is the $80\%/N = 1.74\%$ threshold line (dotted) when *N* = 46.

After eliminating all of the completions, the *IPR*s of which fell below 1.74%, only 17 completions were left. The remaining completions and eliminated completions are depicted in Fig. 2.2(b) by regular size triangles and smaller triangles, respectively.

Our EKF was then re-applied to this single producer, but for only the 17 remaining completions. The resulting IPR values are depicted in Fig. 2.4. Observe that, although some portion of the three lowest IPR curves are still below the threshold, we are only considering the mean IPRs of the most recent month, which occurs at the right-end of the data, and none of the mean IPR values for that month fall below this threshold. Consequently, none of the 17 injectors were eliminated, and therefore, our processing for P-130 is completed. If, perchance, some of the 17 injectors had been eliminated, we would have then repeated this procedure until a situation was reached where no more of the injectors are eliminated. During such an iteration process, the threshold would be kept at 80%/*N*, where *N* is the *initial* number of injectors, in this case, 46. *N* is not chosen to be the surviving number of injectors because the resulting threshold would become so large that it is possible that too many injectors would be eliminated.

The EKF procedure that we have described is performed for each of the producers in the section. If parallel processing is available then this processing can be done in parallel because of its producer-centric nature.

2.2.7 Validation using Historical Matching

How does one validate our EKF method? Unless a field test is performed, there are no truth data available as there would be when synthetic data or reservoir simulation data are used (as in [110]). Although field tests can be performed, they are disruptive and expensive. Another approach, the one we have used is history matching, i.e., we go back in history and locate a time point at which a significant step change in an injection rate has occurred (as suggested to us by petroleum engineers), and use the estimated IPR at that point to "forecast" the production rate at some future time. This forecasted production rate is then compared with the real production rate, since the latter is available to us as a part of the historical data record we began with. Forecasting of the production rate is accomplished by using the predictor equation (2-21) of our EKF. We illustrate this process next by an example.

By carefully inspecting the historical injection rates for all the remaining 17 completions for P-130, we noticed that most of them had a significant rise in their injection rates around day 1045; therefore, day M=1045 was chosen as the starting day for our forecasts. Starting from day M, we used the estimated parameter values obtained at that day, i.e. (j = 1,…, N):

$$\hat{\boldsymbol{x}}_j(M \mid M) = \left[\sqrt{I\hat{P}R_j(M \mid M)}, \sqrt{\hat{\alpha}_j(M \mid M)}, \hat{p}_j^c(M-1 \mid M), \hat{p}_j^c(M \mid M)\right]^T \tag{2-27}$$

and the planned injection rates, i.e., $i_j(M)$,..., $i_j(M+30)$ (j = 1,..., N), to forecast the next month's production rates, day-by-day, i.e.:

$$\begin{cases} \hat{\boldsymbol{x}}(M+1 \mid M) = \boldsymbol{f}\left[\hat{\boldsymbol{x}}(M \mid M)\right] \\ \hat{p}(M+1 \mid M) = \boldsymbol{H}\hat{\boldsymbol{x}}(M+1 \mid M) \end{cases}$$
$$\begin{cases} \hat{\boldsymbol{x}}(M+2 \mid M) = \boldsymbol{f}\left[\hat{\boldsymbol{x}}(M+1 \mid M)\right] \\ \hat{p}(M+2 \mid M) = \boldsymbol{H}\hat{\boldsymbol{x}}(M+2 \mid M) \end{cases} \quad \text{(2-28)}$$
$$\vdots$$
$$\begin{cases} \hat{\boldsymbol{x}}(M+30 \mid M) = \boldsymbol{f}\left[\hat{\boldsymbol{x}}(M+29 \mid M)\right] \\ \hat{p}(M+30 \mid M) = \boldsymbol{H}\hat{\boldsymbol{x}}(M+30 \mid M) \end{cases}$$

where $\boldsymbol{H}$ is the same as in (2-21).

After one month, new injection and production rate measurements [$i_{m,j}(k)$ and $p_m(k)$ (j = 1,..., N; $k = M + 1$,..., $M + 30$)] are available, so we update our EKF using those new measurements, to obtain the updated state vector, i.e., $\widehat{\boldsymbol{x}}(M+30|M+30)$, that are then used to forecast another month's production rates by using (2-28) again. This process is repeated until we have run out of data history.

We observed that IPR curves are not flat during an entire month (e.g., see Fig. 2.4); hence, we computed their gradients at the day the forecast starts, and assumed the IPRs would follow the gradients during the following one month's forecasts, e.g., the first month's forecasts start at M = 1045, and the gradient of $\sqrt{I\hat{P}R_j(M|M)}$ and $\sqrt{\hat{\alpha}_j(M|M)}$, $g_{\sqrt{IPR_j}}(M)$ and $g_{\sqrt{\alpha_j}}(M)$ (j =1,..., N), can be approximated as:

$$\begin{cases} g_{\sqrt{IPR_j}}(M) = \dfrac{\sqrt{I\hat{P}R_j(M+1 \mid M)} - \sqrt{I\hat{P}R_j(M \mid M)}}{(M+1)-M} \\ g_{\sqrt{\alpha_j}}(M) = \dfrac{\sqrt{\hat{\alpha}_j(M+1 \mid M)} - \sqrt{\hat{\alpha}_j(M \mid M)}}{(M+1)-M} \end{cases} \tag{2-29}$$

It follows, therefore, that:

$$\begin{cases} \sqrt{I\hat{P}R_j(M+1 \mid M)} = g_{\sqrt{IPR_j}}(M) + \sqrt{I\hat{P}R_j(M \mid M)} \\ \sqrt{\hat{\alpha}_j(M+1 \mid M)} = g_{\sqrt{\alpha_j}}(M) + \sqrt{\hat{\alpha}_j(M \mid M)} \end{cases} \tag{2-30}$$

Because we assume $\sqrt{IPR_j}$ and $\sqrt{\alpha_j}$ follow the gradient for the entire month, (2-30) provides the following daily interpolation formulas that were used by us (d = 1,…, 30):

$$\begin{cases} \sqrt{I\hat{P}R_j(M+d \mid M)} = g_{\sqrt{IPR_j}}(M) + \sqrt{I\hat{P}R_j(M+d-1 \mid M)} \\ \sqrt{\hat{\alpha}_j(M+d \mid M)} = g_{\sqrt{\alpha_j}}(M) + \sqrt{\hat{\alpha}_j(M+d-1 \mid M)} \end{cases} \tag{2-31}$$

Results have shown that forecasts incorporating these interpolations outperformed the forecasts that did not use them, so they have been used in all of our results that are described below.

The overall historical production rate for P-130 is plotted as the solid line in Fig. 2-5, whereas the dotted line shows the monthly forecasts, starting at day 1045. Observe that the production starts to increase after day 1045 and that the forecasted production has detected this increase, although there are some forecasting errors. For viewing convenience, we zoom in to the prediction interval and this is plotted in Fig. 2-6(a), and, the prediction error is plotted in Fig. 2-6(b).

Some summaries of the monthly errors are given in Table 2.1. It contains monthly error summaries for different values of a scalar parameter, *s*, which is used in Section 2.3.4 to determine the size of the ellipse for establishing the initial set of injectors. Our present case corresponds to the row in Table 2.1 for which $s = 1.0$.

The monthly forecasting began on day 1045 and continued for 8 months. In Table 2.1, the average daily prediction errors for each month, and the average daily error over the entire eight-month period, have been computed. Additionally, Error-Production-Ratio (EPR), which is the ratio of the average daily prediction error to the average daily production rate over 8 months, has been computed. EPR is useful because petroleum engineers have told us that the measurement-noise level for this section is around 10-20%, e.g. if the measured production is 300 barrels, then about 30-60 barrels is noise. The EPR for $s = 1$ is 10.23% which is a very good value.

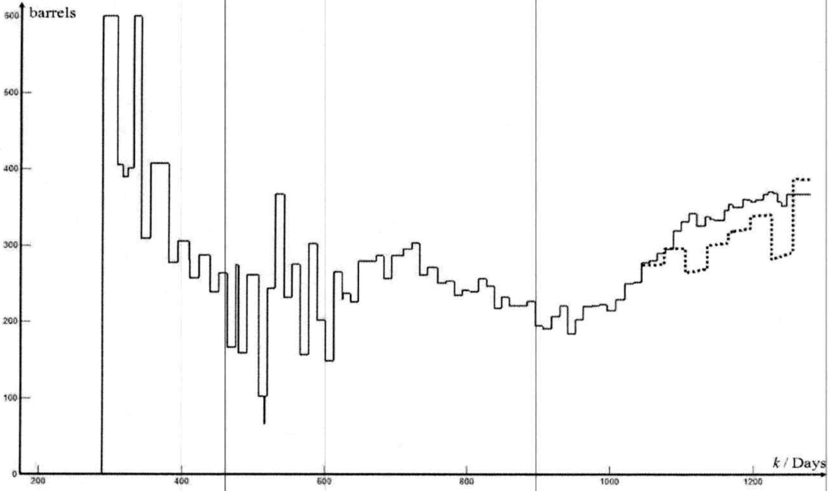


Figure 2.5 Historical production rate for P-130 (solid line) and predicted production rate (dotted line) when $s = 1$. Prediction begins at day 1045.

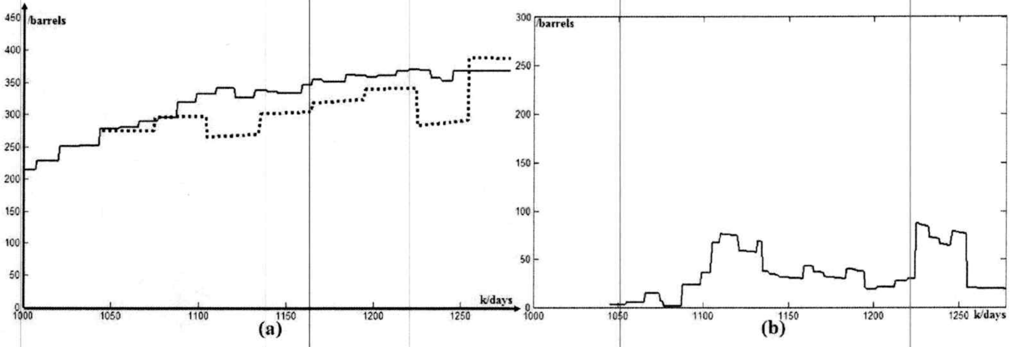


Figure 2.6 When s = 1, (a) Historical production rate for P-130 (solid line), and predicted production rate (dotted line); and (b) error between the actual and predicted production rate.

Table 2.1 Average daily production prediction error for eight consecutive months, its overall average and the overal average Error-Production-Ratio

	1st month	2nd month	3rd month	4th month	5th month	6th month	7th month	8th month	average	*EPR*
S=1.0	7.3067	16.259	66.977	34.464	34.588	23.108	75.625	19.180	34.688	10.23%
S=0.9	3.7455	15.506	38.252	26.247	38.229	22.287	43.321	13.253	28.779	8.49%
S=0.8	7.2293	15.273	33.060	16.235	43.111	13.786	47.965	13.034	26.583	7.84%
S=0.7	4.4729	14.829	26.305	17.878	43.224	14.721	35.270	13.643	13.268	3.91%
S=0.6	5.8975	14.161	26.667	18.717	43.360	14.663	51.121	20.796	13.234	3.90%
S=0.5	9.7345	16.310	26.735	18.209	48.188	10.872	56.535	11.865	13.833	4.08%

2.2.8 Optimization of Initial Set of Injectors

An issue that plays a very critical role in our EKF processing is how to choose a set of initial injectors for the producer-centric model. Although it has been previously mentioned that an ellipse with 700 feet major axis and 500 feet minor axis is used to select the initial set of injectors, in general, an optimal size for the ellipse may be very different for different producers. If the ellipse is too small, one would miss some influential injectors from the very beginning. On the other hand, if the ellipse is too large it could include many irrelevant injectors. Such injectors may significantly bias the first round of EKF processing, because they can affect the threshold elimination process that has been described above.

Inspired by techniques used in standard optimization problems, and the history matching results described above, our strategy for finding an optimal ellipse size (for each producer) was to use the average daily prediction error as an objective function and to minimize it with respect to ellipse size. To do this, we used an ellipse with 700 feet major axis and 500 feet minor axis as a standard ellipse and then scaled its major and minor axes by the same scalar, s; hence, as different s values are used one obtains ellipses of different sizes.

In theory, s should be discretized very finely and also range from a small enough value to a large enough value so that the minimized point of the objective function is its global minimum. Unfortunately, using too many values of s is computationally very costly; hence, in this study we chose $s = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. We have observed that when $s = 1.0$ (which is the case discussed above) the ellipse is already large enough and usually includes more than 20 injectors, and (as we will see below), when $s = 0.5$ the ellipse only includes a few neighboring injectors. Additionally, we observed, after several tests, that s does not have to be discretized very finely, e.g., when s changes from 0.95 to 0.9 the ellipse only shrinks a little and no injectors from the $s = 0.95$ ellipse are left out of the $s = 0.9$ ellipse.

EKF processing for different values of s proceeds in exactly the same manner as described above for s = 1. Results for $s = 0.9$, 0.6 and 0.5 are given in Figs. 2.7-2.15. Summaries of the prediction errors for $s = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ are also given in Table 2.1. Observe that the objective function is minimized when $s = 0.6$. Thus, for P-130, the optimal ellipse used to choose the initial injector set is one with $0.6 \times 700 = 420$ feet major- and $0.6 \times 500 = 300$ feet minor axis.

2.2.9 Producer Centric to Injector Centric Conversion

In above sections, we presented a complete procedure to process only one producer-centric model. Of course, this procedure has to be applied to every producer in the entire section (or in the entire field, if such data are available) The IPR results of doing this can then be summarized in a table, which in our case has N_P rows (producers) and N_I columns (injectors). A small portion of this table is depicted in Table 2.2. The non-zero values in this table are for those injectors that have survived our EKF processing-procedure. Zero values are for the remaining injectors. This table allows one to quickly look up the IPR value between any producer-injector pair.

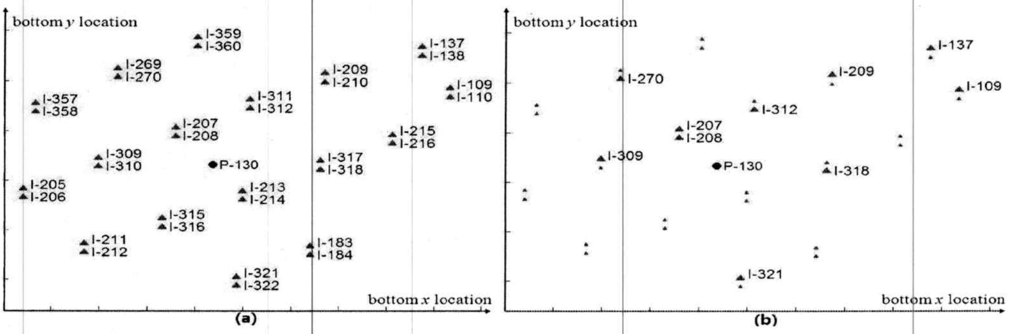


Figure 2.7 When $s = 0.9$, (a) P-130's local area and initial completions included in its model; and (b) the 10 completions that remained after the elimination process.

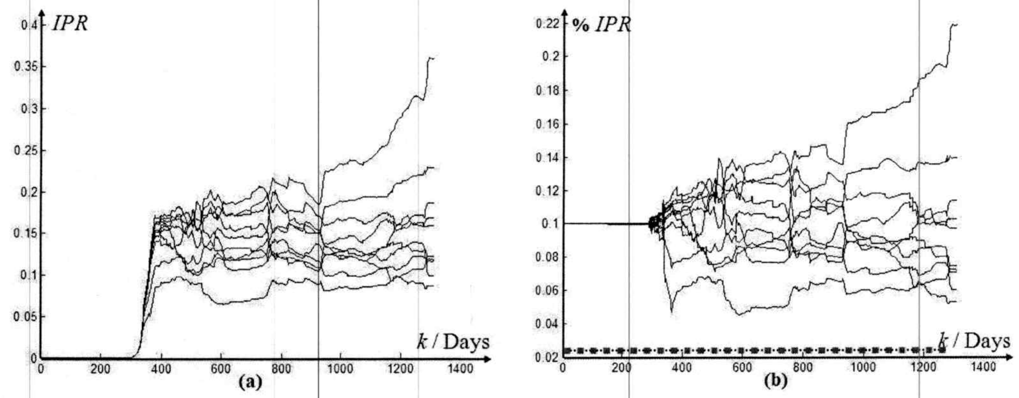


Figure 2.8 When s = 0.9, (a) *IPR* curves for the 10 remaining completions after the elimination process; and (b) normalized *IPR* curves (solid lines) and elimination threshold (dotted line).

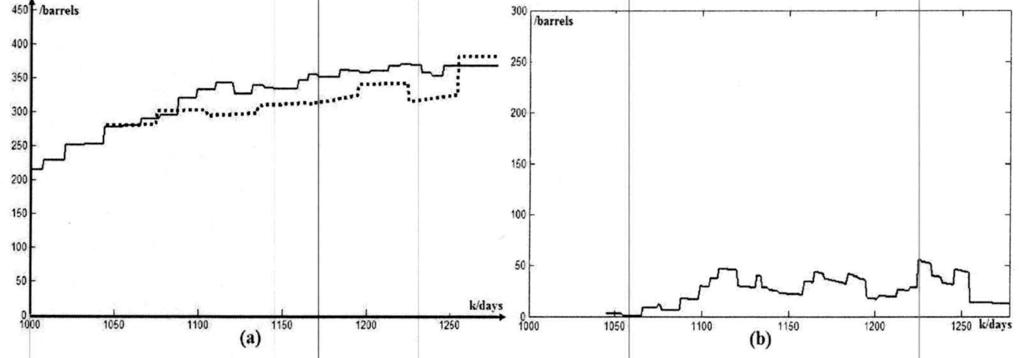


Figure 2.9 When s = 0.9, (a) Historical production rate for P-130 (solid line), and the predicted production rate (dotted line); and (b) error between the actual and the predicted rate.

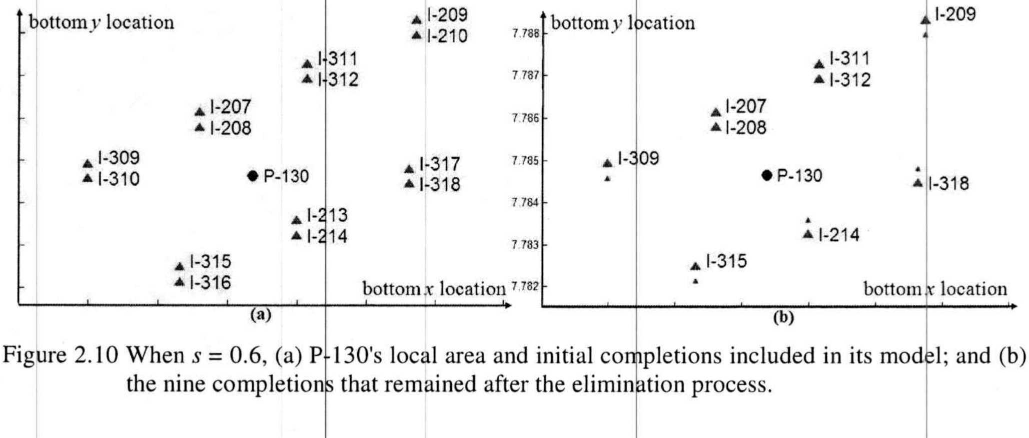


Figure 2.10 When s = 0.6, (a) P-130's local area and initial completions included in its model; and (b) the nine completions that remained after the elimination process.

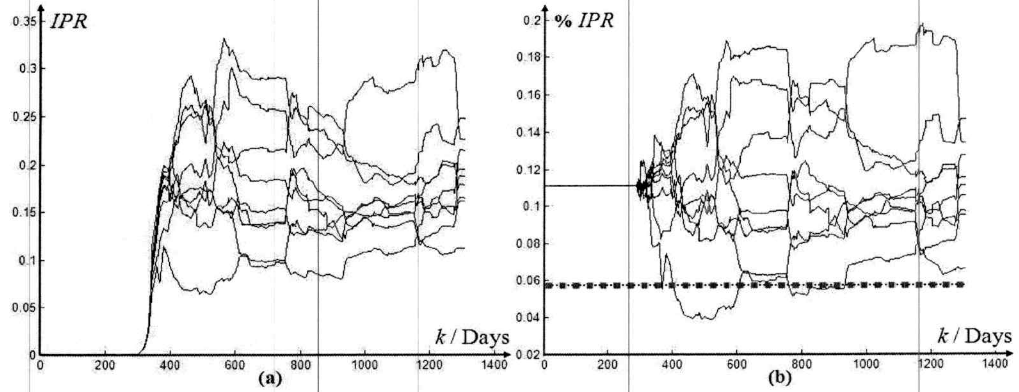


Figure 2.11 When s = 0.6, (a) *IPR* curves for the nine remaining completions after elimination; and (b) normalized *IPR* curves (solid lines) and the elimination threshold (dotted line).

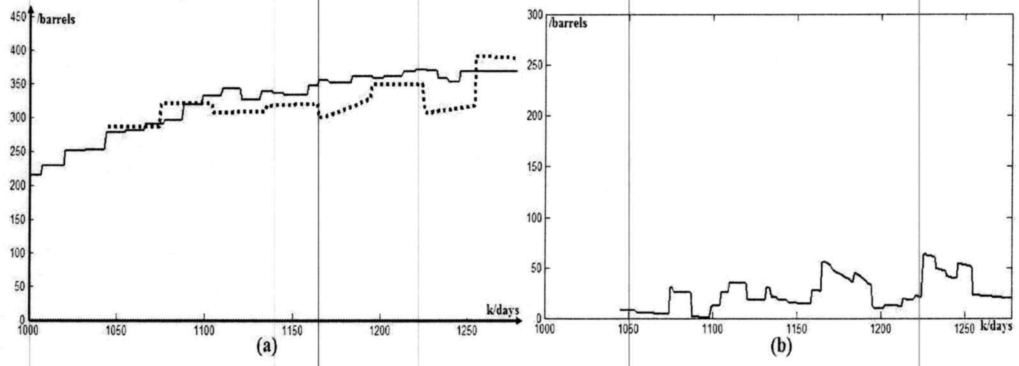


Figure 2.12 When s = 0.6, (a) Historical production rate for P-130 (solid line), and the predicted production rate (dotted line); and (b) error between the actual and the predicted rate.

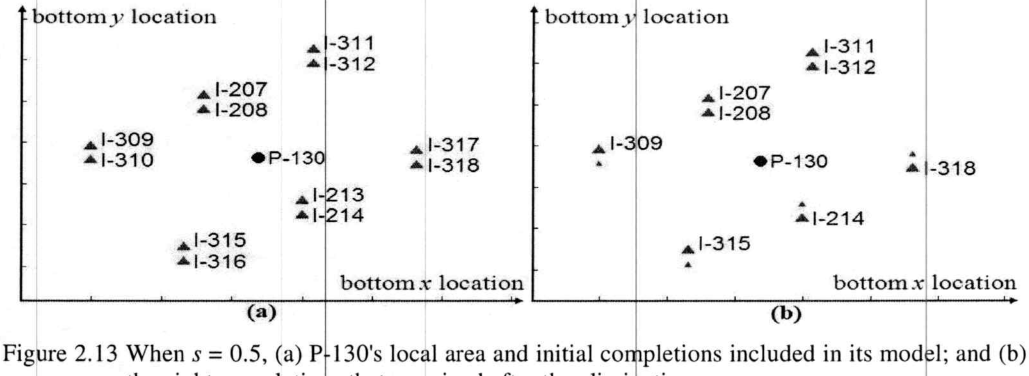


Figure 2.13 When s = 0.5, (a) P-130's local area and initial completions included in its model; and (b) the eight completions that remained after the elimination process.

Reservoir engineers are more interested in an injector-centric viewpoint than in a producer-centric viewpoint, because they have control over water allocation at each injector; hence, for them, it would be better if the results were injector-centric. Observe that Table 2.2 can be viewed in two different ways. Data are entered into it one row at a time (producer-centric), but it can then be viewed column-wise (injector-centric). Unfortunately, the numerical IPR values are still difficult to interpret. One solution to this is to normalize the

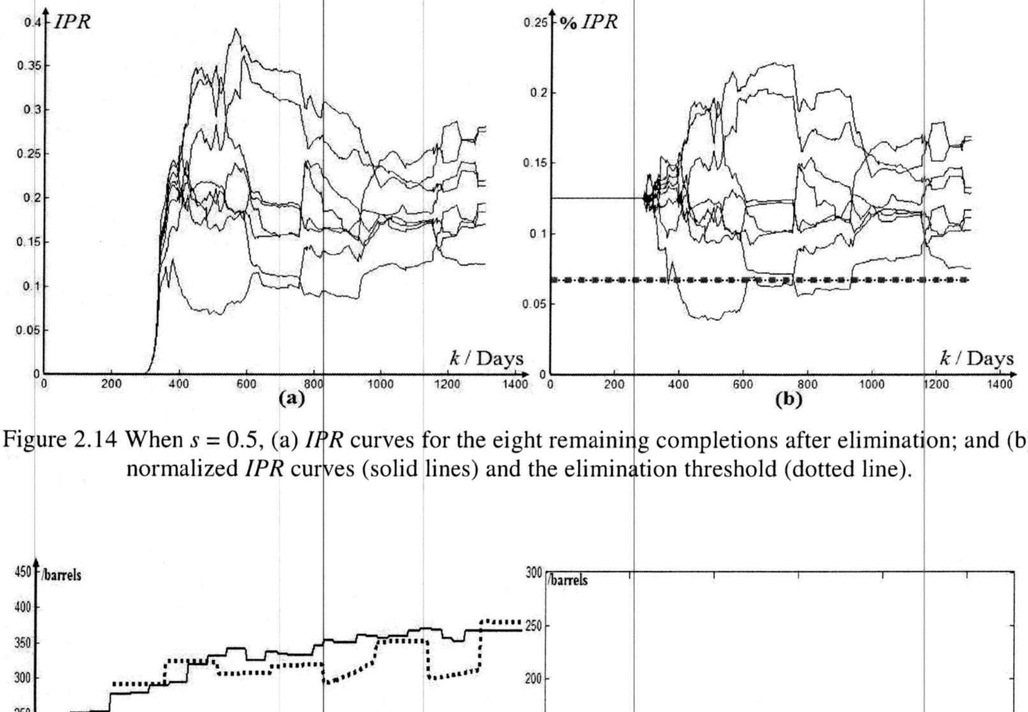


Figure 2.14 When s = 0.5, (a) IPR curves for the eight remaining completions after elimination; and (b) normalized IPR curves (solid lines) and the elimination threshold (dotted line).

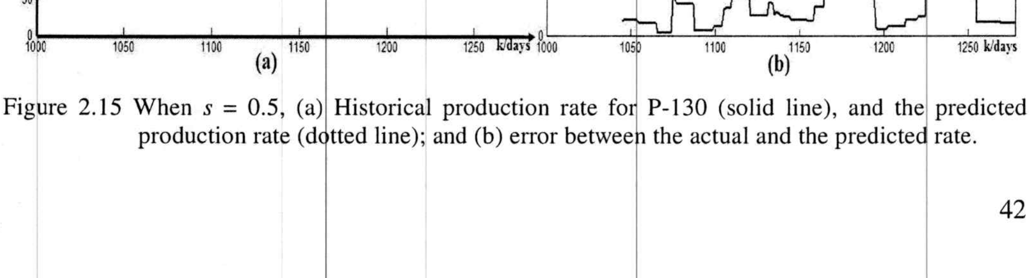


Figure 2.15 When s = 0.5, (a) Historical production rate for P-130 (solid line), and the predicted production rate (dotted line); and (b) error between the actual and the predicted rate.

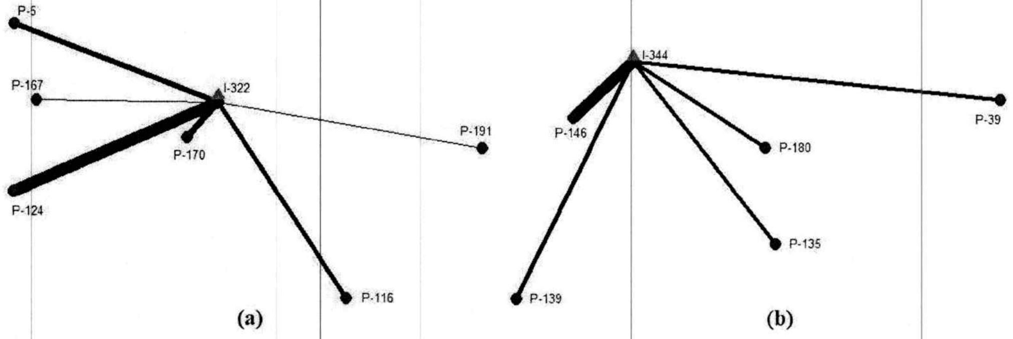


Figure 2.16 Graphical representations of the normalized IPR values for (a) injector I-322; and (b) injector I-344 (see Table 2.3).

Table 2.2 Upper left portion of the (un-normalized) IPR table for the entire section of N_P producers (rows) and N_I injectors (column)

	I-1	I-2	I-3	I-4	I-5	I-6	I-7	I-8	I-9	I-10
P-1	0.0211	0	0.1571	0	0	0	0.0692	0	0	0
P-2	0	0	0	0	0	0	0	0	0	0
P-3	0	0	0	0	0	0	0	0	0	0
P-4	0	0	0	0	0	0	0	0	0	0
P-5	0	0	0	0	0	0	0	0	0	0
P-6	0.0683	0	0	0	0	0	0	0	0	0
P-7	0	0	0	0	0	0	0.0339	0	0	0
P-8	0	0	0	0	0	0	0	0	0	0
P-9	0	0	0	0	0.1429	0	0	0.1334	0	0
P-10	0.0946	0	0	0.0852	0	0	0.1477	0	0	0
P-11	0.1062	0	0	0.0782	0	0	0	0	0	0
P-12	0.0678	0	0	0.0541	0	0	0.1014	0	0	0
P-13	0.0512	0	0	0	0	0	0	0	0	0
P-14	0	0	0	0	0	0	0	0	0	0
P-15	0	0	0	0	0	0	0	0	0	0.0402
P-16	0	0	0	0	0	0	0	0	0	0.0329
P-17	0	0	0	0	0	0	0	0	0	0.0543
P-18	0	0	0	0	0	0	0	0	0	0
P-19	0.0360	0	0	0	0	0	0	0	0	0
P-20	0	0	0	0	0	0	0	0	0	0
P-21	0	0	0	0	0	0	0	0	0	0
P-22	0	0	0	0	0	0	0	0	0	0
P-23	0	0	0	0	0	0	0	0	0	0.0911
P-24	0	0	0	0	0	0	0	0	0	0
P-25	0	0	0	0	0	0	0	0	0	0
P-26	0	0	0	0.0536	0	0	0.0504	0	0	0
P-27	0	0.0654	0	0	0	0	0	0	0	0

Table 2.3 Normalized IPRs for injectors I-322 and I-344 (Entries are in %)

	P-124	P-170	P-5	P-116	P-167	P-191
I-322	41.77	21.42	20.51	16.16	0.54	0.20

	P-146	P-139	P-180	P-39	P-135
I-344	41.77	17.70	14.32	14.14	12.07

IPR values into percentage values by dividing the values in each column by the sum of values in the column. This allows us to observe what percent of the total water that is allocated to an injector goes to a specific producer. We show such results in Table 2.3 for two injectors. Observe that very large amounts of water from injectors, I-322 and I-344, go to producers, P-124 and P-146, respectively.

Fig. 2.16 depicts a novel graphical way to represent these injector-centric results. Observe that we have connected an injector to all of the producers that have nonzero IPRs using lines of different widths. The higher the %IPR value is, the thicker the line is, and the lower the %IPR value is the thinner the line is.

2.3 An Inequality-Constrained Extended Kalman Filter

Independent EKF processing of the producer-centric SVMs is at least partially responsible (there could be other possible causes yet be discovered) for some estimated injector-centric sums being greater than one. This can be best illustrated using the designed simulated tests given in Section 2.3.1. Then, in Section 2.3.2, we present a real field case, where the constraint on injector-centric sum has been severely violated, to show how the situation can actually occur in practice.

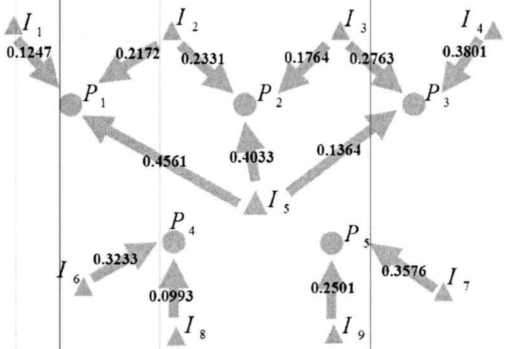


Figure 2.17 Geographic map and designed IPR values for the simulated mini-oilfield. Numerical values on the arrows denote corresponding IPR values.

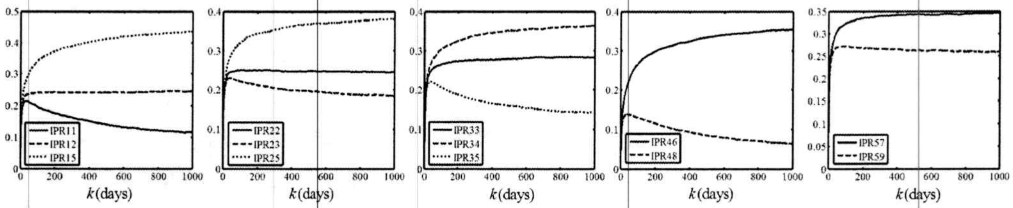


Figure 2.18 Average daily estimated IPR values for the five independent producer-centric models, assuming exact knowledge of the contributing injectors to the producers.

2.3.1 Simulated Data Case Study

First, we generated a simulated oilfield that contains five producers P_1-P_5 and nine injectors I_1-I_9 as depicted in Fig. 2.17, where circles and triangles represent producers and injectors, respectively. Note that 100 Monte Carlo simulations were performed for all simulated tests in this chapter, where the length of each simulated test is D = 1000 days. As indicated by the arrows in Fig. 2.17, P_1-P_3 are influenced by three neighboring injectors, whereas P_4 and P_5 are influenced by only two neighboring injectors. The numerical value on top of each arrow is the fixed IPR value chosen by us for each injector-producer pair. The IPR values for all injector-producer pairs are summarized in Table 2.4. Note that the ICSs and the PCSs of the IPRs are also provided, and none of the ICSs of the IPRs exceeds one, although the ICS for I_3 is very close to one.

We performed two separate tests. In the first test, the EKF was run for each producer-centric SVM, assuming exact knowledge of the contributing injectors to that producer, e.g., only I_2, I_3 and I_5 were included in P_2's SVM, whereas only I_7 and I_9 were included in P_5's SVM. The average estimated IPR values of 100 Monte-Carlo simulations are depicted in Fig. 2.18, and observe that after some time the average estimated IPR curves are reasonably close to their corresponding designed values. The average estimated IPR values at the end of the test (k = 1000) are summarized in Table 2.5. Observe that the PCSs are quite close to their true values (Table 2.4), whereas the ICSs, although not as accurate as the PCSs, are still reasonably accurate.

Many times, in the real world, accurate knowledge of the contributing injectors to a producer is unavailable; thus, in our second test, we ran the EKF for each producer-centric SVM without making the assumption about knowing exactly which injectors affect whichproducers, as we did in our first test. We used a simple heuristic strategy, similar to

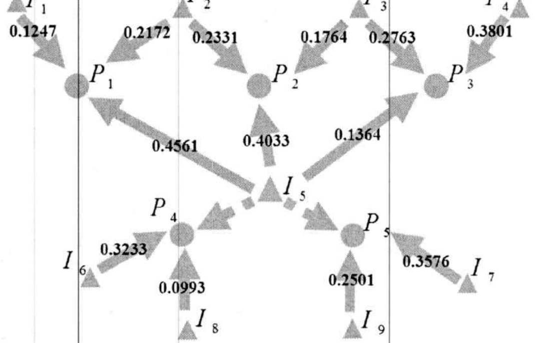


Figure 2.19 Geographic map indicating the three closest injectors included in each of the five independent producer-centric models.

Table 2.4 Designed IPRs for the simulated oilfield in Fig. 2.17

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	PCS[a]
P_1	0.1247	0.2172	0	0	0.4561	0	0	0	0	0.7980
P_2	0	0.2331	0.1764	0	0.4033	0	0	0	0	0.8128
P_3	0	0	0.2763	0.3801	0.1364	0	0	0	0	0.7928
P_4	0	0	0	0	0	0.3233	0	0.0993	0	0.4226
P_5	0	0	0	0	0	0	0.3576	0	0.2501	0.6077
ICS[b]	0.1247	0.4503	0.4527	0.3801	0.9958	0.3233	0.3576	0.0993	0.2501	

a. PCS: Producer-Centric Sum.

b. ICS: Injector-Centric Sum.

Table 2.5 Average estimated IPRs (EKF) for the simulated oilfield, assuming all producers are correctly modeled

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	PCS
P_1	0.1148	0.2454	0	0	0.4355	0	0	0	0	0.7957
P_2	0	0.2451	0.1849	0	0.3802	0	0	0	0	0.8102
P_3	0	0	0.2840	0.3641	0.1424	0	0	0	0	0.7905
P_4	0	0	0	0	0	0.3541	0	0.0643	0	0.4184
P_5	0	0	0	0	0	0	0.3466	0	0.2606	0.6072
ICS	0.1148	0.4905	0.4689	0.3641	0.9581	0.3541	0.3466	0.0643	0.2606	

what is often used in real oilfields, i.e., we selected the three geographically closest injectors to a producer as its contributing injectors. In Fig. 2.19, the injectors having arrows pointing to a producer indicate that they are included in that producer's SVM. Note that P_1-P_3's contributing injectors are correctly modeled as in the first test, whereas P_4 and P_5 have mistakenly included I_5 in their SVMs due to I_5's closeness to them, although I_5 doesn't contribute to them at all. The estimated IPR values at $k = 1000$ are summarized in Table 2.6. Comparing Tables 2.5 and 2.6, the results for P_1-P_3 are exactly the same, so we only focus on the results for P4 and P5, and observe that:

1. The actual contributing injectors for P_4 (I_6 and I_8) have had their estimated IPR values reduced. Similar changes occurred to the actual contributing injectors of P_5 (I_7 and I_9).

2. For P_4, I_5 was allocated approximately the same estimated value as the total IPR reductions from I_6 and I_8. Similarly, for P_5, I_5 was allocated approximately the same estimated value as the total IPR reductions from I_7 and I_9.
3. As a result of the above observation, the PCSs for P_4 and P_5 remain almost the same.
4. Most importantly, in Table 2.6, the ICS of I_5 is greater than one.

To further illustrate some of these observations, let us focus on the specific producer P4, because this can provide us valuable insights into the performance of different methods. In Table 2.5, where P_4 was correctly modeled, the EKF estimated (average) IPR_{48} to be0.0643, but in Table 2.6, where P_4 was incorrectly modeled, (average) IPR_{48} has been reduced to 0.0494 and "pseudo-(average) IPR_{45}" became 0.0485, so it is difficult to tell which injector is actually contributing to P_4. Note, also, that (average) IPR_{46} was reduced

Table 2.6 Average estimated IPRs (EKF) for the simulated oilfield, where each producer includes the three closest injectors.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	PCS
P_1	0.1148	0.2454	0	0	0.4355	0	0	0	0	0.7957
P_2	0	0.2451	0.1849	0	0.3802	0	0	0	0	0.8102
P_3	0	0	0.2840	0.3641	0.1424	0	0	0	0	0.7905
P_4	0	0	0	0	0.0485	0.3212	0	0.0494	0	0.4191
P_5	0	0	0	0	0.0364	0	0.3295	0	0.2416	0.6075
ICS	0.1148	0.4905	0.4689	0.3641	1.0430	0.3212	0.3295	0.0494	0.2416	

Table 2.7 Estimated IPRs between I_{38} and its affected producers in the real field using EKF and different CEKFs.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	ICS
EKF	.137	.118	.100	.090	.126	.060	.086	.042	.146	.049	.179	.105	.166	1.40
MAP	.091	.069	.061	.055	.085	.022	.023	.008	.110	.009	.148	.075	.152	0.91
MS	.102	.079	.063	.055	.087	.026	.049	.007	.111	.011	.143	.071	.145	0.95
$r = 2$	.117	.092	.072	.074	.062	.002	.041	.001	.097	.001	.168	.089	.162	0.98

from 0.3541 to 0.3212, so (average) IPR_{46} and (average) IPR_{48} altogether were reduced by 0.0478 which is very close to the "pseudo-(average) IPR_{45}" of 0.0485.

Our two tests have demonstrated that *an injector assumes a small "pseudo-IPR" value when it is incorrectly included in a producer-centric model, and causes the IPRs of all the actual contributing injectors in that model to be reduced. If this injector is incorrectly included in too many such models, the ICS of this injector can become greater than one.*

2.3.2 Real Data Case Study

Although the simulated tests in Section 2.3.1 demonstrate the problem, the injector-centric sum of I_5 is only slightly greater than one, which might downplay the significance of the problem; but, this is only because our simulated oilfield is quite simple and doesn't have a large number of wells as in real oilfields. It is easy to imagine that, if I_5 was included in many more producer-centric models, its injector-centric sum would grow much greater than one.

To demonstrate this, an injector, labeled I_{38}, was selected from the same real oilfield studied and processed by independent EKFs in Section 2.2. A map of I_{38} (large triangle), its affected producers (circles), and all other injectors included in the producer-centric models (small triangles) are depicted in Fig. 2.20. Observe that I_{38} is included in 13 neighboring producer-centric models, labeled P_1-P_{13}. I_{38}'s estimated IPRs (at k = 1471) with each of its affected producers are summarized in row 1 of Table 2.7. Observe that its ICS is much larger than one; hence, using the unconstrained EKF in Section 2.2 can lead to significant violation of the physical constraint that $ICS \leq 1$.

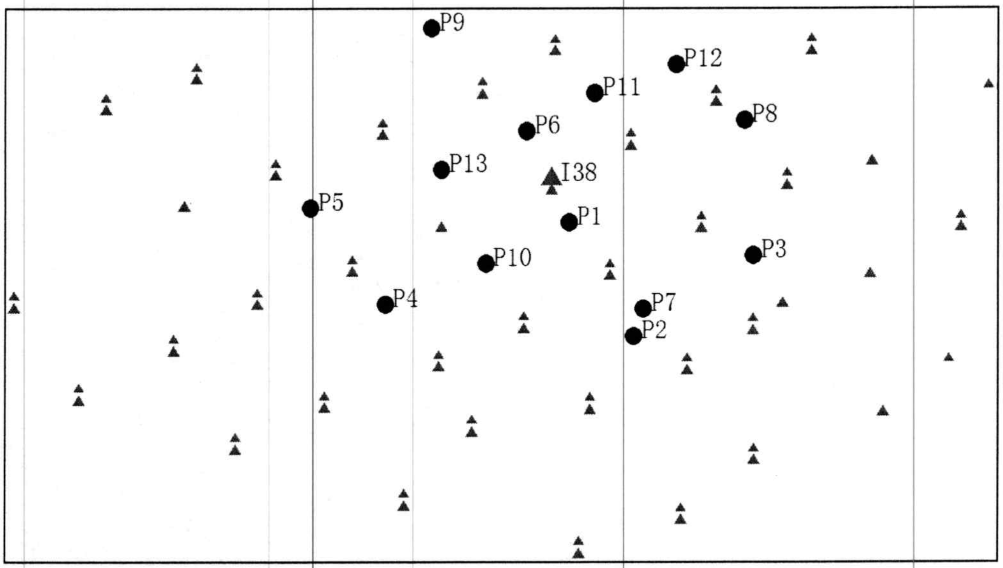


Figure 2.20 Geographic map of I_{38}, its affected producers and all of their contributing injectors.

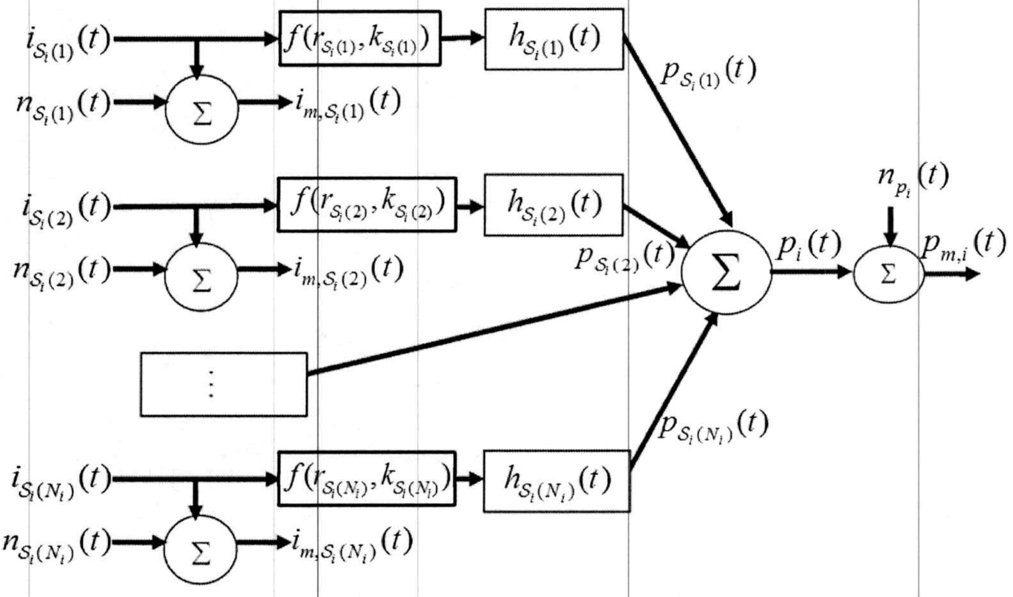

$i_{\mathcal{S}_i(1)}(t)$
$n_{\mathcal{S}_i(1)}(t)$
Σ
$f(r_{\mathcal{S}_i(1)}, k_{\mathcal{S}_i(1)})$
$i_{m,\mathcal{S}_i(1)}(t)$
$h_{\mathcal{S}_i(1)}(t)$
$p_{\mathcal{S}_i(1)}(t)$
$i_{\mathcal{S}_i(2)}(t)$
$n_{\mathcal{S}_i(2)}(t)$
Σ
$f(r_{\mathcal{S}_i(2)}, k_{\mathcal{S}_i(2)})$
$i_{m,\mathcal{S}_i(2)}(t)$
$h_{\mathcal{S}_i(2)}(t)$
$p_{\mathcal{S}_i(2)}(t)$
$\vdots$
Σ
$p_i(t)$
$n_{p_i}(t)$
Σ
$p_{m,i}(t)$
$p_{\mathcal{S}_i(N_i)}(t)$
$i_{\mathcal{S}_i(N_i)}(t)$
$n_{\mathcal{S}_i(N_i)}(t)$
Σ
$f(r_{\mathcal{S}_i(N_i)}, k_{\mathcal{S}_i(N_i)})$
$i_{m,\mathcal{S}_i(N_i)}(t)$
$h_{\mathcal{S}_i(N_i)}(t)$

Figure 2.21 Reservoir Model for a reservoir with N injectors and a single producer.

To further demonstrate the significance of the problem, it is worth mentioning that this particular oilfield had 57 out of 380 injectors that violated the constraint on ICS, and about 20% of those 57 injectors had situations more severe than the one just shown for I_{38}.

2.3.3 Inequality-Constrained State Variable Model

Section 2.2 has laid down the foundation for single producer-centric model. Because imposing cross-producer constraints during the estimation process demands the coupling of multiple producer-centric models, the notations in this Chapter are slightly more intricate. Specifically, a producer-centric reservoir model for P_i (i = 1,..., N_P) that includes N_i independently contributing injectors is depicted in Fig. 2.20, where $i_{\mathcal{S}_i(j)}(t)$, $n_{\mathcal{S}_i(j)}(t)$ and $i_{m,\mathcal{S}_i(j)}(t)$ (j = 1,..., N_i) are the actual injection rate that flows into the reservoir, the corresponding injection rate measurement noise, and the measured injection rate of $\mathcal{S}_i(j)$ (P_i's j^{th} contributing injector), respectively; $p_i(t)$, $n_{p_i}(t)$ and $p_{m,i}(t)$ are the actual production rate, the corresponding production rate measurement noise, and the measured production rate of P_i, respectively; $p_{\mathcal{S}_i(j)}(t)$ represents the amount of production rate in $p_i(t)$ contributed by $\mathcal{S}_i(j)$. It follows (i = 1,..., N_P):

$$\begin{aligned} p_i(t) &= \sum_{j=1}^{N_i} p_{\mathcal{S}_i(j)}(t) \\ &= \sum_{j=1}^{N_i} [f(r_{\mathcal{S}_i(j)}, k_{\mathcal{S}_i(j)}) \times i_{\mathcal{S}_i(j)}(t)] * h_{\mathcal{S}_i(j)}(t) \end{aligned} \tag{2-32}$$

where $f(r_{\mathcal{S}_i(j)}, k_{\mathcal{S}_i(j)})$ are scale functions which show how much of each injection rates flows in the direction toward P_i, and are viewed as a linear or non-linear scalar function of the distance, $r_{\mathcal{S}_i(j)}$, and the permeability, $k_{\mathcal{S}_i(j)}$, between P_i and $\mathcal{S}_i(j)$. Because noise-free data, $i_{\mathcal{S}_i(j)}(t)$ ($j = 1,\ldots, N_i$) and $p_i(t)$, are not directly available, we use their measured values, $i_{m,\mathcal{S}_i(j)}(t)$ ($j = 1,\ldots, N_i$) and $p_{m,i}(t)$, for our data processing.

The two-parameter auto-regressive (AR) model used in Section 2.2 to characterize the impulse response between P_i and $\mathcal{S}_i(j)$ is reproduced below ($i = 1,\ldots, N_P$, $j = 1,\ldots, N_i$):

$$h_{\mathcal{S}_i(j)}(t) = b_{\mathcal{S}_i(j)} t e^{-a_{\mathcal{S}_i(j)} t} \tag{2-33}$$

And, the numerical IPR value between P_i and $\mathcal{S}_i(j)$ has the following expression ($i = 1,\ldots, N_P$, $j = 1,\ldots, N_i$):

$$IPR_{\mathcal{S}_i(j)} = \frac{\gamma_{\mathcal{S}_i(j)} f(r_{\mathcal{S}_i(j)}, k_{\mathcal{S}_i(j)})}{(1-\alpha_{\mathcal{S}_i(j)})^2} = \frac{\gamma_{\mathcal{S}_i(j)}'}{(1-\alpha_{\mathcal{S}_i(j)})^2} \tag{2-34}$$

where $\alpha_{\mathcal{S}_i(j)} = e^{-a_{\mathcal{S}_i(j)} \times T}$, $\gamma_{\mathcal{S}_i(j)} = b_{\mathcal{S}_i(j)} \times \alpha_{\mathcal{S}_i(j)} \times T$ $\gamma_{\mathcal{S}_i(j)} = b_{\mathcal{S}_i(j)} \alpha_{\mathcal{S}_i(j)} T$ and T is the sampling period. Finally, the subsystem between P_i and $\mathcal{S}_i(j)$ can be modeled as:

$$p_{\mathcal{S}_i(j)}(k+1) = 2\alpha_{\mathcal{S}_i(j)}(k) p_{\mathcal{S}_i(j)}(k) - \alpha_{\mathcal{S}_i(j)}^2(k) p_{\mathcal{S}_i(j)}(k-1) + \gamma_{\mathcal{S}_i(j)}' i_{m,\mathcal{S}_i(j)}(k) + n_{p_{\mathcal{S}_i(j)}}(k) \tag{2-35}$$

$$\begin{aligned} p_{\mathcal{S}_i(j)}(k+1) = 2\alpha_{\mathcal{S}_i(j)}(k) p_{\mathcal{S}_i(j)}(k) - \alpha_{\mathcal{S}_i(j)}^2(k) p_{\mathcal{S}_i(j)}(k-1) \\ + IPR_{\mathcal{S}_i(j)}(k)(1-\alpha_{\mathcal{S}_i(j)}(k))^2 i_{m,\mathcal{S}_i(j)}(k) + n_{p_{\mathcal{S}_i(j)}}(k) \end{aligned} \tag{2-36}$$

where $n_{p_{\mathcal{S}_i(j)}}(k)$ is the additive noise. Note that P_i is affected by N_i injectors (see Fig. 2.21); hence, N_i second-order finite-difference-equation models are needed for P_i. Collecting these N_i models into (2-32), a SVM for P_i can be constructed as described next.

2.3.4 A New Modified State Variable Model

In Section 2.2, based on the same reservoir and producer-injector model, different SVMs have been established. The first SVM based on (2-35) directly estimated $\alpha_{S_i(j)}$ and $\gamma'_{S_i(j)}$ so that $IPR_{S_i(j)}$ had to then be computed by (2-34). A second SVM based on (2-36) estimated $\sqrt{IPR_{S_i(j)}}$ and $\sqrt{\alpha_{S_i(j)}}$. It was developed for two major reasons: 1) When $\sqrt{IPR_{S_i(j)}}$ is treated as a state variable, the EKF not only provides an estimate of $\sqrt{IPR_{S_i(j)}}$, but it also provides its error (pseudo-) variance, which can be used to calculate a numerical upper and lower bound for the estimated $\sqrt{IPR_{S_i(j)}}$; and 2) It is physically meaningful to only have nonnegative-valued $\alpha_{S_i(j)}$, $\gamma'_{S_i(j)}$ and $IPR_{S_i(j)}$. Due to strong noise and other uncertainty factors in the real data, the first SVM sometimes produced negative-valued $\alpha_{S_i(j)}$ and $\gamma'_{S_i(j)}$. The second SVM overcame this problem because, regardless of the sign of the estimated $\sqrt{IPR_{S_i(j)}}$ and $\sqrt{\alpha_{S_i(j)}}$, their squared values always give positive-valued $IPR_{S_i(j)}$ and $\alpha_{S_i(j)}$.

In this section, we propose a third SVM based on (2-36) to directly estimate $IPR_{S_i(j)}$ and $\alpha_{S_i(j)}$, because all constraints on the estimated parameters during the estimation processing are then linear, whereas the constraints, when $\sqrt{IPR_{S_i(j)}}$ and $\sqrt{\alpha_{S_i(j)}}$ are estimated, are nonlinear.

For the new SVM, the 4 × 1 state vector between P_i and $S_i(j)$, $\boldsymbol{x}_{S_i(j)}(k)$, is:

$$\begin{aligned}\boldsymbol{x}_{S_i(j)}(k) &= [x_{S_i(j),1}(k) \quad x_{S_i(j),2}(k) \quad x_{S_i(j),3}(k) \quad x_{S_i(j),4}(k)]^T \\ &= [IPR_{S_i(j)}(k) \quad \alpha_{S_i(j)}(k) \quad p_{S_i(j)}(k-1) \quad p_{S_i(j)}(k)]^T\end{aligned} \qquad (2\text{-}37)$$

A SVM for P_i can be established ($i = 1,\ldots, M$), using (2-36) and (1), as:

$$\boldsymbol{x}_i(k+1) = \begin{bmatrix} \boldsymbol{x}_{\mathcal{S}_i(1)}(k+1) \\ \boldsymbol{x}_{\mathcal{S}_i(2)}(k+1) \\ \vdots \\ \boldsymbol{x}_{\mathcal{S}_i(N_i)}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\mathcal{S}_i(1)}(\boldsymbol{x}_{\mathcal{S}_i(1)}(k)) + \boldsymbol{n}_{\boldsymbol{x}_{\mathcal{S}_i(1)}}(k) \\ \boldsymbol{f}_{\mathcal{S}_i(2)}(\boldsymbol{x}_{\mathcal{S}_i(2)}(k)) + \boldsymbol{n}_{\boldsymbol{x}_{\mathcal{S}_i(2)}}(k) \\ \vdots \\ \boldsymbol{f}_{\mathcal{S}_i(N_i)}(\boldsymbol{x}_{\mathcal{S}_i(N_i)}(k)) + \boldsymbol{n}_{\boldsymbol{x}_{\mathcal{S}_i(N_i)}}(k) \end{bmatrix} = \boldsymbol{f}_i(\boldsymbol{x}_i(k)) + \boldsymbol{n}_{\boldsymbol{x}_i}(k) \tag{2-38}$$

$$\begin{aligned} p_i(k+1) &= p_{\mathcal{S}_i(1)}(k+1) + p_{\mathcal{S}_i(2)}(k+1) + \cdots + p_{\mathcal{S}_i(N_i)}(k+1) + n_{p_i}(k+1) \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \times \boldsymbol{x}_i(k+1) + n_{p_i}(k+1) \\ &= \boldsymbol{H}_i \boldsymbol{x}_i(k+1) + n_{p_i}(k+1) \end{aligned} \tag{2-39}$$

Equations (2-38) and (2-39) are our state equation and measurement equation, respectively, and, $\boldsymbol{n}_{\boldsymbol{x}_i}(k) = [\boldsymbol{n}_{\boldsymbol{x}_{S_i(1)}} \; \boldsymbol{n}_{\boldsymbol{x}_{S_i(2)}} \; \cdots \; \boldsymbol{n}_{\boldsymbol{x}_{S_i(N_i)}}]^T$ and $n_{p_i}(k+1)$ are additive zero-mean white noises with $4N_i \times 4N_i$ block diagonal covariance matrix $\boldsymbol{Q}_{\boldsymbol{x}_i}(k)$ and variance $r_{p_i}(k+1)$, respectively. It follows from (2-36) and (2-37) that each $\boldsymbol{x}_{S_i(j)}(k+1)$ in (2-38) is described by the following four-state model ($j = 1,\ldots, N_i$):

$$\begin{aligned} \boldsymbol{x}_{\mathcal{S}_i(j)}(k+1) &= \boldsymbol{f}_{\mathcal{S}_i(j)}(\boldsymbol{x}_{\mathcal{S}_i(j)}(k)) + \boldsymbol{n}_{\boldsymbol{x}_{\mathcal{S}_i(j)}}(k) \\ &= \begin{bmatrix} x_{\mathcal{S}_i(j),1}(k) \\ x_{\mathcal{S}_i(j),2}(k) \\ x_{\mathcal{S}_i(j),4}(k) \\ 2x_{\mathcal{S}_i(j),2}(k)x_{\mathcal{S}_i(j),4}(k) - x^2_{\mathcal{S}_i(j),2}(k)x_{\mathcal{S}_i(j),3}(k) \\ \quad + x_{\mathcal{S}_i(j),1}(k)(1 - x_{\mathcal{S}_i(j),2}(k))^2 i_{m,\mathcal{S}_i(j)}(k) \end{bmatrix} + \boldsymbol{n}_{\boldsymbol{x}_{\mathcal{S}_i(j)}}(k) \end{aligned} \tag{2-40}$$

where $\boldsymbol{n}_{\boldsymbol{x}_{S_i(j)}} = [n_{IPR_{S_i(j)}}(k) \;\; n_{\alpha_{S_i(j)}}(k) \;\; 0 \;\; n_{p_{S_i(j)}}(k)]^T$.

The EKF processing of the above SVM provides us with $\hat{\boldsymbol{x}}_i(k+1|k+1)$, which estimates $\boldsymbol{x}_i(k+1)$ based on measurements up to and including time $k+1$ [$p_{m,i}(k+$

1) $p_{m,i}(k) \ \cdots \ p_{m,i}(1)]$, and the pseudo-error-covariance matrix of $\hat{\boldsymbol{x}}_i(k+1|k+1)$, $\boldsymbol{P}_i(k+1|k+1)$.

2.3.5 Cross-Producer State Variable Model

In this section, we explain how to impose constraints on some of the states in our SVM during the EKF processing. We, again, assume N_P producers, $P_1,\ldots, P_{N_P}$, and N_I injectors, $I_1,\ldots, I_{N_I}$. The constraints on each IPR and α values occur within each producer-centric model, whereas the constraints on injector-centric sums of the IPRs are across multiple producer-centric models. Consequently, all producer-centric models have to be coupled together into a complete *cross-producer* SVM. This can be achieved by combining the state vectors of all the producer-centric SVMs to form the following augmented state vector:

$$\boldsymbol{x}(k) = \begin{bmatrix} \boldsymbol{x}_1(k)^T & \boldsymbol{x}_2(k)^T & \cdots & \boldsymbol{x}_M(k)^T \end{bmatrix}^T \tag{2-41}$$

The resulting cross-producer SVM is:

$$\begin{aligned} \boldsymbol{x}(k+1) = \begin{bmatrix} \boldsymbol{x}_1(k+1) \\ \boldsymbol{x}_2(k+1) \\ \vdots \\ \boldsymbol{x}_{N_P}(k+1) \end{bmatrix}_{\sum_{i=1}^{M} 4N_i \times 1} &= \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{x}_1(k),k] \\ \boldsymbol{f}_2[\boldsymbol{x}_2(k),k] \\ \vdots \\ \boldsymbol{f}_{N_P}[\boldsymbol{x}_{N_P}(k),k] \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_{x_1}(k) \\ \boldsymbol{n}_{x_2}(k) \\ \vdots \\ \boldsymbol{n}_{x_{N_P}}(k) \end{bmatrix} \\ &= \boldsymbol{f}_{\text{cp}}[\boldsymbol{x}(k),k] + \boldsymbol{n}_x(k) \end{aligned} \tag{2-42}$$

and

$$\boldsymbol{p}(k+1)=\begin{pmatrix} p_1(k+1) \\ p_2(k+1) \\ \vdots \\ p_M(k+1) \end{pmatrix}=\begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{H}_{N_P} \end{pmatrix}\begin{pmatrix} \boldsymbol{x}_1(k+1) \\ \boldsymbol{x}_2(k+1) \\ \vdots \\ \boldsymbol{x}_{N_P}(k+1) \end{pmatrix}+\begin{pmatrix} n_{p_1}(k+1) \\ n_{p_2}(k+1) \\ \vdots \\ n_{p_{N_P}}(k+1) \end{pmatrix}$$
$$=\boldsymbol{H}_{cp}\boldsymbol{x}(k+1)+\boldsymbol{n}_p(k+1) \tag{2-43}$$

2.3.6 Inequality-Constrained State Vectors

In this section, all the constraints are first formulated as linear inequalities in terms of $\boldsymbol{x}(k)$.

Because the first and second component of each $\boldsymbol{x}_{\mathcal{S}_i(j)}(k)$ are $IPR_{\mathcal{S}_i(j)}(k)$ and $\alpha_{\mathcal{S}_i(j)}(k)$, respectively [see (2-37)], the constraints that all the individual *IPR* and α values, at any time point k, must fall in the interval [0, 1] can be expressed as ($\forall i = 1, \dots, N_P$, $\forall j = 1, \dots, N_i$ and $\forall l = 1,2$):

$$0 \leq x_{\mathcal{S}_i(j),l}(k) \leq 1 \tag{2-44}$$

This set of inequalities can be written in terms of $\boldsymbol{x}(k)$, as:

$$\boldsymbol{0} \leq \boldsymbol{C}_1\boldsymbol{x}(k)=\begin{pmatrix} \boldsymbol{B}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B}_{N_P} \end{pmatrix}\begin{pmatrix} \boldsymbol{x}_1(k) \\ \boldsymbol{x}_2(k) \\ \vdots \\ \boldsymbol{x}_{N_P}(k) \end{pmatrix} \leq \boldsymbol{1} \tag{2-45}$$

where $\boldsymbol{B}_i$ is $N_i \times 4N_i$ $(i = 1, \dots, N_P)$, as:

$$\boldsymbol{B}_i = \left(\begin{array}{cccc|cccc|c|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & \ddots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \end{array}\right) \tag{2-46}$$

The constraints that the injector-centric sums of the IPR values, at any time point k, must also all fall in the interval [0,1] can be written in terms of $\boldsymbol{x}(k)$, as:

$$\mathbf{0} \le \boldsymbol{C}_2 \boldsymbol{x}(k) = \begin{pmatrix} \boldsymbol{d}_{11} & \boldsymbol{d}_{12} & \cdots & \boldsymbol{d}_{1,N_P} \\ \boldsymbol{d}_{21} & \boldsymbol{d}_{22} & \ddots & \boldsymbol{d}_{2,N_P} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{d}_{N,1} & \boldsymbol{d}_{N,2} & \cdots & \boldsymbol{d}_{N_P,N_P} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1(k) \\ \boldsymbol{x}_2(k) \\ \vdots \\ \boldsymbol{x}_M(k) \end{pmatrix} \le \mathbf{1} \tag{2-47}$$

where ($n = 1, \dots, N_I$ and $i = 1, \dots, N_P$):

$$\boldsymbol{d}_{n,i} = \begin{bmatrix} d_{n,i,1} & 0 & 0 & 0 & d_{n,i,2} & 0 & 0 & 0 & \cdots & d_{n,i,N_i} & 0 & 0 & 0 \end{bmatrix}; \tag{2-48}$$

and ($j = 1, \dots, N_i$):

$$d_{n,i,j} = \begin{cases} 1, \text{ if } I_n = \mathcal{S}_i(j) \\ 0, \text{ if } I_n \ne \mathcal{S}_i(j) \end{cases} \tag{2-49}$$

The two sets of constraints in (2-45) and (2-47) are combined as:

$$\mathbf{0} \le \boldsymbol{C}\boldsymbol{x}(k) = \begin{pmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \end{pmatrix} \boldsymbol{x}(k) \le \mathbf{1} \tag{2-50}$$

2.3.7 Inequality-Constrained State Estimation

Simon, et al. [163-165] developed the linear state-equality/inequality constrained methods for the Kalman Filter. In this section, we generalize those methods to the EKF using our cross-producer SVM [(2-42) and (2-43)] that is subject to (2-50).

There are several variants of the Constrained EKF (CEKF) method. We first provide its most general form, and then look at two of its special cases.

The General Projection CEKF (GP CEKF) approach is: Given $\underline{\hat{x}}(k+1|k+1)$ and a symmetric positive definite weighting matrix *W*, compute a new estimate $\underline{\overline{x}}(k+1)$, such that

$$\begin{aligned}&\underset{\bar{\boldsymbol{x}}(k+1)}{\arg\min}\left\{[\bar{\boldsymbol{x}}(k+1)-\hat{\boldsymbol{x}}(k+1\,|\,k+1)]^T \times \boldsymbol{W} \times [\bar{\boldsymbol{x}}(k+1)-\hat{\boldsymbol{x}}(k+1\,|\,k+1)]\right\}\\ &\text{such that: } \mathbf{0} \le \boldsymbol{C}\ \bar{\boldsymbol{x}}(k+1) \le \mathbf{1}\end{aligned} \tag{2-51}$$

The reasoning behind (2-51) is: if the standard EKF estimator $\hat{\boldsymbol{x}}(k+1|k+1)$ doesn't satisfy (2-50), we look for another estimator, $\overline{\boldsymbol{x}}(k+1)$, that is as close to $\hat{\boldsymbol{x}}(k+1|k+1)$ as possible while at the same time satisfying (2-50). Note that if $\hat{\boldsymbol{x}}(k+1|k+1)$ satisfies (2-50), $\overline{\boldsymbol{x}}(k+1)$ is set equal to $\hat{\boldsymbol{x}}(k+1|k+1)$, i.e., the CEKF estimator $\overline{\boldsymbol{x}}(k+1)$ will be the same as the EKF estimator $\hat{\boldsymbol{x}}(k+1|k+1)$.

Equation (2-51) is a standard quadratic programming problem subject to a set of linear inequality constraints. Such a problem can be solved by many existing algorithms, e.g., Active Set Algorithms [136].

A special case of (2-51) is called a Maximum Probability CEKF (MAP CEKF) approach, and is formulated by setting $\boldsymbol{W} = \boldsymbol{P}(k+1|k+1)^{-1}$, i.e.: Given $\hat{\boldsymbol{x}}(k+1|k+1)$ and $\boldsymbol{P}(k+1|k+1)^{-1}$, compute a new estimate $\underline{\overline{x}}(k+1)$, such that:

$$\begin{aligned}&\arg\min_{\overline{\boldsymbol{x}}(k+1)}\left\{[\overline{\boldsymbol{x}}(k+1)-\hat{\boldsymbol{x}}(k+1|k+1)]^T \times \boldsymbol{P}(k+1|k+1)^{-1} \times [\overline{\boldsymbol{x}}(k+1)-\hat{\boldsymbol{x}}(k+1|k+1)]\right\}\\ &\text{such that: } \mathbf{0} \le \boldsymbol{C}\ \overline{\boldsymbol{x}}(k+1) \le \mathbf{1}\end{aligned} \tag{2-52}$$

Another special case of (2-51) is called Mean Square CEKF (MS CEKF), and is formulated by setting $\boldsymbol{W} = \boldsymbol{I}$, i.e.: Given $\hat{\boldsymbol{x}}(k+1|k+1)$, compute a new estimate $\overline{\boldsymbol{x}}(k+1)$, such that:

$$\begin{aligned}&\arg\min_{\overline{\boldsymbol{x}}(k+1)}\left\{[\overline{\boldsymbol{x}}(k+1)-\hat{\boldsymbol{x}}(k+1|k+1)]^T \times [\overline{\boldsymbol{x}}(k+1)-\hat{\boldsymbol{x}}(k+1|k+1)]\right\}\\ &\text{such that: } \mathbf{0} \le \boldsymbol{C}\ \overline{\boldsymbol{x}}(k+1) \le \mathbf{1}\end{aligned} \tag{2-53}$$

To summarize, the following *CEKF Procedure I* is presented: Beginning with $k = 0$,

1. Apply the standard EKF [(2.22)-(2.26)] to the cross-producer model [(2-42) and (2-43)] to obtain $\hat{\boldsymbol{x}}(k+1|k+1)$.
2. Check if $\hat{\boldsymbol{x}}(k+1|k+1)$ violates the constraints: if so, go to Step 3; if not, go to Step 5.
3. Given a weighting matrix $\boldsymbol{W}$, (2-51) is solved to obtain $\overline{\boldsymbol{x}}(k+1)$.
4. $\hat{\boldsymbol{x}}(k+1|k+1)$ is set equal to $\overline{\boldsymbol{x}}(k+1)$.
5. Set $k = k+1$, and return to Step 1 (until the end of the data).

It's worth noting that the coupling of the states of different producer-centric models occurs during the constrained optimization process (Step 3), which is why it's called *constraint coupling*.

2.3.8 Case Study

To assess the performance of the MAP and MS CEKF methods, and to gain some insights into the design of an even better weighting matrix, the two methods are first applied to the simulated oilfield studied in Section 2.3.1. As in Section 2.3.1, we ran 100 Monte-Carlo simulations. The resulting average IPR values at k = 1000 are summarized in Tables 2.8 and 2.9 for the MAP and MS CEKF, respectively. Observe their results are quite close, which indicates one can either use the MAP or MS CEKF. Therefore, in the rest of this section, we focus only on MS CEKF results (Table 2.9). Comparing them against the EKF results (Table 2.6), observe that: 1) The IPR values between I_5 and all producers have been reduced such that the ICS of I_5 is less than one; 2) In each producer model, the IPRs of all other injectors have increased by a total amount that is approximately the same as the amount that the IPR of I_5 has been reduced by.

Table 2.8 Estimated IPRs (MAP CEKF) for the simulated oilfield, where each producer includes the three closest injectors.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	PCS
P_1	0.1235	0.2523	0	0	0.4198	0	0	0	0	0.7956
P_2	0	0.2519	0.1929	0	0.3653	0	0	0	0	0.8101
P_3	0	0	0.2931	0.3717	0.1257	0	0	0	0	0.7905
P_4	0	0	0	0	0.0389	0.3262	0	0.0540	0	0.4191
P_5	0	0	0	0	0.0274	0	0.3336	0	0.2464	0.6074
ICS	0.1235	0.5042	0.4860	0.3717	0.9771	0.3262	0.3336	0.0540	0.2464	

Table 2.9 Estimated IPRs (MS CEKF) for the simulated oilfield, where each producer includes the three closest injectors.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	PCS
P_1	0.1214	0.2510	0	0	0.4221	0	0	0	0	0.7945
P_2	0	0.2506	0.1911	0	0.3673	0	0	0	0	0.8090
P_3	0	0	0.2914	0.3707	0.1273	0	0	0	0	0.7894
P_4	0	0	0	0	0.0380	0.3260	0	0.0540	0	0.4180
P_5	0	0	0	0	0.0269	0	0.3336	0	0.2461	0.6066
ICS	0.1214	0.5016	0.4825	0.3707	0.9816	0.3260	0.3336	0.0540	0.2461	

The above IPR changes are *positive* for P_4 and P_5. To explain this, let us focus on P_4 again, and observe, in Tables 2.8 and 2.9, that: CEKF has forced the "pseudo-IPR_{45}" to be reduced from 0.0485 to 0.0380, an amount that is shared by I_6 and I_8. Specifically, IPR_{48} increases from 0.0494 to 0.0540, so it is clearer that I_8 is more likely to be contributing to P_4 than is I_5.

On the other hand, the above IPR changes are *negative* for P_1-P_3, because: P_1-P_3 all have their contributing injectors correctly modeled, so Table 2.6 has already provided thebest possible estimates using the EKF methods. But, in Table 2.9, it can be observed that the average estimates of IPR_{15}- IPR_{35} have become smaller than their best estimates, and conversely, all other average IPR estimates in P_1-P_3 have become greater than their best estimates.

Consequently, we observe that *the CEKF guarantees the constraints on the state vectors are met, and produces more accurate IPR values for producers that are incorrectly modeled, but it negatively biases IPR values for producers that are correctly modeled.*

The MAP and MS CEKF methods were applied to the real oilfield studied in Section 2.3.2, and the resulting IPR values of I_{38} (at k = 1471) are summarized in rows 2 and 3 of Table 2.7, respectively. The results for both methods are quite close, and, compared to those in row 1, all the IPRs have been reduced so that their ICS is less than one.

2.3.9 Reduced-Length Constrained Extended Kalman Filter

To reduce the negative biases that the current CEKF introduced into the correctly modeled producers, a better weighting matrix plays the key role; but, designing such a weighting matrix is difficult for the current CEKF structure, because the weighting matrix has to account for every state variable in $\boldsymbol{x}(k)$. In this section, we show how this full-length CEKF can be reduced to a reduced-length CEKF, that only involves the constrained IPR state variables, and how a better weighting matrix can be designed for the reduced-length CEKF.

Let $\boldsymbol{x}_{IPR}(k)$ ($\sum_{i=1}^{N_P} N_i \times 1$) be composed of those state variables in $\boldsymbol{x}(k)$ that represent just the IPRs, i.e.:

$$\begin{aligned}&\boldsymbol{x}_{IPR}(k) = \\ &\Big[x_{\mathcal{S}_1(1),1}(k) \quad x_{\mathcal{S}_1(2),1}(k) \quad \cdots \quad x_{\mathcal{S}_1(N_1),1}(k) \quad x_{\mathcal{S}_2(1),1}(k) \quad x_{\mathcal{S}_2(2),1}(k) \quad \cdots \quad x_{\mathcal{S}_2(N_2),1}(k) \quad \cdots \\ &\qquad \cdots \quad x_{\mathcal{S}_M(1),1}(k) \quad x_{\mathcal{S}_M(2),1}(k) \quad \cdots \quad x_{\mathcal{S}_M(N_M),1}(k)\Big]\end{aligned} \tag{2-54}$$

The constraints on the individual and the sum of the IPR values [(2-45) and (2-47)] can be re-expressed in terms of $\boldsymbol{x}_{IPR}(k)$:

$$\mathbf{0} \le \boldsymbol{x}_{IPR}(k) \le \mathbf{1} \tag{2-55}$$

$$\mathbf{0} \le \boldsymbol{C}_{sum}\boldsymbol{x}_{IPR}(k) = \begin{pmatrix} \boldsymbol{e}_{11} & \boldsymbol{e}_{12} & \cdots & \boldsymbol{e}_{1M} \\ \boldsymbol{e}_{21} & \boldsymbol{e}_{22} & \cdots & \boldsymbol{e}_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{e}_{N1} & \boldsymbol{e}_{N2} & \cdots & \boldsymbol{e}_{NM} \end{pmatrix} \boldsymbol{x}_{IPR}(k) \le \mathbf{1} \tag{2-56}$$

where ($n = 1,\ldots, N_I$ and $i = 1,\ldots, N_P$):

$$\boldsymbol{e}_{n,i} = [d_{n,i,1} \quad d_{n,i,2} \quad \cdots \quad d_{n,i,N_i}] \tag{2-57}$$

and $d_{n,i,j}$ ($j = 1,\ldots, N_i$) have been defined in (2-49). Consequently, the complete constraint on the IPR values [(2.50)] can be re-expressed as:

$$\boldsymbol{0} \le \boldsymbol{C}_{IPR}\boldsymbol{x}_{IPR}(k) = \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{C}_{sum} \end{pmatrix} \boldsymbol{x}_{IPR}(k) \le \boldsymbol{1} \tag{2-58}$$

We now let $\boldsymbol{W}_{IPR}$ be a weighting matrix just for $\boldsymbol{x}_{IPR}(k)$. Extensive experiments have shown that, without losing performance, Procedure I can be replaced by the following CEKF Procedure II, which significantly reduces the dimensionality of the CEKF: Beginning with $k = 0$,

1. Same as Step 1 in Procedure I.
2. Same as Step 2 in Procedure I.
3. Extract $\hat{\boldsymbol{x}}_{IPR}(k+1|k+1)$ from $\hat{\boldsymbol{x}}(k+1|k+1)$ and solve the following problem to obtain $\overline{\boldsymbol{x}}_{IPR}(k)$:

$$\begin{aligned}&\underset{\overline{\boldsymbol{x}}_{IPR}(k+1)}{\arg\min}\left\{[\overline{\boldsymbol{x}}_{IPR}(k+1) - \hat{\boldsymbol{x}}_{IPR}(k+1|k+1)]^T \times \boldsymbol{W}_{IPR} \times [\overline{\boldsymbol{x}}_{IPR}(k+1) - \hat{\boldsymbol{x}}_{IPR}(k+1|k+1)]\right\} \\ &\text{such that: } \boldsymbol{0} \le \boldsymbol{C}_{IPR}\overline{\boldsymbol{x}}_{IPR}(k+1) \le \boldsymbol{1}\end{aligned} \tag{2-59}$$

4. Return the elements of $\overline{\boldsymbol{x}}_{IPR}(k+1)$ to corresponding places in $\hat{\boldsymbol{x}}(k+1|k+1)$.
5. Set $k = k+1$, and return to Step 1 (until the end of the data).

In our previous tests using CEKF Procedure I, we observed that diagonal weighting matrix $\boldsymbol{W} = \boldsymbol{I}$ and weighting matrix $\boldsymbol{W} = \boldsymbol{P}(k+1|k+1)^{-1}$ produced quite close results;

therefore, it seems reasonable to focus on the simpler diagonal weighting matrix for Procedure II.

In this section, we use a diagonal matrix for $\boldsymbol{W}_{IPR}$, and assign heavier weight to those diagonal weighting coefficients that correspond to large-valued estimated IPR values and a lighter weight to those that correspond to small-valued estimated IPRs. Our logic for this is, when an injector contributes a large-valued IPR to a producer, it is more likely that this injector does indeed contribute to that producer; on the other hand, when an injector contributes a small-valued IPR to a producer, it is less likely that this injector contributes to that producer. Assigning large (small) weighting coefficients to large (small) IPR values helps to guarantee that, when trying to satisfy the constraints in (2-59), more- (less-) likely contributing injectors maintain (reduce) their corresponding IPR values.

Consequently, our proposed weighting matrix is chosen to be diagonal where each diagonal element is chosen to be the r^{th} power of the corresponding estimated IPR value, i.e.:

$$\boldsymbol{W}_{IPR} = \begin{pmatrix} \boldsymbol{W}_{IPR,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_{IPR,2} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{W}_{IPR,M} \end{pmatrix} \tag{2-60}$$

where $\boldsymbol{W}_{IPR,i}$ $(i = 1,\ldots, M)$ is:

$$\boldsymbol{W}_{IPR,i} = \begin{pmatrix} \hat{\boldsymbol{x}}^{r}_{\mathcal{S}_i(1),1}(k+1\,|\,k+1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \hat{\boldsymbol{x}}^{r}_{\mathcal{S}_i(2),1}(k+1\,|\,k+1) & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \hat{\boldsymbol{x}}^{r}_{\mathcal{S}_i(N_i),1}(k+1\,|\,k+1) \end{pmatrix} \tag{2-61}$$

Table 2.10 Estimated IPRs (Designed CEKF with r = 2) for the simulated oilfield, where each producer includes the three closest injectors.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	PCS
P_1	0.1155	0.2460	0	0	0.4342	0	0	0	0	0.7957
P_2	0	0.2457	0.1857	0	0.3788	0	0	0	0	0.8102
P_3	0	0	0.2864	0.3662	0.1371	0	0	0	0	0.7897
P_4	0	0	0	0	0.0151	0.3357	0	0.0648	0	0.4156
P_5	0	0	0	0	0.0197	0	0.3364	0	0.2492	0.6053
ICS	0.1155	0.4917	0.4621	0.3662	0.9849	0.3357	0.3364	0.0648	0.2492	

In this way, large (small) estimated IPR values automatically have large (small) weighting coefficients. The specific value of r was chosen to be two, by trial and error, on our simulated oilfield. The average estimated daily IPRs of 100 Monte-Carlo simulations are depicted in Fig. 2.9, and their values at k = 1000 are summarized in Table 2.10. Comparing Table 2.10 to previous CEKF tests (Tables 2.8 and 2.9), observe that the IPRs between I_5 and P_1-P_3 have only been reduced by small amounts, whereas the "pseudo-IPR" values between I_5 and P_4-P_5 have been reduced by much larger amounts, e.g., comparing the results of P_4 in Tables 2.9 and 2.10, observe that: "Pseudo-IPR_{45}" has been reduced significantly from 0.0380 to 0.0151, and the reduced amount has been re-allocated to I_6 and I_8. Specifically, IPR_{48} has increased from 0.0540 to 0.0648 which is very close to the result of 0.0643 when P_4 is correctly modeled (Table 2.5).

Among all our simulated tests, the numerical results in Table 2.10 are the closest to those in Table 2.5, where all producers are correctly modeled.

Procedure II, using W_{IPR} with r = 2, was also applied to our real oilfield, and the resulting IPRs of I_{38} (at k = 1471) are summarized in row 4 of Table 2.7. Observe, in Table 2.7, that large (small) IPRs were reduced by relatively smaller (larger) amounts, which supports our analysis.

2.4 Robust Production Rate Interpolation

In this section, we examine how often the production rate is measured in practice. A small pilot area selected from a real oilfield is depicted in Fig. 2.22. From a sampling frequency perspective, the producers in this pilot area can be divided into two groups: *typical producers* and *extreme producers*.

The majority of the producers belong to the typical category; an example of the complete production history of a typical producer is depicted in Fig. 2.23. The usual sampling frequency of this producer can be observed as we zoom in to a local window (day 410 ~ day 540) shown in Fig. 2.24(a); it is on average one measurement every eight to ten days. However, as can be seen in Fig. 2.24(b) (a local window from day 285 ~ day 430 for the same producer), there are also cases in the production history where the sampling interval could be every large, e.g., observe that no measurement is available from day 347 to day 416.

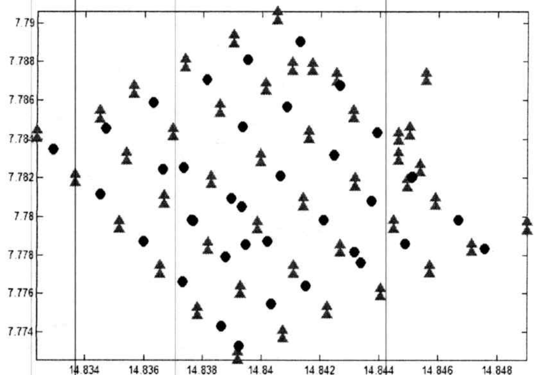


Figure 2.22 A pilot area selected from a real oilfield.

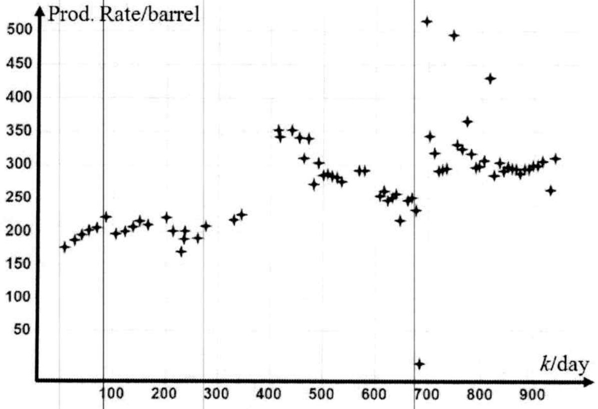


Figure 2.23 Complete production history of a typical producer in the pilot area shown in Fig. 4.1.

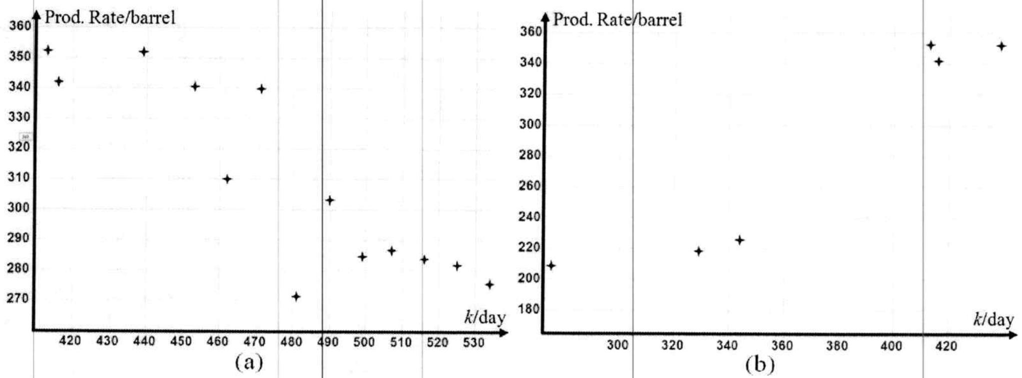


Figure 2.24 (a) a local window (day 410 ~ day 540) of the production history that shows the usual sampling frequency of the production rate (one measurement every eight ~ ten days); (b) a local window (day 285 ~ day 430) of the production history that shows some extreme sampling intervals of the production rate, e.g., there is no measurement available from day 347 to day 416.

On the other hand, a small number of producers fall into the extreme category: throughout the entire production history, only very limited amount of measurements were taken for such producers. An example of the production history of an extreme producer, which only has nine measurements, is depicted in Fig. 2.25.

2.4.1 Intermittent Measurement Analysis

To study how the intermittent production rate might affect the performance of our EKF estimation, we first construct a single producer-centric model with three contributing injectors as depicted in Fig. 2.26, where a constant IPR value has been specified for each injector-producer pair. It should be noted that the average of these three IPR values is$(0.2172 + 0.4561 + 0.0993)/3 = 0.2575$.

For each injector, its injection rate at every time point is modeled as an independently and uniformly distributed random variable with mean 300 bpd and standard deviation 70 bpd. The injection rates of these three injectors are then fed into the Liu-Mendel Model to generate the gross production rates for the producer; and then 10% Gaussian measurement noise are added to the production rates. A window (day 1250 ~ day 1320) of the average production rate of 100 Monte-Carlo simulations is depicted by the solid line in Fig. 2.27. The average IPR curves of the three injectors based on 100 Monte-Carlo simulations of our EKF strategy proposed in Section 2.2 (no CEKF needs to be involved here as there is only one producer) using full set of production measurements are depicted in 2.28; and, observe that these curves gradually converge to their correct designed values, respectively.

On the other hand, assuming one measurement is available every eight days, the average sampled production rate is depicted by the discrete dots in Fig. 2.27. Now, we want to examine how the estimation results might be influenced if such intermittent measurements are used directly without any interpolation, namely, during the estimation process, we perform both predictor and corrector steps like a standard EKF for those time points where a measurement is available, but only perform the predictor step for all other time points where there is no measurement at all (such time points are also referred to as missing-measurement time points).

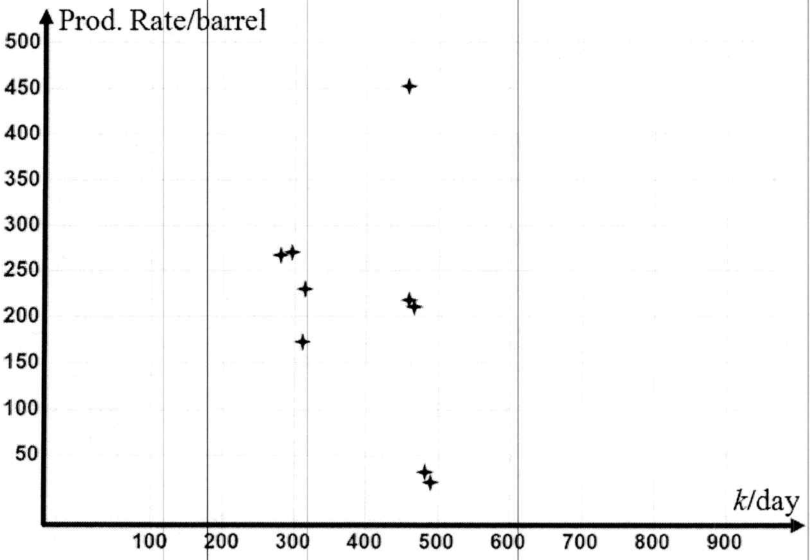


Figure 2.25 An example of an extreme producer that only has nine measurements.

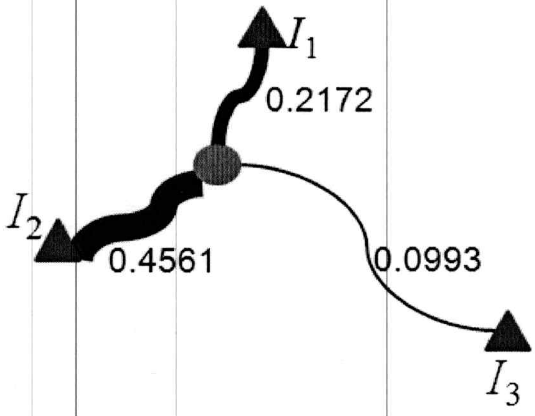


Figure 2.26 A single producer-centric test case with three contributing injectors.

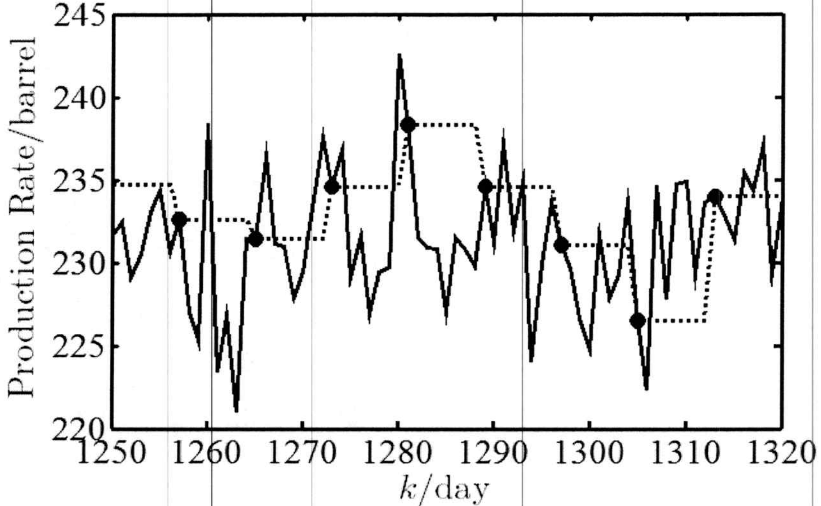


Figure 2.27 Average actual (solid line), sampled (discrete dots) and Zero-Order-Held (dotted line) production rate of 100 Monte-Carlo Simulations based on the test case in Fig. 2.25.

To be mathematically specific, let D denote the total number of time points, let M $(M < D)$ denote the total number of measurements available to us, and let k_i and k_{i+1} $(1 \leq i \leq M-1)$ denote the time points where the i^{th} and i+1$^{\text{th}}$ measurements are taken, respectively. In Fig. 2.28, k_i, k_{i+1} and the missing-measurement time points between them, k_i+1, k_i+2,..., and $k_{i+1}-1$, are labeled along the time line. The measurement vector at k_{i+1}, $\boldsymbol{Z}(k_{i+1})$, is the stack of all previously and currently available measurements, i.e.:

$$\boldsymbol{Z}(k_{i+1})=\left[z(k_{i+1})^T \quad z(k_i)^T \quad \cdots \quad z(k_1)^T\right]^T \tag{2-62}$$

Without loss of generality, assume we are at time point k_i and are given the state estimation results, $\hat{\boldsymbol{x}}(k_i|k_i)$ and $\boldsymbol{P}(k_i|k_i)$; after this point, only predictor step can be applied for time points k_i+1, k_i+2,..., $k_{i+1}-2$ and $k_{i+1}-1$ until a new measurement become available at time point k_{i+1} (see Fig. 2.29). The predictor equations for the missing-measurement time points after k_i are given as follows:

$$\begin{cases} \hat{\boldsymbol{x}}(k_i+1|k_i)=\boldsymbol{f}[\hat{\boldsymbol{x}}(k_i|k_i)] \\ \boldsymbol{P}(k_i+1|k_i)=\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_i+1|k_i)]\boldsymbol{P}(k_i|k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_i+1|k_i)]^T+\boldsymbol{Q}(k_i) \end{cases}$$
$$\begin{cases} \hat{\boldsymbol{x}}(k_i+2|k_i)=\boldsymbol{f}[\hat{\boldsymbol{x}}(k_i+1|k_i)] \\ \boldsymbol{P}(k_i+2|k_i)=\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_i+2|k_i)]\boldsymbol{P}(k_i+1|k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_i+2|k_i)]^T+\boldsymbol{Q}(k_i+1) \end{cases} \tag{2-63}$$
$$\vdots$$
$$\begin{cases} \hat{\boldsymbol{x}}(k_{i+1}-1|k_i)=\boldsymbol{f}[\hat{\boldsymbol{x}}(k_{i+1}-2|k_i)] \\ \boldsymbol{P}(k_{i+1}-1|k_i)=\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1|k_i)]\boldsymbol{P}(k_{i+1}-2|k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1|k_i)]^T+\boldsymbol{Q}(k_{i+1}-2) \end{cases}$$

At time point k_{i+1}, a new measurement become available; therefore, both predictor and corrector equations can be applied again, respectively:

$$\begin{cases} \hat{\boldsymbol{x}}(k_{i+1}|k_i)=\boldsymbol{f}[\hat{\boldsymbol{x}}(k_{i+1}-1|k_i)] \\ \boldsymbol{P}(k_{i+1}|k_i)=\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}|k_i)]\boldsymbol{P}(k_{i+1}-1|k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}|k_i)]^T+\boldsymbol{Q}(k_{i+1}-1) \end{cases} \tag{2-64}$$

$$\begin{cases} \hat{\boldsymbol{x}}(k_{i+1}|k_{i+1})=\hat{\boldsymbol{x}}(k_{i+1}|k_i)+\boldsymbol{K}(k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1}|k_i) \\ \boldsymbol{P}(k_{i+1}|k_{i+1})=[\boldsymbol{I}-\boldsymbol{K}(k_{i+1})\boldsymbol{H}(k_{i+1})]\boldsymbol{P}(k_{i+1}|k_i) \end{cases} \tag{2-65}$$

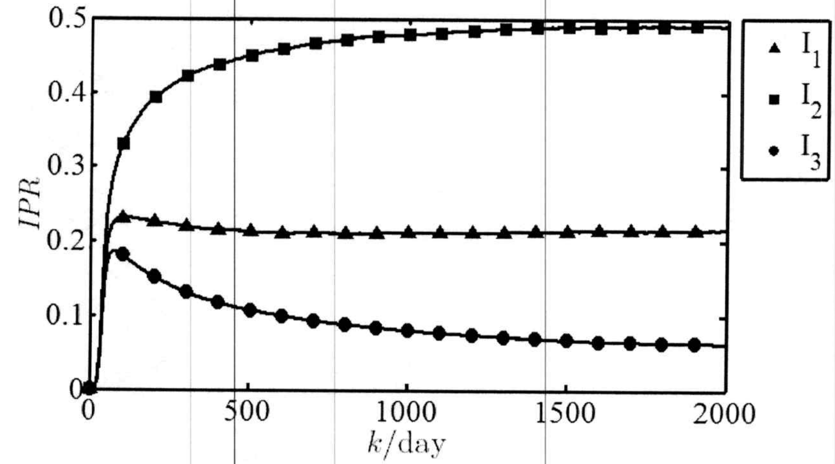


Figure 2.28 Average IPR values of the three injectors based on 100 Monte-Carlo simulations of EKF estimation using full set of production measurements.

where

$$\boldsymbol{K}(k_{i+1}) = \boldsymbol{P}(k_{i+1} \mid k_i)\boldsymbol{H}(k_{i+1})^T[\boldsymbol{H}(k_{i+1})\boldsymbol{P}(k_{i+1} \mid k_i)\boldsymbol{H}(k_{i+1})^T + \boldsymbol{R}(k_{i+1})]^{-1} \tag{2-66}$$

The derivations of the corrector equations, (2-65) and (2-66), are given below:

The estimate of $\boldsymbol{x}(k_{i+1})$ based on all available measurements up to k_{i+1}, $\hat{\boldsymbol{x}}(k_{i+1}|k_{i+1})$, can be expressed as (see lesson 12 of [121]):

$$\begin{aligned} \hat{\boldsymbol{x}}(k_{i+1} \mid k_{i+1}) &= E\{\boldsymbol{x}(k_{i+1}) \mid \boldsymbol{Z}(k_{i+1})\} \\ &= E\{\boldsymbol{x}(k_{i+1}) \mid \boldsymbol{Z}(k_i)\} + E\{\boldsymbol{x}(k_{i+1}) \mid \tilde{z}(k_{i+1} \mid k_i)\} - \boldsymbol{m}_x(k_{i+1}) \end{aligned} \tag{2-67}$$

where

$$E\{\boldsymbol{x}(k_{i+1}) \mid \tilde{z}(k_{i+1} \mid k_i)\} = \boldsymbol{P}_{x\tilde{z}}(k_{i+1}, k_{i+1} \mid k_i)\boldsymbol{P}_{\tilde{z}\tilde{z}}^{-1}(k_{i+1} \mid k_i)\tilde{z}(k_{i+1} \mid k_i) + \boldsymbol{m}_x(k_{i+1}) \tag{2-68}$$

Substitute (2-68) into (2-67), it follows that:

$$\hat{\boldsymbol{x}}(k_{i+1} \mid k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{P}_{x\tilde{z}}(k_{i+1}, k_{i+1} \mid k_i)\boldsymbol{P}_{\tilde{z}\tilde{z}}^{-1}(k_{i+1} \mid k_i)\tilde{z}(k_{i+1} \mid k_i) \tag{2-69}$$

where

$$\begin{aligned} \boldsymbol{P}_{x\tilde{z}}(k_{i+1}, k_{i+1} \mid k_i) &= E\{[\hat{\boldsymbol{x}}(k_{i+1} \mid k_i) + \tilde{\boldsymbol{x}}(k_{i+1} \mid k_i)][\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{v}(k_{i+1})]^T\} \\ &= \boldsymbol{P}(k_{i+1} \mid k_i)\boldsymbol{H}(k_{i+1})^T \end{aligned}, \tag{2-70}$$

and

$$\begin{aligned} \boldsymbol{P}_{\tilde{z}\tilde{z}}(k_{i+1} \mid k_i) &= E\{\tilde{z}(k_{i+1} \mid k_i)\tilde{z}(k_{i+1} \mid k_i)^T\} \\ &= E\{[\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{v}(k_{i+1})][\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{v}(k_{i+1})]^T\} \\ &= \boldsymbol{H}(k_{i+1})\boldsymbol{P}(k_{i+1} \mid k_i)\boldsymbol{H}(k_{i+1})^T + \boldsymbol{R}(k_{i+1}) \end{aligned} \tag{2-71}$$

Consequently, substitute (2-70) and (2-71) into (2-69), we have:

$$\hat{\boldsymbol{x}}(k_{i+1} \mid k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{K}(k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1} \mid k_i) \tag{2-72}$$

where $\boldsymbol{K}(k_{i+1})$ is given in (2-66). Then it follows from (2-72) that:

$$\begin{aligned}\tilde{\boldsymbol{x}}(k_{i+1} \mid k_{i+1}) &= \tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) - \boldsymbol{K}(k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1} \mid k_i) \\ &= \tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) - \boldsymbol{K}(k_{i+1})\left[\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{v}(k_{i+1})\right] \\ &= \left[\boldsymbol{I} - \boldsymbol{K}(k_{i+1})\boldsymbol{H}(k_{i+1})\right]\tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) - \boldsymbol{K}(k_{i+1})\boldsymbol{v}(k_{i+1})\end{aligned} \tag{2-73}$$

Then, the error covariance matrix for $\hat{\boldsymbol{x}}(k_{i+1} \mid k_{i+1})$ can be directly written as:

$$\begin{aligned}\boldsymbol{P}(k_{i+1} \mid k_{i+1}) = \left[\boldsymbol{I} - \boldsymbol{K}(k_{i+1})\boldsymbol{H}(k_{i+1})\right]\boldsymbol{P}(k_{i+1} \mid k_i)\left[\boldsymbol{I} - \boldsymbol{K}(k_{i+1})\boldsymbol{H}(k_{i+1})\right]^T \\ + \boldsymbol{K}(k_{i+1})\boldsymbol{R}(k_{i+1})\boldsymbol{K}(k_{i+1})^T\end{aligned} \tag{2-74}$$

which can be simplified as (see lesson 17 of [121]):

$$\begin{aligned}&\boldsymbol{P}(k_{i+1} \mid k_{i+1}) = \left(\boldsymbol{I} - \boldsymbol{KH}\right)\boldsymbol{P}\left(\boldsymbol{I} - \boldsymbol{KH}\right)^T + \boldsymbol{KRK}^T \\ &= \left(\boldsymbol{I} - \boldsymbol{KH}\right)\boldsymbol{P} - \boldsymbol{PH}^T\boldsymbol{K}^T + \boldsymbol{KHPH}^T\boldsymbol{K}^T + \boldsymbol{KRK}^T \\ &= \left(\boldsymbol{I} - \boldsymbol{KH}\right)\boldsymbol{P} - \boldsymbol{PH}^T\boldsymbol{K}^T + \boldsymbol{K}\left(\boldsymbol{HPH}^T + \boldsymbol{R}\right)\boldsymbol{K}^T \\ &= \left(\boldsymbol{I} - \boldsymbol{KH}\right)\boldsymbol{P} - \boldsymbol{PH}^T\boldsymbol{K}^T + \boldsymbol{PH}^T\boldsymbol{K}^T \\ &= \left[\boldsymbol{I} - \boldsymbol{K}(k_{i+1})\boldsymbol{H}(k_{i+1})\right]\boldsymbol{P}(k_{i+1} \mid k_i)\end{aligned} \tag{2-75}$$

Equations (2-72) and the last line of (2-75) are the EKF corrector equations for time point k_{i+1} [see (2-65)].■

Applying the above intermittent EKF strategy to our producer-centric model in Fig. 2.26, the average IPR curves of the three injectors based on 100 Monte-Carlo simulations are shown in Fig. 2.30 for four different scenarios, namely, the production rate is available once every (a) three days, (b) eight days, (c) 15 days, and (d) 30 days. Observe, in Fig. 2.30, that, as the sampling frequency become lower and lower, it takes longer and longer for the estimated IPR curves to converge to the correct design values, respectively. For example, in Fig. 2.30(b) where the sampling frequency, one measurement every eight days, is our most usual sampling frequency in practice, the convergence to the correct designed values are delayed for about 1000 days (almost three years) compared to that in Fig. 2.28.

Therefore, it has become clear that using intermittent measurements without any interpolation is not feasible for our EKF approach.

2.4.2 Zero-Order Hold Interpolation

In practice, one of the simplest and most often used data-interpolation (data-conditioning) methods is the Zero-Order Hold interpolation, which carries on the available measurement all the way until a new measurement become available. For example, based on 100 Monte-Carlo simulations, the average Zero-Order-Held (dotted line) production rate using the sampled production rate (discrete dots) are depicted in Fig. 2.27.

Using the Zero-Order-Held production rates as measurements for the producer-centric model in Fig. 2.26, the average IPR curves of the three injectors based on 100 Monte-Carlo simulations of our standard EKF strategy are shown in Fig. 2.31 for four different scenarios, namely, the production rate is available once every (a) three days, (b) eight days, (c) 15 days, and (d) 30 days. Observe, in Fig. 2.31, that, as the sampling frequency become lower and lower, the three estimated IPR curves all gradually converge to values closer and closer to the average of the three designed values, 0.2575.

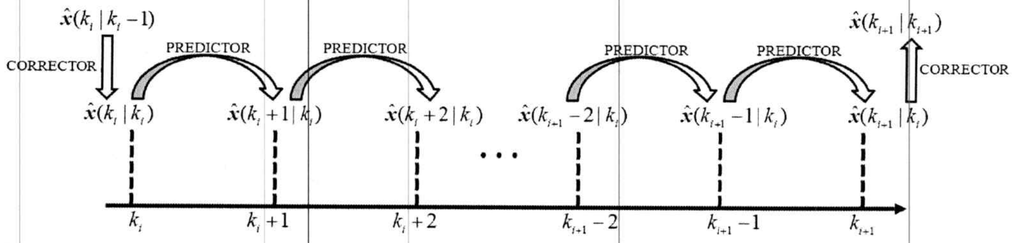

$\hat{x}(k_i \mid k_i - 1)$
CORRECTOR
PREDICTOR
PREDICTOR
PREDICTOR
PREDICTOR
$\hat{x}(k_{i+1} \mid k_{i+1})$
CORRECTOR
$\hat{x}(k_i \mid k_i)$
$\hat{x}(k_i + 1 \mid k_i)$
$\hat{x}(k_i + 2 \mid k_i)$
$\hat{x}(k_{i+1} - 2 \mid k_i)$
$\hat{x}(k_{i+1} - 1 \mid k_i)$
$\hat{x}(k_{i+1} \mid k_i)$
$\cdots$
k_i
$k_i + 1$
$k_i + 2$
$k_{i+1} - 2$
$k_{i+1} - 1$
k_{i+1}

Figure 2.29 EKF processing using intermittent measurements.

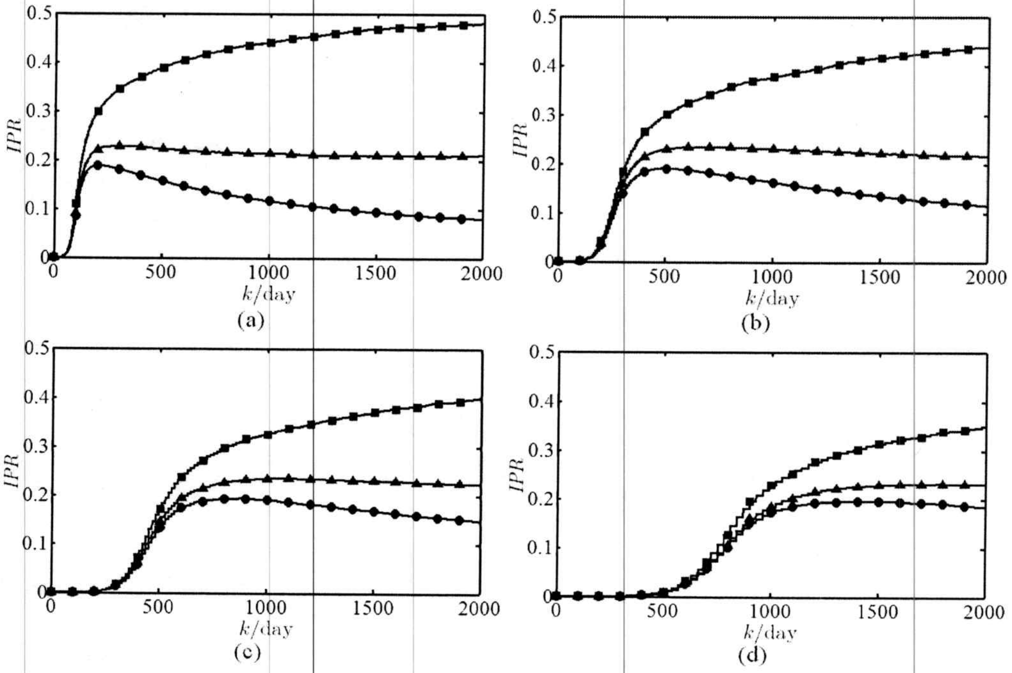


Figure 2.30 Average IPR values of the three injectors based on 100 Monte-Carlo simulations of EKF estimation using intermittent production measurements sampled once every (a) three days; (b) eight days; (c) 15 days; (d) 30 days.

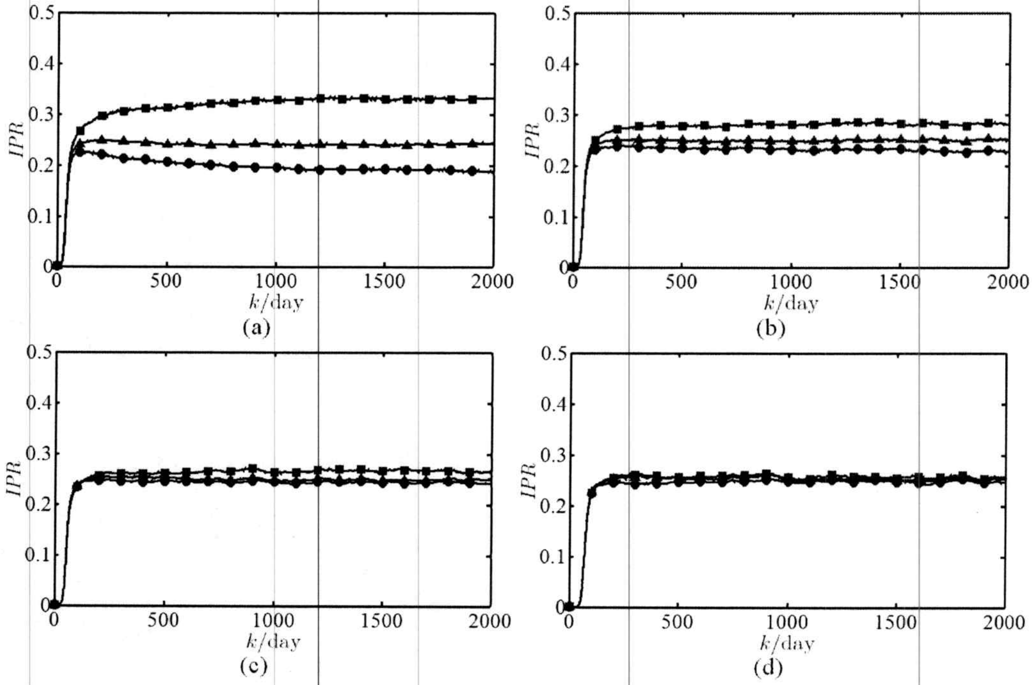


Figure 2.31 Average IPR values of the three injectors based on 100 Monte-Carlo simulations of EKF estimation using Zero-Order-Held production rates when the actual production rate is sampled once every (a) three days; (b) eight days; (c) 15 days; (d) 30 days.

Therefore, it also become clear that using the Zero-Order-Held measurements can also significantly biases our EKF estimation. To avoid all these negative impacts, a more reliable interpolation method capable of accurately reconstructing missing production rates is needed.

2.4.3 Extended Kalman Smoother using Intermittent Measurement

Before introducing our proposed data-interpolation method in the next section, we first review some necessary background of the Extended Kalman Smoother (EKS) in this section. A smoothed estimate of a state vector $\hat{\boldsymbol{x}}(\mathrm{k})$ ($k = 1,2,\ldots,D$) not only uses measurements that occur earlier than k, plus the one at k (if available), but also uses measurements after k, the so called "future measurements". One can also simply view EKS as EKF carried out in a backward fashion (see Fig. 2.32).

In Fig. 2.32, we still focus on the intermittent measurement scenario, i.e., let k_i and k_{i+1} $(1 \le i \le M-1)$ denote the time points where the i^{th} and $i+1^{\text{th}}$ measurements are taken, respectively; and, k_i+1, k_i+2,..., and $k_{i+1}-1$ are still the missing-measurement time points. Assume we are at time point k_{i+1}; and, $\hat{\boldsymbol{x}}(k_{i+1}|k_{i+1})$ and $\boldsymbol{P}(k_{i+1}|k_{i+1})$ have been obtain state estimations for time points before k_{i+1} based on available measurements up to k_{i+1}, namely, $\hat{\boldsymbol{x}}(k|k_{i+1})$ and $\boldsymbol{P}(k|k_{i+1})$ for $k = k_{i+1}-1, k_{i+1}-2, \dots, k_i$, as follows:

$$\begin{cases} \hat{\boldsymbol{x}}(k \mid k_{i+1}) = \hat{\boldsymbol{x}}(k \mid k_i) + \boldsymbol{M}(k \mid k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1} \mid k_i) \\ \qquad\qquad\;\; = \hat{\boldsymbol{x}}(k \mid k_i) + \boldsymbol{A}(k)[\hat{\boldsymbol{x}}(k_{i+1} \mid k_{i+1}) - \hat{\boldsymbol{x}}(k_{i+1} \mid k_i)] \\ \boldsymbol{P}(k \mid k_{i+1}) = \left\{ \boldsymbol{I} - \boldsymbol{M}(k \mid k_{i+1})\boldsymbol{H}(k_{i+1}) \prod_{j=k_{i+1}-1}^{k} \boldsymbol{F}_X[\hat{\boldsymbol{x}}(j \mid k_i)] \right\} \boldsymbol{P}(k \mid k_i) \end{cases} \tag{2-74}$$

where

$$\begin{aligned} \boldsymbol{M}(k \mid k_{i+1}) = \boldsymbol{P}(k \mid k_i) & \left\{ \prod_{j=k}^{k_{i+1}-1} \boldsymbol{F}_X[\hat{\boldsymbol{x}}(j \mid k_i)]^T \right\} \boldsymbol{H}(k_{i+1})^T \times \\ & [\boldsymbol{H}(k_{i+1})\boldsymbol{P}(k_{i+1} \mid k_i)\boldsymbol{H}(k_{i+1})^T + \boldsymbol{R}(k_{i+1})]^{-1} \end{aligned} \tag{2-75}$$

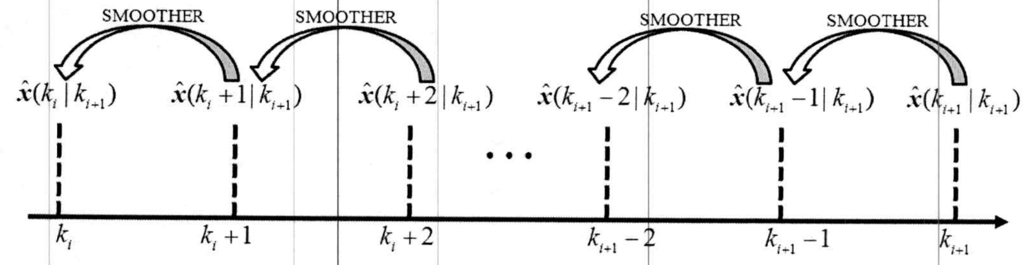

SMOOTHER
SMOOTHER
SMOOTHER
SMOOTHER
$\hat{x}(k_i \mid k_{i+1})$
$\hat{x}(k_i+1 \mid k_{i+1})$
$\hat{x}(k_i+2 \mid k_{i+1})$
$\hat{x}(k_{i+1}-2 \mid k_{i+1})$
$\hat{x}(k_{i+1}-1 \mid k_{i+1})$
$\hat{x}(k_{i+1} \mid k_{i+1})$
$\cdots$
k_i
k_i+1
k_i+2
$k_{i+1}-2$
$k_{i+1}-1$
k_{i+1}

Figure 2.32 EKS processing using intermittent measurements.

and

$$\boldsymbol{A}(k) = \boldsymbol{P}(k \mid k_i)\left\{\prod_{j=k}^{k_{i+1}-1} \boldsymbol{F}_X[\hat{\boldsymbol{x}}(j \mid k_i)]^T\right\}\boldsymbol{P}(k_{i+1} \mid k_i)^{-1} \tag{2-76}$$

The derivations of (2-74)-(2-76) are provided below:

To begin, we first derive the *single-stage* EKS equations: The estimate of $\boldsymbol{x}(k_{i+1}-1)$ based on all measurements up to k_{i+1}, $\hat{\boldsymbol{x}}(k_{i+1}-1|k_{i+1})$, can be expressed as (see lesson 12 of [121]):

$$\begin{aligned}\hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_{i+1}) &= E\{\boldsymbol{x}(k_{i+1}-1) \mid \boldsymbol{Z}(k_{i+1})\} \\ &= E\{\boldsymbol{x}(k_{i+1}-1) \mid \boldsymbol{Z}(k_i)\} + E\{\boldsymbol{x}(k_{i+1}-1) \mid \tilde{z}(k_{i+1} \mid k_i)\} - \boldsymbol{m}_x(k_{i+1}-1)\end{aligned} \tag{2-77}$$

where

$$E\{\boldsymbol{x}(k_{i+1}-1) \mid \tilde{z}(k_{i+1} \mid k_i)\} = \boldsymbol{P}_{x\tilde{z}}(k_{i+1}-1, k_{i+1} \mid k_i)\boldsymbol{P}_{\tilde{z}\tilde{z}}^{-1}(k_{i+1} \mid k_i)\tilde{z}(k_{i+1} \mid k_i) + \boldsymbol{m}_x(k_{i+1}-1) \tag{2-78}$$

Substitute (2-78) into (2-77), it follows that:

$$\hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_i) + \boldsymbol{P}_{x\tilde{z}}(k_{i+1}-1, k_{i+1} \mid k_i)\boldsymbol{P}_{\tilde{z}\tilde{z}}^{-1}(k_{i+1} \mid k_i)\tilde{z}(k_{i+1} \mid k_i) \tag{2-79}$$

where

$$\boldsymbol{P}_{x\tilde{z}}(k_{i+1}-1, k_{i+1} \mid k_i) = E\{[\hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_i) + \tilde{\boldsymbol{x}}(k_{i+1}-1 \mid k_i)][\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1} \mid k_i) + \boldsymbol{v}(k_{i+1})]^T\} \tag{2-80}$$

And, using Taylor expansion, we have:

$$\begin{aligned}&\boldsymbol{x}(k_{i+1}) \approx \\ &\boldsymbol{f}[\hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_i)] + \boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_i)][\boldsymbol{x}(k_{i+1}-1) - \hat{\boldsymbol{x}}(k_{i+1}-1 \mid k_i)] + \boldsymbol{w}(k_{i+1}-1)\end{aligned}; \tag{2-81}$$

then, it follows that

$$\hat{\boldsymbol{x}}(k_{i+1} | k_i) \approx \boldsymbol{f}[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)] \tag{2-82}$$

Subtracting (2-82) from (2-81), we have:

$$\tilde{\boldsymbol{x}}(k_{i+1} | k_i) = \boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-1 | k_i) + \boldsymbol{w}(k_{i+1}-1) \tag{2-83}$$

Substitute (2-83) into (2-80), and then it follows that:

$$\boldsymbol{P}_{x\tilde{z}}(k_{i+1}-1, k_{i+1} | k_i) = \boldsymbol{P}(k_{i+1}-1 | k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]^T \boldsymbol{H}(k_{i+1})^T \tag{2-84}$$

Note that $\boldsymbol{P}_{\tilde{z}\tilde{z}}(k_{i+1}|k_i)$ has been given in (2-71), so substituting (2-71) and (2-84) into (2-79), we have:

$$\hat{\boldsymbol{x}}(k_{i+1}-1 | k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1}-1 | k_i) + \boldsymbol{M}(k_{i+1}-1 | k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1} | k_i) \tag{2-85}$$

where

$$\begin{aligned} &\boldsymbol{M}(k_{i+1}-1 | k_{i+1}) \\ &= \boldsymbol{P}(k_{i+1}-1 | k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]^T \boldsymbol{H}(k_{i+1})^T[\boldsymbol{H}(k_{i+1})\boldsymbol{P}(k_{i+1} | k_i)\boldsymbol{H}(k_{i+1})^T + \boldsymbol{R}(k_{i+1})]^{-1} \end{aligned} \tag{2-86}$$

Substituting (2-66) into (2-86), we have:

$$\boldsymbol{M}(k_{i+1}-1 | k_{i+1}) = \boldsymbol{A}(k_{i+1}-1)\boldsymbol{K}(k_{i+1}) \tag{2-87}$$

where

$$\boldsymbol{A}(k_{i+1}-1) = \boldsymbol{P}(k_{i+1}-1 | k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]^T \boldsymbol{P}(k_{i+1} | k_i)^{-1} \tag{2-88}$$

Substituting (2-87) into (2-85), we have:

$$\begin{aligned}\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_{i+1}) &= \hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)+\boldsymbol{A}(k_{i+1}-1)\boldsymbol{K}(k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1}\,|\,k_i)\\ &= \hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)+\boldsymbol{A}(k_{i+1}-1)[\hat{\boldsymbol{x}}(k_{i+1}\,|\,k_{i+1})-\hat{\boldsymbol{x}}(k_{i+1}\,|\,k_i)]\end{aligned} \tag{2-89}$$

where the last line follows from (2-65). To derive the error-covariance matrix $\boldsymbol{P}(k_{i+1}-1|k_{i+1})$, it follows from (2-85) that:

$$\begin{aligned}&\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_{i+1})\\ &=\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1}\,|\,k_i)\\ &=\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\left[\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1}\,|\,k_i)+\boldsymbol{v}(k_{i+1})\right]\\ &=\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\times\\ &\qquad\left\{\boldsymbol{H}(k_{i+1})\left[\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)+\boldsymbol{w}(k_{i+1}-1)\right]+\boldsymbol{v}(k_{i+1})\right\}\\ &=\left[\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\right]\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)\\ &\qquad-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{w}(k_{i+1}-1)-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\boldsymbol{v}(k_{i+1})\end{aligned} \tag{2-90}$$

Consequently, the error covariance matrix $\boldsymbol{P}(k_{i+1}-1|k_{i+1})$ can be directly written as:

$$\begin{aligned}&\boldsymbol{P}(k_{i+1}-1\,|\,k_{i+1})\\ &=\left[\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\right]\boldsymbol{P}(k_{i+1}-1\,|\,k_i)\\ &\quad\left[\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\right]^T+\\ &\qquad\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\left[\boldsymbol{H}(k_{i+1})\boldsymbol{Q}(k_{i+1}-1)\boldsymbol{H}(k_{i+1})^T+\boldsymbol{R}(k_{i+1})\right]\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})^T\end{aligned} \tag{2-91}$$

which can be simplified as:

$$\begin{aligned}\boldsymbol{P}(k_{i+1}-1\,|\,k_{i+1})&=\left(\boldsymbol{I}-\boldsymbol{MHF}_X\right)\boldsymbol{P}\left(\boldsymbol{I}-\boldsymbol{MHF}_X\right)^T+\boldsymbol{M}\left[\boldsymbol{HQH}^T+\boldsymbol{R}\right]\boldsymbol{M}^T\\ &=\left(\boldsymbol{I}-\boldsymbol{MHF}_X\right)\boldsymbol{P}-\boldsymbol{PF}_X^T\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{MHF}_X\boldsymbol{PF}_X^T\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{M}\left[\boldsymbol{HQH}^T+\boldsymbol{R}\right]\boldsymbol{M}^T\\ &=\left(\boldsymbol{I}-\boldsymbol{MHF}_X\right)\boldsymbol{P}-\boldsymbol{PF}_X^T\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{M}\left\{\boldsymbol{H}\left[\boldsymbol{F}_X\boldsymbol{PF}_X^T+\boldsymbol{Q}\right]\boldsymbol{H}^T+\boldsymbol{R}\right\}\boldsymbol{M}^T\\ &=\left(\boldsymbol{I}-\boldsymbol{MHF}_X\right)\boldsymbol{P}-\boldsymbol{PF}_X^T\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{PF}_X^T\boldsymbol{H}^T\boldsymbol{M}^T\\ &=\left\{\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-1\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\right\}\boldsymbol{P}(k_{i+1}-1\,|\,k_i)\end{aligned} \tag{2-92}$$

Note that (2-89) and the last line of (2-92) are the single-stage EKS equations for $k_{i+1}-1$:

$$\begin{cases} \hat{\boldsymbol{x}}(k_{i+1}-1|k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1}-1|k_i) + \boldsymbol{M}(k_{i+1}-1|k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1}|k_i) \\ \qquad\qquad\qquad\quad = \hat{\boldsymbol{x}}(k_{i+1}-1|k_i) + \boldsymbol{A}(k_{i+1}-1)[\hat{\boldsymbol{x}}(k_{i+1}|k_{i+1}) - \hat{\boldsymbol{x}}(k_{i+1}|k_i)] \\ \boldsymbol{P}(k_{i+1}-1|k_{i+1}) = \{\boldsymbol{I} - \boldsymbol{M}(k_{i+1}-1|k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1|k_i)]\}\boldsymbol{P}(k_{i+1}-1|k_i) \end{cases} \tag{2-93}$$

Next, as our second step, we derive the *double-stage* EKS equations. The estimate of $\boldsymbol{x}(k_{i+1}-2)$ based on all measurements up to k_{i+1}, $\hat{\boldsymbol{x}}(k_{i+1}-2|k_{i+1})$, can be expressed as (see lesson 12 of [121]):

$$\begin{aligned} \hat{\boldsymbol{x}}(k_{i+1}-2|k_{i+1}) &= E\{\boldsymbol{x}(k_{i+1}-2)|\boldsymbol{Z}(k_{i+1})\} \\ &= E\{\boldsymbol{x}(k_{i+1}-2)|\boldsymbol{Z}(k_i)\} + E\{\boldsymbol{x}(k_{i+1}-2)|\tilde{z}(k_{i+1}|k_i)\} - \boldsymbol{m}_x(k_{i+1}-2) \end{aligned} \tag{2-94}$$

where

$$E\{\boldsymbol{x}(k_{i+1}-2)|\tilde{z}(k_{i+1}|k_i)\} = \boldsymbol{P}_{x\tilde{z}}(k_{i+1}-2,k_{i+1}|k_i)\boldsymbol{P}_{\tilde{z}\tilde{z}}^{-1}(k_{i+1}|k_i)\tilde{z}(k_{i+1}|k_i) + \boldsymbol{m}_x(k_{i+1}-2) \tag{2-95}$$

Substitute (2-95) into (2-94), it follows that:

$$\hat{\boldsymbol{x}}(k_{i+1}-2|k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1}-2|k_i) + \boldsymbol{P}_{x\tilde{z}}(k_{i+1}-2,k_{i+1}|k_i)\boldsymbol{P}_{\tilde{z}\tilde{z}}^{-1}(k_{i+1}|k_i)\tilde{\boldsymbol{z}}(k_{i+1}|k_i) \tag{2-96}$$

where

$$\boldsymbol{P}_{x\tilde{z}}(k_{i+1}-2,k_{i+1}|k_i) = E\{[\hat{\boldsymbol{x}}(k_{i+1}-2|k_i) + \tilde{\boldsymbol{x}}(k_{i+1}-2|k_i)][\boldsymbol{H}(k_{i+1})\tilde{\boldsymbol{x}}(k_{i+1}|k_i) + \boldsymbol{v}(k_{i+1})]^T\} \tag{2-97}$$

and it follows from (2-83) that:

$$\tilde{\boldsymbol{x}}(k_{i+1}-1|k_i) = \boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2|k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-2|k_i) + \boldsymbol{w}(k_{i+1}-2) \tag{2-98}$$

Substituting (2-98) into (2-83), we have:

$$\begin{aligned}&\tilde{\boldsymbol{x}}(k_{i+1} | k_i) = \\ &\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]\{\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2 | k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-2 | k_i)+\boldsymbol{w}(k_{i+1}-2)\}+\boldsymbol{w}(k_{i+1}-1)\end{aligned} \tag{2-99}$$

Substituting (2-99) into (2-97), we have:

$$\boldsymbol{P}_{x\tilde{z}}(k_{i+1}-2, k_{i+1} | k_i) = \boldsymbol{P}(k_{i+1}-2 | k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2 | k_i)]^T \boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]^T \boldsymbol{H}(k_{i+1})^T \tag{2-100}$$

Note that $\boldsymbol{P}_{\tilde{z}\tilde{z}}(k_{i+1}|k_i)$ has been given in (2-71), so substituting (2-71) and (2-100) into (2-96), we have:

$$\hat{\boldsymbol{x}}(k_{i+1}-2 | k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1}-2 | k_i) + \boldsymbol{M}(k_{i+1}-2 | k_{i+1})\tilde{z}(k_{i+1} | k_i) \tag{2-101}$$

where

$$\begin{aligned}\boldsymbol{M}(k_{i+1}-2 | k_{i+1}) = \boldsymbol{P}(k_{i+1}-2 | k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2 | k_i)]^T \boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]^T \boldsymbol{H}(k_{i+1})^T \\ [\boldsymbol{H}(k_{i+1})\boldsymbol{P}(k_{i+1} | k_i)\boldsymbol{H}(k_{i+1})^T + \boldsymbol{R}(k_{i+1})]^{-1}\end{aligned} \tag{2-102}$$

Substituting (2-66) into (2-102), we have:

$$\boldsymbol{M}(k_{i+1}-2 | k_{i+1}) = \boldsymbol{A}(k_{i+1}-2)\boldsymbol{K}(k_{i+1}) \tag{2-103}$$

where

$$\boldsymbol{A}(k_{i+1}-2) = \boldsymbol{P}(k_{i+1}-2 | k_i)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2 | k_i)]^T \boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1 | k_i)]^T \boldsymbol{P}(k_{i+1} | k_i)^{-1} \tag{2-104}$$

Substituting (2-103) into (2-101), we have:

$$\begin{aligned}\hat{\boldsymbol{x}}(k_{i+1}-2 | k_{i+1}) &= \hat{\boldsymbol{x}}(k_{i+1}-2 | k_i) + \boldsymbol{A}(k_{i+1}-2)\boldsymbol{K}(k_{i+1})\tilde{z}(k_{i+1} | k_i) \\ &= \hat{\boldsymbol{x}}(k_{i+1}-2 | k_i) + \boldsymbol{A}(k_{i+1}-2)[\hat{\boldsymbol{x}}(k_{i+1} | k_{i+1}) - \hat{\boldsymbol{x}}(k_{i+1} | k_i)]\end{aligned} \tag{2-105}$$

where the last line follows from (2-65). To derive the error-covariance matrix $\boldsymbol{P}(k_{i+1}-2|k_{i+1})$, it follows from (2-101) that:

$$\begin{aligned}
\tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_{i+1}) &= \tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1}\,|\,k_i)\\
&= \tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\times\\
&\qquad \left\{\boldsymbol{H}(k_{i+1})\left[\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)+\boldsymbol{w}(k_{i+1}-1)\right]+\boldsymbol{v}(k_{i+1})\right\}\\
&= \tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)\\
&\qquad -\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{w}(k_{i+1}-1)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{v}(k_{i+1})\\
&= \tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\\
&\qquad \left[\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)]\tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)+\boldsymbol{w}(k_{i+1}-2)\right]\\
&\qquad -\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{w}(k_{i+1}-1)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{v}(k_{i+1})\\
&= \left[\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)]\right]\tilde{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)\\
&\qquad -\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{w}(k_{i+1}-2)\\
&\qquad -\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{w}(k_{i+1}-1)-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{v}(k_{i+1})
\end{aligned}\tag{2-106}$$

Consequently, the error covariance matrix $\boldsymbol{P}(k_{i+1}-2|k_{i+1})$ can be directly written as:

$$\begin{aligned}
&\boldsymbol{P}(k_{i+1}-2\,|\,k_{i+1})\\
&=\left[\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)]\right]\\
&\quad \boldsymbol{P}(k_{i+1}-2\,|\,k_i)\left[\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)]\right]^T\\
&+\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\left[\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{Q}(k_{i+1}-2)\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]^T\right.\\
&\left.+\boldsymbol{Q}(k_{i+1}-1)\right]\boldsymbol{H}(k_{i+1})^T\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})^T+\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{R}(k_{i+1})\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})^T
\end{aligned}\tag{2-107}$$

which can be simplified as:

$$
\begin{aligned}
&\boldsymbol{P}(k_{i+1}-2\,|\,k_{i+1}) = \left[\boldsymbol{I}-\boldsymbol{MHF}_X(k_{i+1}-1)\boldsymbol{F}_X(k_{i+1}-2)\right] \\
&\qquad \times \boldsymbol{P}\left[\boldsymbol{I}-\boldsymbol{MHF}_X(k_{i+1}-1)\boldsymbol{F}_X(k_{i+1}-2)\right]^T \\
&\qquad\quad +\boldsymbol{MH}\left[\boldsymbol{F}_X(k_{i+1}-1)\boldsymbol{Q}(k_{i+1}-2)\boldsymbol{F}_X(k_{i+1}-1)^T+\boldsymbol{Q}(k_{i+1}-1)\right]\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{MRM}^T \\
&= \left[\boldsymbol{I}-\boldsymbol{MHF}_X(k_{i+1}-1)\boldsymbol{F}_X(k_{i+1}-2)\right]\boldsymbol{P}-\boldsymbol{PF}_X(k_{i+1}-2)^T\boldsymbol{F}_X(k_{i+1}-1)^T\boldsymbol{H}^T\boldsymbol{M}^T \\
&\qquad +\boldsymbol{MHF}_X(k_{i+1}-1)\boldsymbol{F}_X(k_{i+1}-2)\boldsymbol{PF}_X(k_{i+1}-2)^T\boldsymbol{F}_X(k_{i+1}-1)^T\boldsymbol{H}^T\boldsymbol{M}^T \\
&\qquad\quad +\boldsymbol{MH}\left[\boldsymbol{F}_X(k_{i+1}-1)\boldsymbol{Q}(k_{i+1}-2)\boldsymbol{F}_X(k_{i+1}-1)^T+\boldsymbol{Q}(k_{i+1}-1)\right]\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{MRM}^T \\
&= \left[\boldsymbol{I}-\boldsymbol{MHF}_X(k_{i+1}-1)\boldsymbol{F}_X(k_{i+1}-2)\right]\boldsymbol{P}-\boldsymbol{PF}_X(k_{i+1}-2)^T\boldsymbol{F}_X(k_{i+1}-1)^T\boldsymbol{H}^T\boldsymbol{M}^T \\
&+\boldsymbol{MH}\left\{\boldsymbol{F}_X(k_{i+1}-1)\left[\boldsymbol{F}_X(k_{i+1}-2)\boldsymbol{PF}_X(k_{i+1}-2)^T+\boldsymbol{Q}(k_{i+1}-2)\right]\boldsymbol{F}_X(k_{i+1}-1)^T\right. \\
&\hspace{18em} \left.+\boldsymbol{Q}(k_{i+1}-1)\right\}\boldsymbol{H}^T\boldsymbol{M}^T+\boldsymbol{MRM}^T \\
&= \left[\boldsymbol{I}-\boldsymbol{MHF}_X(k_{i+1}-1)\boldsymbol{F}_X(k_{i+1}-2)\right]\boldsymbol{P}-\boldsymbol{PF}_X(k_{i+1}-2)^T\boldsymbol{F}_X(k_{i+1}-1)^T\boldsymbol{H}^T\boldsymbol{M} \\
&\hspace{14em} +\boldsymbol{PF}_X(k_{i+1}-2)^T\boldsymbol{F}_X(k_{i+1}-1)^T\boldsymbol{H}^T\boldsymbol{M} \\
&= \left\{\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)]\right\}\boldsymbol{P}(k_{i+1}-2\,|\,k_i)
\end{aligned}
\tag{2-108}
$$

Note that (2-105) and the last line of (2-108) are the double-stage EKS equations for $k_{i+1}-2$:

$$
\begin{cases}
\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_{i+1}) = \hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)+\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\tilde{\boldsymbol{z}}(k_{i+1}\,|\,k_i) \\
\qquad\qquad\qquad = \hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)+\boldsymbol{A}(k_{i+1}-2)[\hat{\boldsymbol{x}}(k_{i+1}\,|\,k_{i+1})-\hat{\boldsymbol{x}}(k_{i+1}\,|\,k_i)] \\
\boldsymbol{P}(k_{i+1}-2\,|\,k_{i+1}) \\
=\left\{\boldsymbol{I}-\boldsymbol{M}(k_{i+1}-2\,|\,k_{i+1})\boldsymbol{H}(k_{i+1})\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-1\,|\,k_i)]\boldsymbol{F}_X[\hat{\boldsymbol{x}}(k_{i+1}-2\,|\,k_i)]\right\}\boldsymbol{P}(k_{i+1}-2\,|\,k_i)
\end{cases}
\tag{2-109}
$$

Finally, based on (2-93) and (2-109), (2-74)-(2-76) can be proved by mathematical induction.■

2.4.4 Iterated Extended Kalman Filter and Smoother

Without loss of generality, assume we want to interpolate all missing production rates between the two time points, k_i and k_{i+1}, where measurements are available. Note that, in

Sections 2.2 and 2.3, all individual production rates contributed by each injector, $p_j^c(k)$ ($k = k_i, k_i + 1, \dots, k_{i+1}, j = 1,2, \dots, N_I$), are modeled as state variables in the state vector $\boldsymbol{x}(k)$; therefore, once an estimate of $\boldsymbol{x}(k)$ become available, we will also have an estimate of $p_j^c(k)$, which allows us, by using the measurement equation in (2-15), to obtain an estimate of the gross production rate $p(k)$.

If we rely solely on the EKF to provide us with the estimate of $p(k)$, we would only be utilizing the measurements up to time point k_i, $\boldsymbol{Z}(k_i)$. However, a robust interpolation method should also utilize the measurement after the missing-measurement time points, $\boldsymbol{z}(k_{i+1})$. In order to do so, we introduce the EKS into our estimation process.

Specifically, once the estimates of the states, $\hat{\boldsymbol{x}}(k|k_{i+1})$ ($k = k_i, k_i + 1, \dots, k_{i+1}$), are computed by the EKF using measurements up to k_i, we are then able to use the EKS equations, (2-74)-(2-76), to obtain a new set of estimates of the states using measurements up to k_{i+1}, $\hat{\boldsymbol{x}}(k|k_{i+1})$ ($k = k_{i+1}, k_{i+1} - 1, \dots, k_i$), in a backward manner.

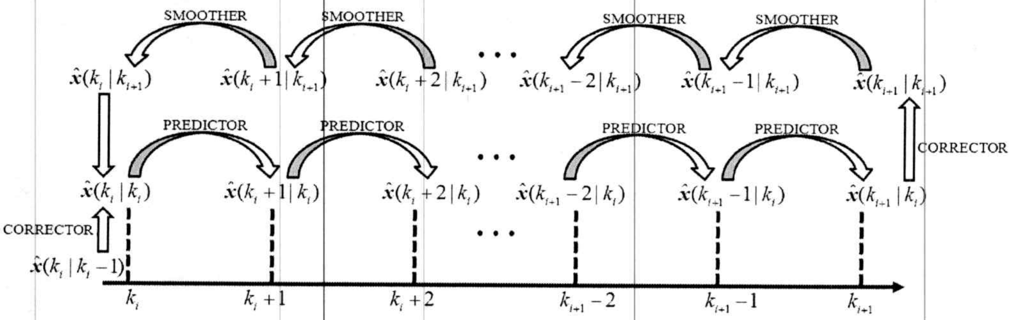

SMOOTHER
SMOOTHER
SMOOTHER
SMOOTHER
$\hat{x}(k_i \mid k_{i+1})$
$\hat{x}(k_i + 1 \mid k_{i+1})$
$\hat{x}(k_i + 2 \mid k_{i+1})$
$\hat{x}(k_{i+1} - 2 \mid k_{i+1})$
$\hat{x}(k_{i+1} - 1 \mid k_{i+1})$
$\hat{x}(k_{i+1} \mid k_{i+1})$
PREDICTOR
PREDICTOR
PREDICTOR
PREDICTOR
CORRECTOR
$\hat{x}(k_i \mid k_i)$
$\hat{x}(k_i + 1 \mid k_i)$
$\hat{x}(k_i + 2 \mid k_i)$
$\hat{x}(k_{i+1} - 2 \mid k_i)$
$\hat{x}(k_{i+1} - 1 \mid k_i)$
$\hat{x}(k_{i+1} \mid k_i)$
CORRECTOR
$\hat{x}(k_i \mid k_i - 1)$
k_i
$k_i + 1$
$k_i + 2$
$k_{i+1} - 2$
$k_{i+1} - 1$
k_{i+1}

Figure 2.33 Iterative Extended Kalman Filter and Smoother.

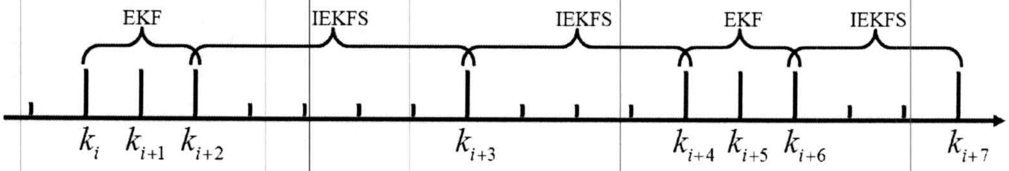


Figure 2.34 State Estimation based on EKF and IEKFS for intermittent measurements.

Once we come back to time point k_i, we can update the estimated states in $\hat{x}(k_i|k_i)$ with the estimated states in $\hat{x}(k_i|k_{i+1})$; and, then, repeat such EKF and EKS process iteratively (see Fig. 2.33) until the estimated states have converged or we believe an accurate enough estimate has been achieved.

Finally, we can feed the iteratively estimated state vector into the measurement equation, obtain an estimate of the gross production rate, and use it as the interpolated measurement. Therefore, this approach is named *Iterative Extended Kalman Filter and Smoother* (IEKFS) method.

So far, our IEKFS has been focusing on state estimation in the interval between (including) two available intermittent measurements. However, in practice, consecutive and intermittent measurements occur alternatively as shown in Fig. 2.34. Therefore, as indicated in Fig. 2.34, the overview of our global strategy is to use one-pass EKF when consecutive measurements are available, and to use IEKFS when only intermittent measurements are available. Of course, for interpolation purpose, we would only need the estimated states at those missing-measurement time points; but we need the EKF for the consecutive-measurement time points to update and prepare the estimated states for the intermittent measurement interval that follows.

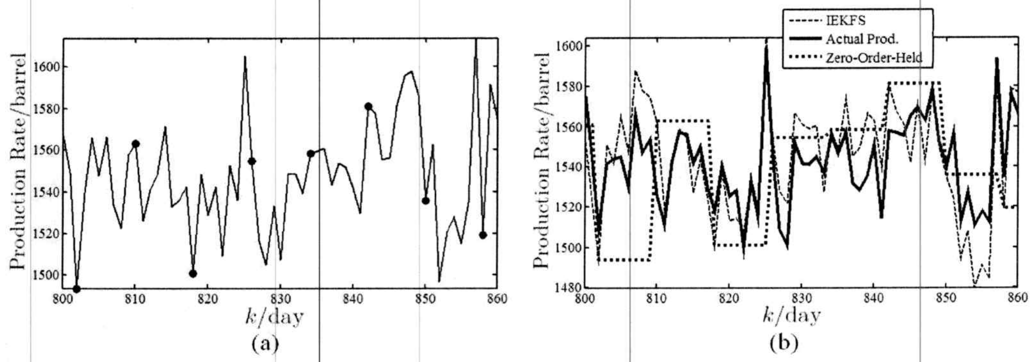


Figure 2.35 Based on 100 Monte-Carlo simulations of the producer-centric model from Fig. 2.26, a window (day 800 ~ day 860) of (a) average production rates with measurement noise (solid line) and our sampled measurements (discrete dots); (b) average actual production rates without noise (solid line), IEKFS interpolated production rates (dashed line) and Zero-Order-Held production rates (dotted line).

2.5 Case Study

2.5.1 Hypothetical Test Case Revisited

To test the performance of our IEKFS approach, we first go back to the single producer-centric model introduced in Fig. 2.26. And we mainly focus on the scenario where the noised production rates are sampled once every eight days [see Fig. 2.35(a)], because it best reflects the real world situation.

In Fig. 2.35(b), we show how the IEKFS interpolated and Zero-Order-Held production rates compare to the actual production rates without measurement noise; and observe that, unlike the Zero-Order-Held rates, the IEKFS approach manage to capture the ups and downs (trends) of the actual production rates, which is very important for our state estimation process. To numerically evaluate the two approaches, we compute their Root Mean Square Errors (RMSEs) compared to the actual production rates, and summarize them in Table 2.11. Observe, in Table 2.11, that the RMSE of the IEKFS approach is 48.8% smaller than that of the Zero-Order-Hold approach.

The second evaluation step is to use the IEKFS interpolated production rates as measurements for our EKF estimation process, the results of which are depicted in Fig. 2.36. Observe, in Fig. 2.36, that the average IPR values of the three injectors quickly converge to the correct designed values like Fig. 2.28; in other words, the IPR values estimated using IEKFS interpolated production rates when the original sampling frequency is one measurement every eight days are almost equivalent to IPR values estimated using the full set of measurements.

2.5.2 Reservoir Simulator Case Study

To further validate our IEKFS approach, we study two reservoir simulator generated cases in this section. Note that, for all these test cases, we still assume the sampling frequency is always one measurement every eight days.

The first test case includes four producers and five injectors; its permeability map is depicted in Fig. 2.37, where the entire field is homogenous and has very low permeability except a high permeability channel between injector 1 and producer 1 and a medium permeability channel between injector 3 and producer 4. Also note that, since the total number of injectors is very small, we include all injectors in every producer-centric model and do not eliminate any of them during the EKF process.

Table 2.11 Root Mean Square Errors (bpd) of the IEKFS and Zero-Order Hold interpolation methods for the single producer-centric model from Fig. 2.26.

	IEKFS	Zero-Order Hold
$RMSE = \sqrt{\sum_{k=1}^{N}[p(k) - \hat{p}(k)]^2/N}$	17.2 bpd	33.6 bpd

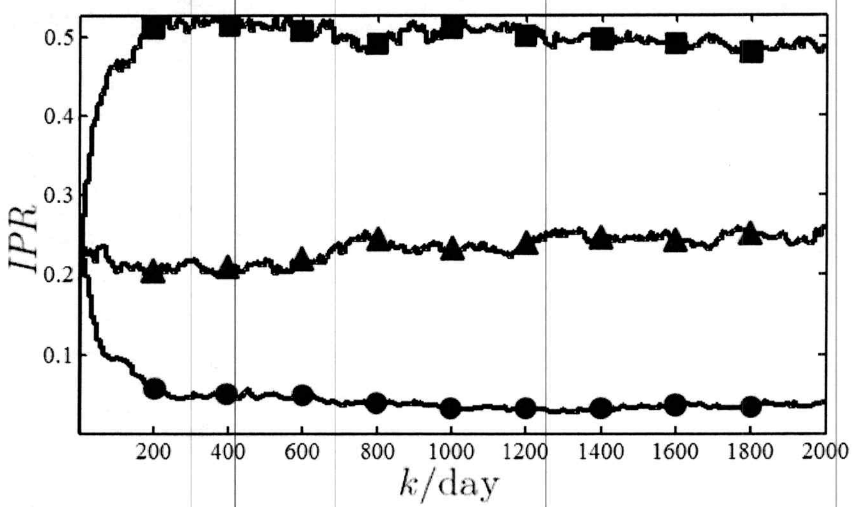


Figure 2.36 Average IPR values of the three injectors based on 100 Monte-Carlo simulations of EKF estimation using IEKFS interpolated production rates (one measurement every eight days).

The RMSEs of the interpolated production rates by the IEKFS and Zero-Order Hold methods for all four producers are summarized in Table 2.12, where we can observe that the IEKFS always outperform the Zero-Order Hold method. And, in Fig. 2.38, we present the estimated IPR values between producer 2 and all five injectors for four scenarios: (a) full set of the measurements are used; (b) only intermittent measurements are used; (c) Zero-Order-Held production rates are used; and (d) the IEKFS interpolated production rates are used.

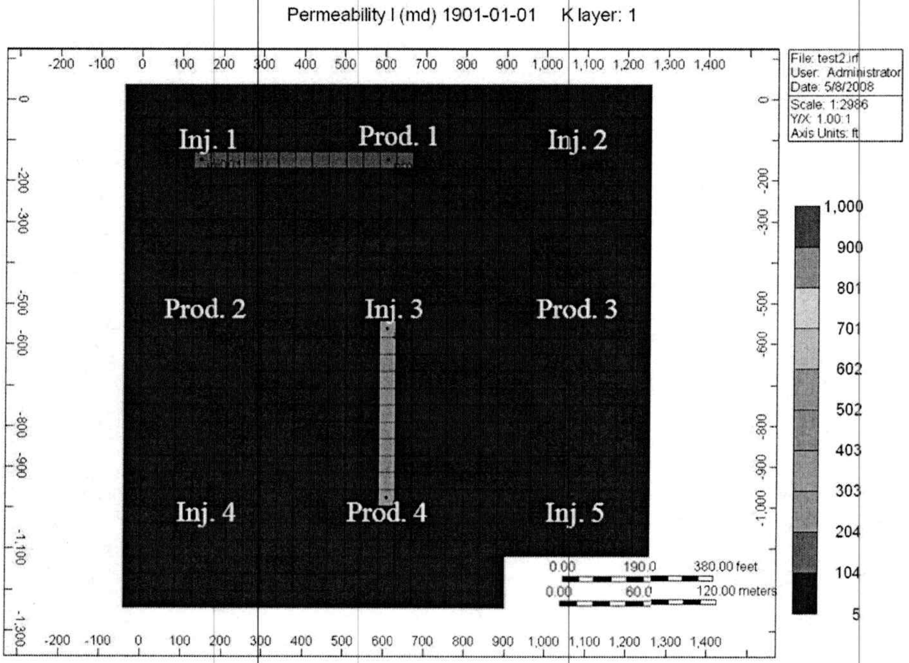


Figure 2.37 Permeability map of reservoir simulator generated case 1.

Table 2.12 Root Mean Square Errors (bpd) of the IEKFS and Zero-Order Hold interpolated production rates for the four producer-centric models in reservoir simulator case 1.

	Producer 1	Producer 2	Producer 3	Producer 4
IEKFS	85.11	4.80	12.30	56.54
Zero-Order Hold	186.66	25.20	40.17	126.37

Table 2.13 Root Mean Square Errors (bpd) of the IEKFS and Zero-Order Hold interpolated production rates for the four producer-centric models in reservoir simulator case 2.

	Producer 1	Producer 2	Producer 3	Producer 4
IEKFS	9.28	5.36	11.70	96.75
Zero-Order Hold	28.00	28.36	35.00	193.84

Observe, in Fig. 2.38(a), that all IPR curves have very small values before day 1500; this is because most water from the injectors hasn't reached producer 2 yet. Shortly after that day, the IPR curve of injector 4 increased by a relatively large amount. This is because injector 4 is one of the closest injectors to producer 2; and although injectors 1 and 3 are equivalently close to producer 2, most of their water has been contributed to producers 1 and 4, respectively, due to their high and medium permeability channels. Therefore, given full set of measurements, the EKF can provide us with IPR results that correctly reflect the physical setup of the reservoir simulator experiment.

On the other hand, we observe that the IPR curves in Figs. 2.38(b) and (c) are significantly biased; whereas the IPR curves in Fig. 2.38(d) are almost the same as their counterparts in Fig. 2.38(a). Similar results can be obtained for the other three producer-centric models in this test case. These suggest that the IEKFS interpolated production rates allow us to obtain almost equivalent results as if we had used full set of the measurements.

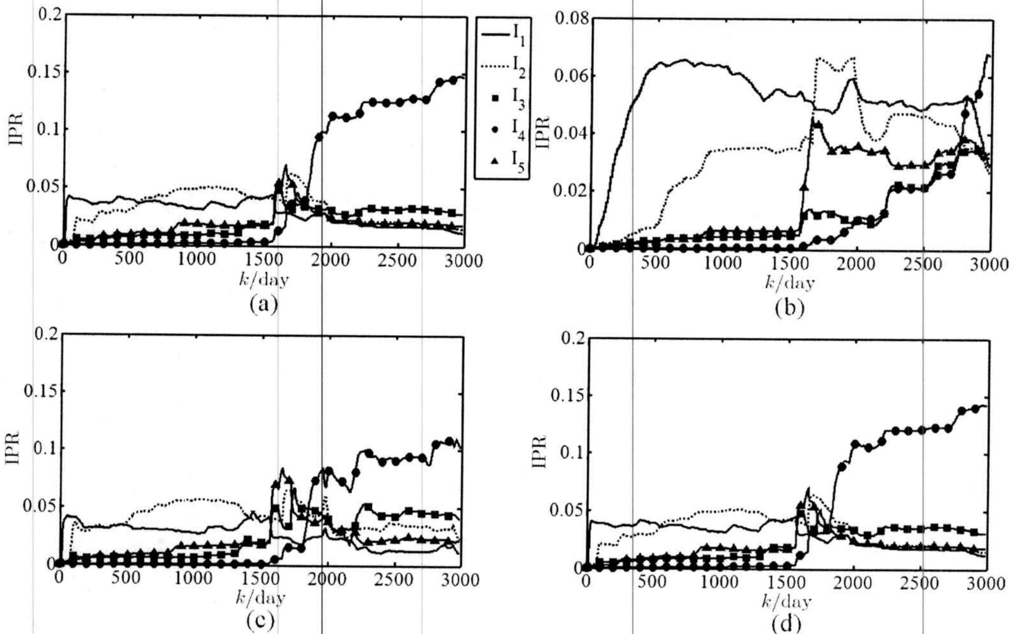


Figure 2.38 For reservoir simulator case 1, IPR values between producer 2 and all five injectors estimated by EKF that uses (a) full set of the measurements; (b) intermittent measurements; (c) Zero-Order-Held production rates; and (d) IEKFS interpolated production rates.

The second test case includes the same number of producers and injectors as the first case; its permeability map is depicted in Fig. 2.39, where the entire field is homogenous and has very low permeability except an extremely high permeability channel across the field. Again, all injectors are included in every producer-centric model during the EKF process.

The RMSEs of the interpolated production rates by the IEKFS and Zero-Order Hold methods for all four producer-centric models are summarized in Table 2.13, where the IEKFS still always outperform the Zero-Order Hold method. And, in Fig. 2.40, we present the estimated IPR values between producer 2 and all five injectors for four different scenarios as we did for the first case.

In Fig. 2.40(a), for example, we observe that injector 1 has the highest IPR values throughout the test, which is physically meaningful; because it is the closest injector located in the high permeability channel with producer 2.

And, similarly, we observe that the IPR curves in Figs. 2.40(b) and (c) are significantly biased; whereas the IPR curves in Fig. 2.40(d) are almost the same as their counterparts in Fig. 2.40(a). Similar results can be obtained for the other three producer-centric models in this test case. These, again, verify that the IEKFS interpolation approach enables our EKF to produce IPR values that correctly reflect the physical conditions of the field even when the sampling frequency of the production rates is very low.

2.6 Summary

After testing the EKF extensively on the real data, Section 2.2 first introduced several important modifications to the SVM, allowing the square root of IPR values to be estimated

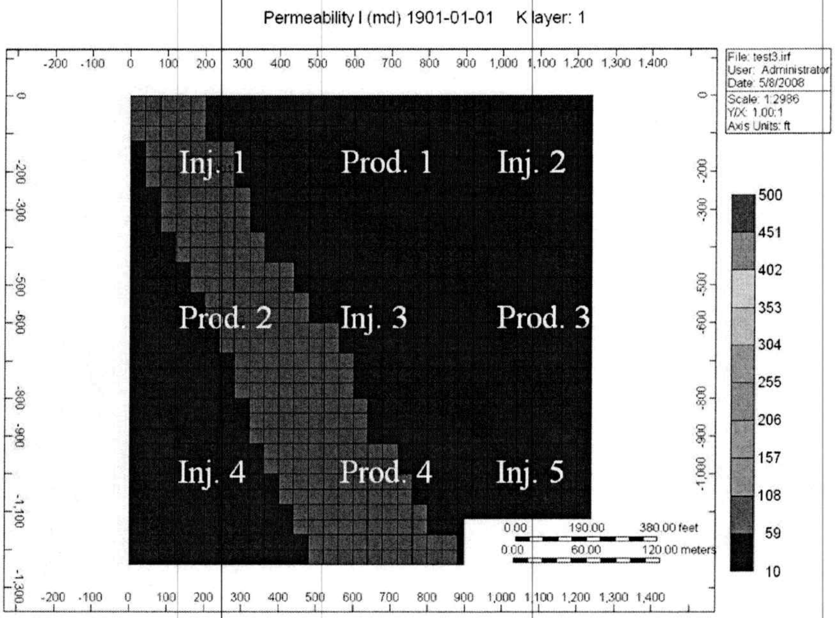
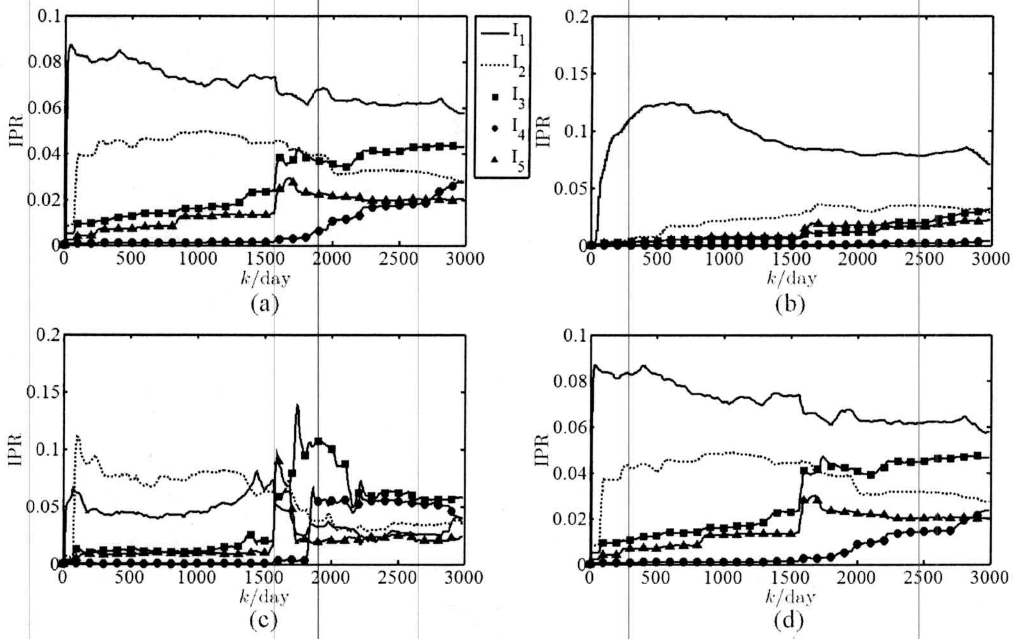


Figure 2.39 Permeability map of reservoir simulator generated case 2.



(a)

(b)

(c)

(d)

Figure 2.40 For reservoir simulator case 2, IPR values between producer 2 and all five injectors estimated by EKF that uses (a) full set of the measurements; (b) intermittent measurements; (c) Zero-Order-Held production rates; and (d) IEKFS interpolated production rates.

directly, which avoids estimating negative results for IPR values. Section 2.2 then went on to describe a way that integrates important expert knowledge on oilfield for the initialization of the EKF processing, present a method that eliminates noncontributing injectors from the model, and show how to validate the accuracy of our EKF results by using production-rate history-matching. Our validation process has also led us to develop a method for selecting an optimum ellipse size (for each producer) for choosing the initial set of injectors by minimizing history-matching errors. Finally, we showed how to convert a table of producer-centric IPR values to a table of injector centric IPR values.

Then, in Section 2.3, we used a simulated mini-oilfield to illustrate how independent EKF processing of producer-centric models can lead to the violation of physically meaningful constraints. We then extended Simon, et al's constrained Kalman Filter methods to a Constrained EKF (CEKF) that, for the first time, has coupled all producer-centric models in the oilfield. Our simulation results have shown that the CEKF guarantees satisfaction of the constraints on the state vectors, and produces more accurate IPR values for producers that are incorrectly modeled, but it biases IPR values for producers that are correctly modeled. To address this problem, we have shown how the dimensionality of the CEKF can be significantly reduced and how a better weighting matrix can be easily designed to maintain IPRs of injectors that are more likely to be contributing to a producer, and at the same time to reduce the "pseudo-IPR" values of those that are more likely not to be contributing to a producer. The CEKF reduces to the EKF when no constraint is violated. We also applied the CEKF to 13 real oilfield producers that are affected by an injector whose constraint is severely violated, and have shown that the constraint was successfully restored by the CEKFs.

Finally, in this Sections 2.4 and 2.5, based on Extended Kalman Filter and Smoother, we proposed an IEKFS approach to dynamically interpolate the missing measurements between two available measurements. Compared to using intermittent measurements directly, which will cause the estimation results to take extremely long time to converge to the correct values, and using Zero-Order-Held measurements, which will cause the estimation results to all converge to an average value, the IEKFS interpolated measurements allow our EKF to quickly and accurately estimate the IPR values even when the sampling frequency is as low as one measurement every eight days. The IEKFS interpolated measurements are very important for the performances of the works we developed in Sections 2.2 and 2.3, and can also be utilized by all other signal-processing-based techniques.

Chapter 3 Universal Image Noise Removal Filter based on Type-2 Fuzzy Logic System and QPSO

3.1 Introduction

In this and the following chapter, we focus on the designing of robust learning systems. Through the two applications studied in Sections 3 and 4, we introduce a paradigm for developing advanced Fuzzy Logic Systems (FLSs) that take different forms.

As mentioned in Section 1.2, the first such an application is regarding Removing Mixed Gaussian and Impulse Noise (MGIN), which is considered to be one of the most essential topics in the domain of image restoration, and is much more challenging than to remove pure Gaussian or impulse noise separately. Therefore, relatively fewer works have been published in this area. This chapter proposes a new integrated approach for MGIN removal that is based on a Non-Singleton Interval Type-2 (NS-IT2) FLS, and explains how to design such a NS-IT2 FLS using a Quantum-behaved Particle Swarm Optimization (QPSO) algorithm. Then the chapter goes on to introduce two supplementary components, a Block-Matching 3-Dimensional Discrete Cosine Transformation (BM3D DCT) filter and a contrast scaling filter, which augment the overall performance of the NS-IT2 FLS. Finally, the chapter shows that this proposed approach indeed provides both quantitatively and visually much better results compared to other often-used non-fuzzy techniques as well as its Type-1 (T1) and singleton IT2 (S-IT2) counterparts.

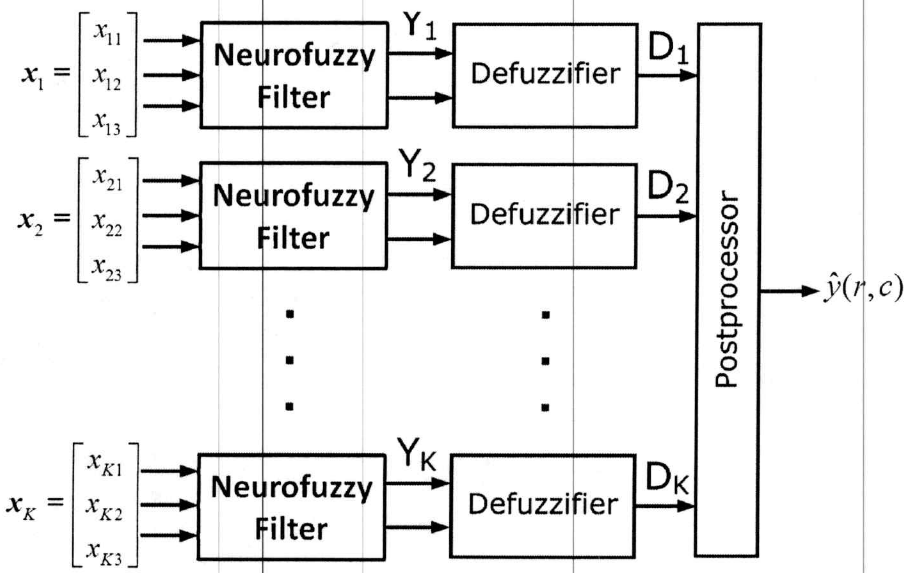

$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix}$
$x_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix}$
$x_K = \begin{bmatrix} x_{K1} \\ x_{K2} \\ x_{K3} \end{bmatrix}$
Neurofuzzy Filter
Neurofuzzy Filter
Neurofuzzy Filter
Y1
Y2
YK
Defuzzifier
Defuzzifier
Defuzzifier
D1
D2
DK
Postprocessor
$\hat{y}(r,c)$

Figure 3.1 Overview of the complete IT2 FLS composed by K neurofuzzy filters [204].

$x(r-1,c-1)$	$x(r-1,c)$	$x(r-1,c+1)$
$x(r,c-1)$	$x(r,c)$	$x(r,c+1)$
$x(r+1,c-1)$	$x(r+1,c)$	$x(r+1,c+1)$

Figure 3.2 Filtering window for neurofuzzy filters, where r and c are the indices of rows and columns, e.g., $x(r,c)$ is the luminance value of the pixel located at row r and column c [204].

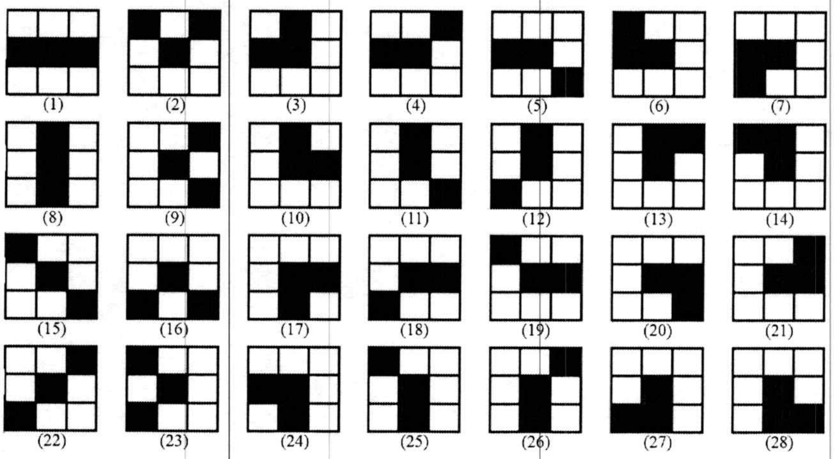


Figure 3.3 28 possible pixel topologies for the neurofuzzy filters [204].

3.2 Noise Removal based on Fuzzy Logic Systems

3.2.1 Neurofuzzy Filters

Our complete FLS has the general structure depicted in Fig. 3.1. It is composed of *K* neurofuzzy filters, each of which can be viewed as a sub-FLS, with the same structure, i.e., the same number of antecedents, the same number of rules, and same type of membership functions; but, each uses a different input data set from the given noisy image, and, therefore, has different values for the design parameters. Each data set is obtained by shifting a 3 × 3 filtering window (see Fig. 3.2) through all pixels (excluding boundary pixels) in the image. Each time the window centers at a pixel, three of the nine pixels in the filtering window are selected to form an input vector, $\boldsymbol{x}_i = [x_{i1} \quad x_{i2} \quad x_{i3}]^T$; for the i^{th} neurofuzzy filter, and this is done $\max\{i\} = K$ times.

Observe, in Fig. 3.3, that there are a total of 28 possible combinations of any three pixels in a filtering window, i.e., there are $K = 28$ neurofuzzy filters. To give a specific example, focus on the 7th neurofuzzy filter that corresponds to pixel topology (7) in Fig. 3.3, and assume that the filtering window is centered at the pixel located at row *r* and column *c* (see Fig. 3.2); then, the input vector for this neurofuzzy filter at this location is $\boldsymbol{x}_7 = [x_{71} \quad x_{72} \quad x_{73}]^T = [x(r+1, c-1) \quad x(r, c-1) \quad x(r, c)]^T$. The output of this neurofuzzy filter is an interval of numbers of the de-noised luminance value at location (*r*,*c*), $Y_7(r,c) = [\underline{y}_7(r,c), \bar{y}_7(r,c)]$. How the neurofuzzy filter actually obtains such an interval of numbers is described in detail in the next section. A point-value for pixel (*r*,*c*), $D_7(r,c)$, is then computed, as:

$$D_7(r,c) = \frac{\underline{y}_7(r,c) + \overline{y}_7(r,c)}{2} \tag{3-1}$$

Similar operations are carried out by all other neurofuzzy filters, each using an input vector obtained based on the corresponding pixel topology. This leads to K = 28 defuzzified values, $D_1(r,c)$, $D_2(r,c)$,..., $D_K(r,c)$. A post-processor block takes these defuzzified values and performs the following computations to obtain the final estimation, $\hat{y}_7(r,c)$, of the noise-free luminance value at location (r,c):

$$D_{AV}(r,c) = \frac{1}{K}\sum_{i=1}^{K} D_i(r,c) \tag{3-2}$$

$$\hat{y}(r,c) = \begin{cases} 0 & \text{if } D_{AV}(r,c) < 0 \\ 255 & \text{if } D_{AV}(r,c) > 255 \\ \text{round}[D_{AV}(r,c)] & \text{otherwise} \end{cases} \tag{3-3}$$

Note that, in practice, all the neurofuzzy filters can be computed in parallel to save computation time.

3.2.2 Non-singleton Interval Type-2 Fuzzy Logic System

Type-2 (T2) FSs [123, 125, 227] have been shown to be more capable of modeling uncertainties than are T1 FSs. As a special case of T2 FSs, IT2 FSs have been widely used in FLS design for various applications [8, 10, 117, 128, 152]. On the other hand, a Non-Singleton (NS) FLS [122] allows users to account for the uncertainties intrinsic to the input measurements, and gives the users more design degree of freedom by modeling input values as FSs instead of crisp numbers.

In this section, each of the neurofuzzy filters is a T1 NS-IT2 FLS, where the input values are modeled as T1 FSs. Without loss of generality, focus on the i^{th} (i = 1, 2,..., K) neurofuzzy filter to obtain an interval estimate, $Y_i(r,c)$, of the pixel value at location (r,c). Recall that the input vector is $\boldsymbol{x}_i = [x_{i1} \quad x_{i2} \quad x_{i3}]^T$, which is obtained by collecting three pixel values in the filtering window based on the corresponding topology. Our proposed NS IT2 FLS is based on N rules. Note that N is different from K in the previous section. K denotes the total number of neurofuzzy filters in our complete FLS, whereas each neurofuzzy filter has exactly N rules. Choosing an appropriate value for N is quite important: if N is too small, the neurofuzzy filter's ability to model uncertainties is going to be limited, which leads to poor performance; and if N is too large, the computational cost is going to be very high. There does exist a standard procedure for choosing an optimal value for N; so, generally, N is chosen in an ad hoc manner. In this chapter, based on our tests and trials, N is chosen to be $3^3 = 27$. Each rule in the NS IT2 FLS take the following forms:

Rule 1: IF $\tilde{X}_{i1}$ is $\tilde{F}_{i1}^1$ and $\tilde{X}_{i2}$ is $\tilde{F}_{i2}^1$ and $\tilde{X}_{i3}$ is $\tilde{F}_{i3}^1$, THEN $y_i = \tilde{G}_i^1$

Rule 2: IF $\tilde{X}_{i1}$ is $\tilde{F}_{i1}^2$ and $\tilde{X}_{i2}$ is $\tilde{F}_{i2}^2$ and $\tilde{X}_{i3}$ is $\tilde{F}_{i3}^2$, THEN $y_i = \tilde{G}_i^2$

...

Rule N: IF $\tilde{X}_{i1}$ is $\tilde{F}_{i1}^N$ and $\tilde{X}_{i2}$ is $\tilde{F}_{i2}^N$ and $\tilde{X}_{i3}$ is $\tilde{F}_{i3}^N$, THEN $y_i = \tilde{G}_i^N$

where the consequent of each rule, $\tilde{G}_i^k$ $(k = 1, \dots, N)$, is an IT2 FS whose centroid [85, 122] is the interval $[y_{il}^k, y_{ir}^k]$; and each antecedent, $\tilde{F}_{i2}^k$ $(j = 1,2,3$ and $k = 1, \dots, N)$, is an IT2 FS described by a *Lower Membership Function* (LMF) and an *Upper Membership Function* (UMF) that have the following forms, respectively $(u \in X)$:

$$\underline{\mu}_{ij}^k(u) = \begin{cases} N(\overline{m}_{ij}^k, \sigma_{ij}^k; u) & \text{if } u \le \dfrac{\underline{m}_{ij}^k + \overline{m}_{ij}^k}{2} \\ N(\underline{m}_{ij}^k, \sigma_{ij}^k; u) & \text{if } u > \dfrac{\underline{m}_{ij}^k + \overline{m}_{ij}^k}{2} \end{cases} \tag{3-4}$$

$$\underline{\mu}_{ij}^k(u) = \begin{cases} N(\underline{m}_{ij}^k, \sigma_{ij}^k; u) & \text{if } u < \underline{m}_{ij}^k \\ 1 & \text{if } \underline{m}_{ij}^k \le u \le \overline{m}_{ij}^k \\ N(\overline{m}_{ij}^k, \sigma_{ij}^k; u) & \text{if } u > \overline{m}_{ij}^k \end{cases} \tag{3-5}$$

where, e.g., $N\left(\underline{\mu}_{ij}^k, \sigma_{ij}^k; u\right) = \exp\left[-\frac{1}{2}(u - \underline{m}_{ij}^k/\sigma_{ij}^k)^2\right]$. Examples of the three antecedent IT2 FSs of the k^{th} rule are depicted by the gray areas in Fig. 3.4, each of which is called a *Footprint of Uncertainty* (FOU).

The input T1 FS, $\tilde{X}_{ij}$ ($j = 1{,}2{,}3$), for the corresponding antecedent has the following MF ($u \in X$):

$$\mu_{ij}^k(u) = N(x_{ij}, \sigma_{\tilde{X}_{ij}}^k; u) \tag{3-5}$$

where the mean of the MF is the actual crisp input value x_{ij}. Examples of the three input T1 FSs are depicted by the dotted lines in Fig. 3.4.

Details regarding the above rule-based inference system can be found in Chapter 11 of [122]. This section briefly reviews all the necessary steps of the computations.

First, the firing interval for each antecedent in a rule, $[\underline{f}_{ij}^k, \overline{f}_{ij}^k]$, is computed, where $\underline{f}_{ij}^k$ ($\overline{f}_{ij}^k$) corresponds to the maximum value of the fuzzy intersections between the LMF (UMF) of $\tilde{F}_{ij}^k$ and the MF of X_{ij}, respectively, i.e.:

$$\underline{f}_{ij}^k = \max_{u\in X}\left\{\min\left[\underline{\mu}_{ij}^k(u), \mu_{ij}^k(u)\right]\right\} \tag{3-7}$$

$$\overline{f}_{ij}^k = \max_{u\in X}\left\{\min\left[\overline{\mu}_{ij}^k(u), \mu_{ij}^k(u)\right]\right\} \tag{3-8}$$

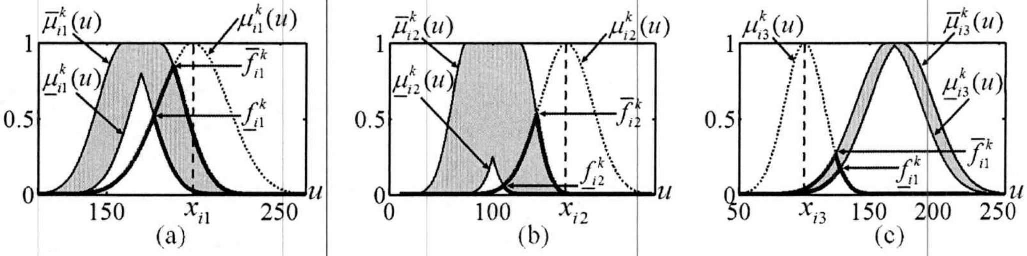
An example of the fuzzy intersections between $\underline{\mu}_{ij}^k(u)$ $[\overline{\mu}_{ij}^k(u)]$ and $\mu_{ij}^k(u)$ for the k^{th} rule are depicted by the heavy dashed (solid) lines in Fig. 3.4, where the corresponding $\underline{f}_{ij}^k$ and $\overline{f}_{ij}^k$ values are also labeled.

Once the firing interval of each antecedent becomes available, the firing interval for each rule, $[\underline{f}_{ij}^k, \overline{f}_{ij}^k]$ $(k = 1,2,\dots,N)$, is obtained, as:

$$f_{il}^k = \min_{j\in\{1,2,3\}} \underline{f}_{ij}^k \tag{3-9}$$

$$f_{ir}^k = \min_{j\in\{1,2,3\}} \overline{f}_{ij}^k \tag{3-10}$$

Finally, the Karnik-Mendel (KM) algorithm [85, 122] or the Enhanced KM (EKM) algorithm [190] is used to compute the following two values:



(a) (b) (c)

Figure 3.4 An example of the LMFs and UMFs, $\underline{\mu}_{ij}^k(u)$ and $\overline{\mu}_{ij}^k(u)$, of the three antecedent IT2 FSs, $F_{ij}^k (j = 1,2,3)$; the MFs, $\mu_{ij}^k(u)$, of the corresponding input T1 FSs , $\tilde{X}_{ij}^k$ $(j = 1,2,3)$; and, the fuzzy intersections between $\underline{\mu}_{ij}^k(u)$ $[\overline{\mu}_{ij}^k(u)]$ and μ_{ij}^k $(j = 1,2,3)$, where (a) $j = 1$; (b) $j = 2$; (c) $j = 3$.

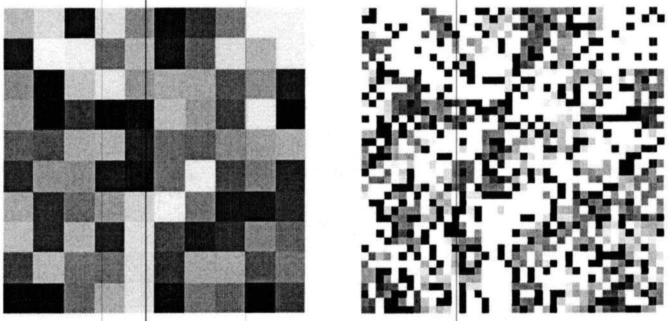
$$\underline{y}_i(r,c) = \min_{f^k \in [f_{il}^k, f_{ir}^k]} \frac{\sum_{k=1}^{N} f^k y_{il}^k}{\sum_{k=1}^{N} f^k} \tag{3-11}$$

$$\overline{y}_i(r,c) = \max_{f^k \in [f_{il}^k, f_{ir}^k]} \frac{\sum_{k=1}^{N} f^k y_{il}^k}{\sum_{k=1}^{N} f^k} \tag{3-12}$$

which are the desired outputs of the i^{th} neurofuzzy filter, namely, $Y_i(r,c) = [\underline{y}_i(r,c), \bar{y}_i(r,c)]$ at location (*r*,*c*).

For the above i^{th} sub-system, each antecedent IT2 FS has three parameters, $\underline{m}_{ij}^k$, $\overline{m}_{ij}^k$ and σ_{ij}^k, each input T1 FS has one parameter, $\sigma_{X_{ij}}^k$, and each consequent has two parameters, y_{il}^k and y_{ir}^k; therefore, there are $3 \times (3+1) + 2 = 14$ parameters for each rule. We chose the number of rules to be $N = 3^3 = 27$; consequently, the total number of parameters of each neurofuzzy filter is $N \times 14 = 378$. Note that the K neurofuzzy filters can be obtained separately in parallel; also, these parameters need only to be optimized once, and then can be stored and used for different images.

The training image set, which is used to tune the parameters of the neurofuzzy filters, consists of an original clean image and its corrupted version, both of which are generated by a computer. The original image, depicted in Fig. 3.5(a), is 40×40 pixels and is divided into $10 \times 10 = 100$ boxes, i.e., each box contains $4 \times 4 = 16$ pixels. Pixels in each box have the same luminance value, an 8-bit integer uniformly distributed between 0 and 255.



(a) (b)

Figure 3.5 (a) original image, and (b) image corrupted by 30% AWGN and 50% impulse noise.

The corrupted image was obtained by adding MGIN to the original image. After extensive simulations, we used 30% AWGN and 50% impulse noise to contaminate the clean image [see Fig. 3.5(b)].

To train the i^{th} neurofuzzy filter, the filtering window was shifted through all pixel locations (excluding the boundary pixels), i.e., $r = 2,3,\ldots,R-1$ and $c = 2,3,\ldots,C-1$, where R and C are the total number of rows and columns of the pixels in the given image; and, a set of input vectors, $\boldsymbol{x}_i$, was collected. The rule-based inference system produced a set of $D_i(r,c)$ values ($r = 2,3,\ldots,R-1$ and $c = 2,3,\ldots,C-1$), which can be viewed as the i^{th} neurofuzzy filter's estimates of the original pixel values at corresponding locations. Note that all the neurofuzzy filters were trained independently; therefore, the post-processer block was not involved in this stage, namely, $D_{AV}(r,c)$ was not computed during the training stage. The "optimal" set of design parameters was obtained by optimizing the following Mean Square Error (MSE) of the estimated pixel values:

$$MSE = \frac{1}{(R-2)(C-1)} \sum_{r=2}^{R-1} \sum_{c=2}^{C-1} [D_i(r,c) - y(r,c)]^2 \qquad (3\text{-}13)$$

where $y(r,c)$ is the original noise-free pixel value at (r,c).

Comment: When this chapter was published as a conference paper [227], a reviewer suggested that it is not very clear whether different images would need different training images. In response, we want to point out that only one training image set (shown in Fig. 3.5) is needed to obtain a NS IT2 FLS that is universally applicable to different images.■

3.3 Particle Swarm Optimization

One category of the well-known techniques that can be employed to tune the parameters of the above neurofuzzy filters (NS-IT2 FLSs) is the *gradient-based* methods [13, 56] proposed by Liang and Mendel [103, 122]. The analytical forms of the gradients of such a NS-IT2 FLS that has many parameters and nonlinear computations are extremely complicated, making their computer implementation very difficult, which has been the main reason that has prevented NS-IT2 FLSs from becoming widely used. Also, the performance of the system can suffer significantly if the solution to minimizing the MSE quickly falls into a local minimum, which can easily occur when gradient-based methods are used.

In recent years, *population-based random optimization* techniques, such as *Particle Swarm Optimization* (PSO) [88], *Genetic Algorithm*} (GA) [63] and *evolutionary programming* (EP) [57], have been widely applied to T1 FLS designs [39, 40, 58, 75, 81, 82, 107]. In this chapter, we advocate the use of a PSO algorithm when searching for the "optimal" design parameters, because it is computationally fast, very easy to implement in computer programs, and, as will be seen in Section 3.5, it enables our system to provide very good results.

3.3.1 Quick Review of PSO

The standard PSO algorithm, first introduced by Kennedy and Eberhart [88], is briefly reviewed in this section. Note that the notations and labeling in this section are different from those in the previous section, because they are completely independent and denote different things here.

Let M denote the swarm (population) size, where each particle in the swarm represents a possible solution to the optimization problem; n denote the dimension of the search space; and, G denote the total number of generations (iterations) of the optimization process.

Each particle i $(i = 1,2,\dots,M)$ has three inherent attributes: 1) a current position $\underline{\boldsymbol{X}}_i = (X_{i,1}, X_{i,2}, \dots, X_{i,n})$ in the search space; 2) a current velocity $\underline{\boldsymbol{V}}_i = (V_{i,1}, V_{i,2}, \dots, V_{i,n})$; and 3) a personal best (*pbest*) position (the position that produces the minimal value in the history of the particle) $\underline{\boldsymbol{P}}_i = (P_{i,1}, P_{i,2}, \dots, P_{i,n})$, i.e.:

$$\underline{P}_i(t+1) = \begin{cases} \underline{P}_i(t) & \text{if } f_{obj}(\underline{X}_i(t+1)) \geq f(\underline{P}_i(t)) \\ \underline{X}_i(t+1) & \text{if } f_{obj}(\underline{X}_i(t+1)) < f(\underline{P}_i(t)) \end{cases} \tag{3-14}$$

where $t = 1,2,\dots,G-1$ is the generation index.

In each generation, each particle updates its velocity as follows $(j = 1,2,\dots,n;$ $i = 1,2,\dots,M)$:

$$\begin{aligned} V_{i,j}(t+1) = w \cdot V_{i,j}(t) + c_1 \cdot r_{1,i}(t)[P_{i,j}(t) - X_{i,j}(t)] \\ + c_2 \cdot r_{2,i}(t)[P_{g,j}(t) - X_{i,j}(t)] \end{aligned} \tag{3-15}$$

where $X_{i,j}(t)$ is the j^{th} element of position of the i^{th} particle in the t^{th} generation; c_1 and c_2 are two constants called *acceleration coefficients*; $r_{1,i}(t)$ and $r_{2,i}(t)$ are two random variables uniformly distributed in the interval [0,1]; w is called *inertia weight* that is usually set to decrease linearly from 0.9 to 0.4 during the course of the search process to help PSO reach convergence; and, $\underline{P}_g(t)$ denotes the global best (*gbest*) position found in the history of the entire swarm, i.e. $(i = 1,2,\ldots, M)$:

$$\underline{P}_g(t) = \arg\min_{\underline{P}_i(t)} f_{obj}(\underline{P}_i(t)) \tag{3-16}$$

At the end of each generation, a new position of a particle is obtained, as:

$$\underline{X}_i(t+1) = \underline{X}_i(t) + \underline{V}_i(t) \tag{3-17}$$

3.3.2 Quantum-Behaved Particle Swarm Optimization

Despite its exceptionally good performance on many different problems, Van den Bergh [11] has shown that the PSO algorithm does not guarantee global optimization; therefore, ever since its debut in 1995, many variations and modifications of the PSO algorithm have been proposed to enhance its performance [97, 173, 232]. Recently, Sun et al [173, 195] proposed a Quantum-behaved PSO (QPSO) algorithm that, theoretically, guarantees optimal solution in the search space, and, they have shown that, in practice, the QPSO algorithm provides better results for many widely-used benchmark tests than does the standard PSO algorithm. Therefore, this particular version of the PSO algorithm is our choice. Its pseudo-code is [195]:

```
**initialize** $\underline{X}_i(1)$ $(i = 1,2, \ldots M)$ randomly
**set** $\underline{P}_i(1) = \underline{X}_i(1)$ $(i = 1,2, \ldots M)$
**for** $t = 1$ to $G$-1 **do**
    calculate $\underline{m}(t) = \frac{1}{M} \sum_{i=1}^{M} \underline{P}_i(t)$
    $\underline{P}_g(t) = \text{argmin} f_{obj}(\underline{P}_i(t))$
    **for** $i = 1$ to $M$ **do**
        **if** $f(\underline{X}_i(t)) < f(\underline{P}_i(t))$ **then** $\underline{P}_i(t) = \underline{X}_i(t)$
        **end if**
        **for** $j = 1$ to $n$ **do**
            $\eta = \text{rand}(0,1)$
            $\underline{P}_{i,j}(t+1) = \eta \times \underline{P}_{i,j}(t) + (1-\eta) \times \underline{P}_{g,j}(t)$
            $u = \text{rand}(0,1)$
            **if** rand(0,1) > 0.5 **then**
                $\underline{X}_{i,j}(t+1) = \underline{P}_{i,j}(t+1) - \beta \left| m_j(t) - \underline{X}_{i,j}(t) \right| \ln \frac{1}{\mathrm{u}}$
            **else**
                $\underline{X}_{i,j}(t+1) = \underline{P}_{i,j}(t+1) + \beta \left| m_j(t) - \underline{X}_{i,j}(t) \right| \ln \frac{1}{\mathrm{u}}$
            **end if**
        **end for**
    **end for**
**end for**
```

Observe that the QPSO algorithm differs from the PSO algorithm mainly in two ways: 1) instead of using (3-14), the best personal position of each particle is updated by taking a weighted average of its previous best personal and the global positions:

$$\underline{P}_{i,j}(t+1) = \eta \times \underline{P}_{i,j}(t) + (1-\eta) \times \underline{P}_{g,j}(t) \qquad \text{(3-18)}$$

where η is a random variable uniformly distributed in (0,1]; and 2) instead of using (3-17), the location of each particle is updated as:

$$\underline{X}_{i,j}(t+1) = \underline{P}_{i,j}(t+1) \pm \beta\left|m_j(t) - \underline{X}_{i,j}(t)\right|\ln\frac{1}{\mathrm{u}} \qquad \text{(3-19)}$$

where $m(t)$ is the average best personal position of the entire swarm and u is also a random variable uniformly distributed in (0,1]. Note that (3-19) was developed based on the solution of the Schrödinger equation, which is why the algorithm has the prefix "Quantum-behaved".

3.4 Additional Enhancement by BM3D and Contrast Scaling

As will be seen in Section 3.5, the complete NS-IT2 FLS explained so far significantly outperforms all other non-fuzzy methods as well as its T1 and S-IT2 counterparts quantitatively; however, when evaluated from a visual perspective, it has the unwanted feature that the contrasts of the de-noised image are reduced. We observed that the IPAMF+BM3D method proposed by Yang and Wu [203] does a relatively good job of maintaining the contrast level after its noise removal process. This inspired our efforts to try to integrate the outcomes of the NS-IT2 FLS and the IPAMF+BM3D. Unfortunately, we observed that any MSE-based nonlinear fusion technique, such as Choquet Integral [130], always favors the result of the NS-IT2 FLS and ignores that of the IPAMF+BM3D, merely because the NS-IT2 FLS always has a much better MSE than does the IPAMF+BM3D. On the other hand, if we simply combine the results of the two methods by giving them equal weight (50%), the combined result has much better visual quality, but its MSE can become worse than some of the non-fuzzy techniques.

We also tested how cascading the NS-IT2 FLS with IPAMF and/or BM3D could affect the results. Through our simulations, we found that sequential processing of the noisy image, first by the NS-IT2 FLS and then by the BM3D DCT [50] filter (IPAMF was found not to be needed), allowed the overall system to gain a certain amount of improvement, both visually and for the MSE. This indicated that the BM3D filter has the ability to clean up some residual noise in the image after it is processed by the NS-IT2 FLS; however, although the BM3D filter can maintain the contrast level of the image it processes, it does nothing to bring back the lost contrasts. In other words, such an NS-IT2 FLS plus BM3D filtering achieves better restoration of the image compared to just the NS-IT2 FLS, but the contrast level of the outcome is still low.

Our goal of raising back the contrasts of the de-noised image turned out to be attainable by simply performing a linear contrast scaling operation to the de-noised image, the details of which are reviewed below. The restored contrast level leads to significantly enhanced visual quality. Sometimes, appending such a contrast scaling filter can cause a limited amount of negative effects to the MSE; however, the MSE is still much better than those of all the non-fuzzy techniques.

To summarize, including BM3D and contrast scaling filters in series with the NS-IT2 FLS, as depicted in Fig. 3.6, enabled us to outperform all existing methods by a significant margin both quantitatively (MSE) and visually. All the simulation results that support these claims are provided in Section 3.5.

Comment: In the remainder of this chapter, we denote the NS-IT2 FLS followed by just the BM3D filter as NS-IT2 FLS+, and the NS-IT2 FLS followed by the BM3D filter and the contrast scaling filter as NS-IT2 FLS++.■

3.4.1 Block-Matching 3-Dimensional (BM3D) Discrete Cosine Transformation (DCT) Filter

The BM3D DCT filter is very well studied in [50]; its computer implementation has been made public available online by its authors. Here we only briefly go over the basic idea and structure of this technique.

The BM3D DCT filter is designed to combat AWGN, and is based on an enhanced sparse transform-domain representation, which is achieved by grouping similar 2D image blocks into 3D arrays called *groups*. Then three-step collaborative filtering is applied to these groups, namely: 1) 2D DCT and 1D Haar wavelet transformation; 2) shrinking of the transformed spectrum; and, 3) inverse 3D transformation. Such collaborative filtering has been shown to attenuate noise, preserve the finest details shared by different blocks, and preserve the unique features of each individual block. The outcome of this collaborative filter is a 3D estimate of these jointly-filtered image blocks which overlap among themselves. Finally, an aggregation method that takes the advantages of the redundancy is used to combine the estimates for each pixel.

3.4.2 Linear Contrast Scaling Filter

Without loss of generality, assume the range of the pixel values in the original noise-free image is [*a*,*b*], where $0 \leq a \leq b \leq 255$. After corruption by MGIN and processing by the NS-IT2 FLS, pixel values always occupy a smaller range [*c*,*d*] within [*a*,*b*], i.e., $a < c$ and $d < b$. This results in a reduced contrast level of the de-noised image. To enhance the contrasts back to their original level, i.e., to map pixels in [*c*,*d*] back to [*a*,*b*], one needs knowledge of the exact values of *a* and *b*, which is usually not available in practice. In this chapter, it is assumed that [*a*,*b*] covers most of [0,255]; so, it is reasonable to simply map [*c*,*d*] back to the full range [0,255]. This works very well for the widely-used test images examined in Section 3.5. In cases where [*a*,*b*] only covers a small portion of [0,255], one can estimate the values of *a* and *b*, and then our proposed strategy still applies.

The possibilities of scaling an output image back into the domain of values occupied by the original image usually include linear image scaling (also known as *linear contrast scaling filter*), linear image scaling with clipping, and absolute value scaling [149]. In our case, clipping is already carried out by (3-3), which ensures that *c* is never negative and *d* never goes over 255, so no additional clipping or absolute value operations are needed here. As a result, one can apply the standard linear contrast scaling filter that is illustrated in the third box in Fig. 3.6, in which every input pixel value $x \in [c, d]$ is mapped into an output pixel value $y \in [0,255]$ via:

$$y = \frac{255}{d-c} \times (x-c) \tag{3-20}$$

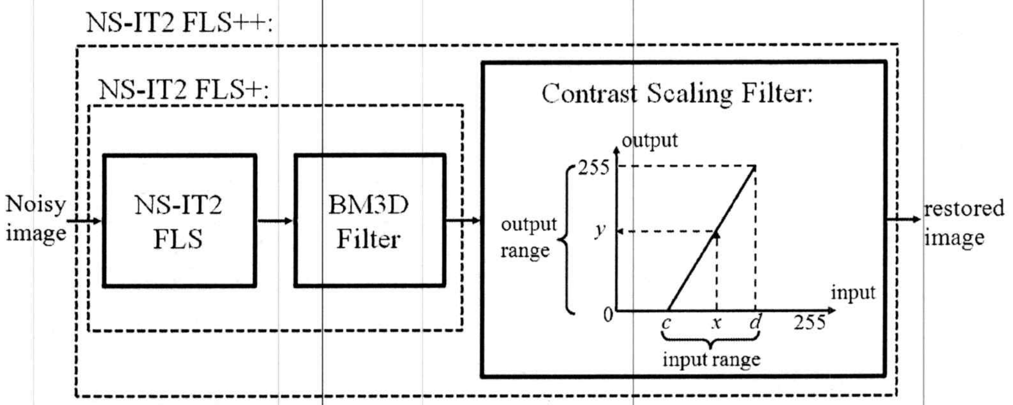


Figure 3.6 The sequential structure of the NS-IT2 FLS+ and NS-IT2 FLS++ approaches.

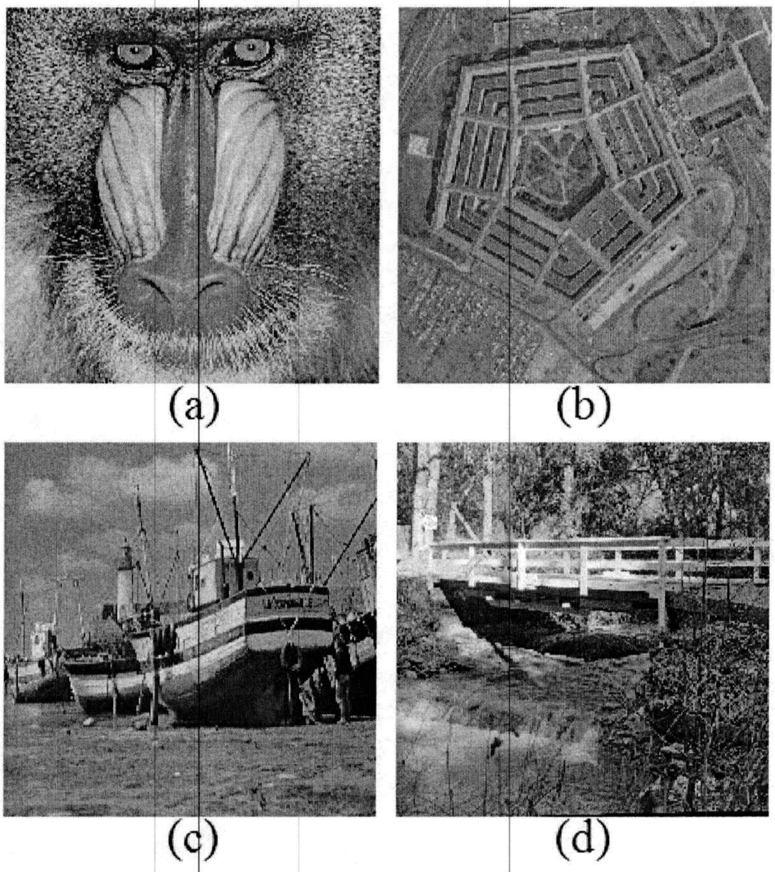


Figure 3.7 Four benchmark test images: (a) baboon; (b) pentagon; (c) boat; (d) bridge.

We do not yet have a way to couple this contrast mapping equation into the MSE equation (3-13); so, while it is possible to remedy the contrast level problem, it is sometimes inevitable to see some downgrades of the MSE, because higher contrast level does not necessarily always correspond to lower MSE. An example in Section 3.5 demonstrates that the IPAMF+BM3D method produces results with higher contrasts, but also with higher MSEs, as compared to the results from the NS-IT2 FLS. Fortunately, the deterioration of the MSE resulting from the contrast scaling operation is very limited, and can generally be ignored.

3.5 Benchmark Testing

The four widely-used benchmark images in Fig. 3.7 were used by us to test the performance of our NS-IT2 FLS and its augmented versions, NS-IT2 FLS+ and NS-IT2 FLS++. These results were compared with five non-fuzzy techniques (median filter, Wiener filter, Gaussian filter, ROAD [61], and IPAMF+BM [203]). We also examined the T1 and S-IT2 counterparts of these NS-IT2 systems in an effort to see whether modeling the inputs as NS-IT2 FSs was helpful. Three different levels of MGIN were generated: 30% AWGN combined with 30%, 50%, and 70% impulse noise, respectively.

3.5.1 Quantitative Analysis

The MSEs of all the above-mentioned techniques for the four images under different noise levels are summarized in Table 3.1, based on which the following observations can be made:

1. The non-augmented FLSs (T1, S-IT2 and NS-IT2 FLSs) significantly outperformed all the non-fuzzy techniques. Among these FLSs, the S-IT2 FLS outperformed the T1 FLS, and the NS-IT2 FLS outperformed the S-IT2 FLS. The improvements from S-IT2 FLS to NS-IT2 FLS are always much greater than those from T1 FLS to IT2 FLS (see Table 3.2). These results demonstrate that there is potential for much improvement if one models the uncertain inputs as S-IT2 FSs over T1 FSs, and even greater improvement if one uses NS-IT2 FSs over S-IT2 FSs.
2. The FLSs augmented with only BM3D (T1 FLS+, S-IT2 FLS+ and NS-IT2 FLS+) always outperform their non-augmented counterparts.
3. The T1 FLS++, S-IT2 FLS++ and NS-IT2 FLS++ can sometimes outperform T1 FLS+, S-IT2 FLS+ and NS-IT2 FLS+, respectively, e.g., see the columns for "Boat" and "Bridge"; however, their performances can also become worse in some cases, e.g., "Baboon" and "Pentagon" when corrupted by 30% AWGN combined with 70% impulse noise.
4. T1 FLS++, S-IT2 FLS++ and NS-IT2 FLS++ always perform better than their non-augmented counterparts, except for "Pentagon" corrupted by 30% AWGN combined with 30% impulse noise, for which the NS-IT2 FLS++ has a slightly larger MSE than that of the NS-IT2 FLS [(544.19-538.88)/544.19=0.98%].
5. T1 FLS++, S-IT2 FLS++, and NS-IT2 FLS++ always outperform all non-fuzzy techniques by a significant amount.

Table 3.1 MSEs of different approaches for the four benchmark images (baboon, pentagon, boat, and bridge) corrupted by 30% AWGN combined with 30%, 50%, and 70% impulse noise, respectively.

	Baboon			Pentagon			Boat			Bridge		
Approach	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
Median Filter	5943	7174	10^4	6164	7242	9934	6043	7232	10^4	5935	7260	10^4
Wiener Filter	4016	3359	3222	4026	2962	2376	4271	3531	3304	4861	4468	4401
Gaussian	5321	5943	6902	5445	5605	6058	5676	6168	7003	6153	6899	7935
ROAD	3085	4210	5379	3286	4899	5571	2844	3929	4965	2879	3935	4953
IPAMF+ BM3D	5953	6720	9462	6575	7542	9215	6069	6753	8592	5992	8096	10^4
T1 FLS	1937	2214	2572	1851	1800	1814	2125	2383	2779	2817	3338	4006
S-IT2	1699	2063	2499	1434	1503	1600	1874	2214	2638	2479	3055	3737
NS-IT2	1084	1526	1989	539	711	897	1087	1501	1972	1574	2203	2874
T1 FLS+	1699	1747	1916	1493	1257	1121	1742	1810	2066	2512	2844	3343
S-IT2+	1391	1542	1819	1013	916	889	1432	1595	1903	2126	2511	3046
NS-IT2+	1016	1326	1670	322	413	529	853	1175	1577	1407	1920	2494
T1 FLS++	1742	1463	1975	1104	825	1327	1369	1624	1808	1760	1748	2511
S-IT2++	1106	1337	2262	669	829	1413	771	905	1659	801	1103	2004
NS-IT2++	840	938	1766	544	591	772	665	623	1091	586	767	1439

Table 3.2 Percentage improvements from T1 to S-IT2 to NS-IT2 FLSs for the four benchmark images (baboon, pentagon, boat, and bridge) corrupted by 30% AWGN combined with 30%, 50%, and 70% impulse noise, respectively

	Baboon			Pentagon			Boat			Bridge		
	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
T1 to S-IT2	12.3	6.82	2.84	22.5	16.5	11.8	11.8	7.08	5.10	12.0	8.48	6.72
S-IT2 to NS-IT2	36.2	26.0	20.4	62.4	52.7	43.9	42.0	32.2	25.2	36.5	27.9	23.1

3.5.2 Visual Quality Analysis

Figs. 3.8 - 3.10 present, respectively, the images “Baboon”, “Pentagon” and “Bridge” corrupted by 30% AWGN combined with 30% impulse noise and their restored versions based on all the aforementioned techniques. Figs. 3.11 - 3.13 present the same kind of results when the images are corrupted by 30% AWGN combined with 50% impulse noise. Results for these two noise levels are very representative; those for 30% AWGN combined with 70% impulse noise lead to similar images, and, therefore, they are not included here. Also, the

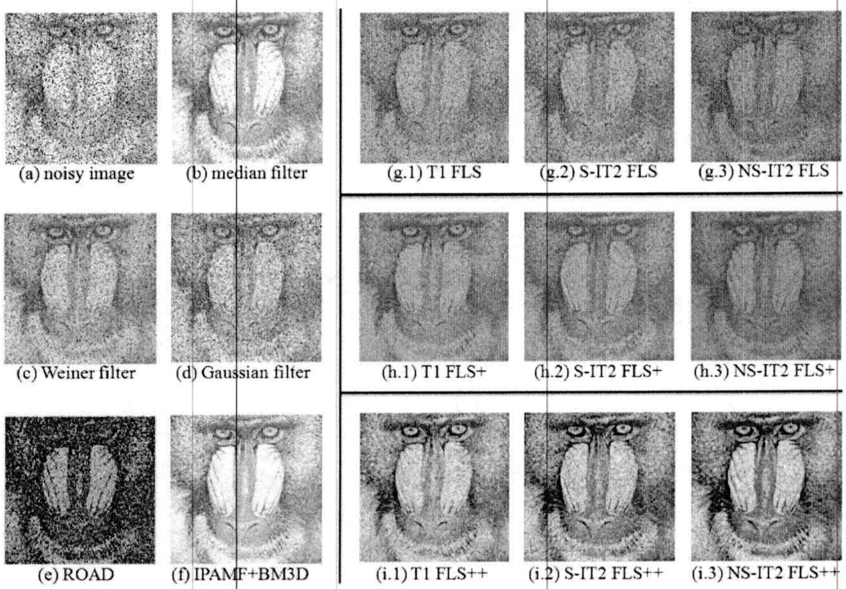


Figure 3.8 (a) Image "baboon" corrupted by 30% AWGN combined with 30% impulse noise; and its restored versions based on (b)-(f) non-fuzzy techniques; (g.1)-(g.3) FLSs without BM3D or contrast scaling; (h.1)-(h.3) FLSs with just the BM3D; and (i.1)-(i.3) FLSs with both BM3D and contrast scaling.

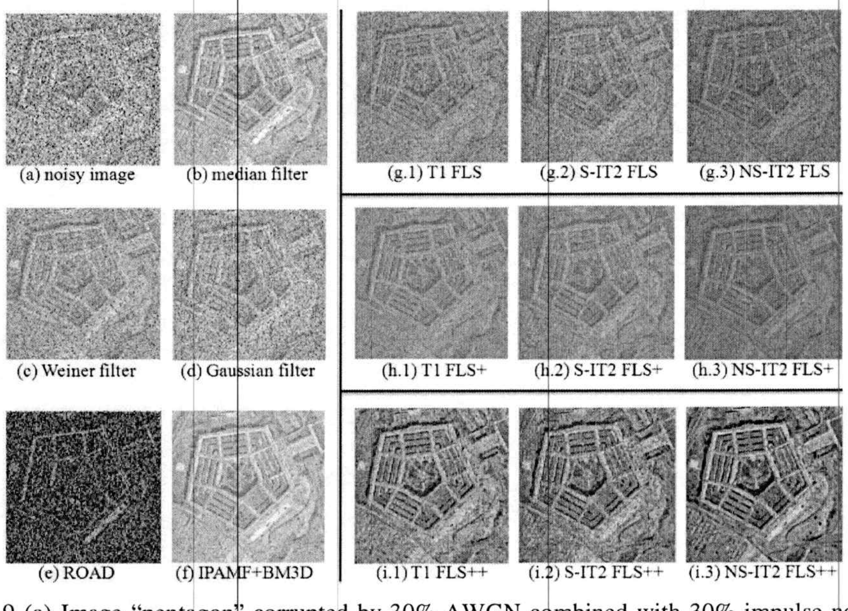


Figure 3.9 (a) Image "pentagon" corrupted by 30% AWGN combined with 30% impulse noise; and its restored versions based on (b)-(f) non-fuzzy techniques; (g.1)-(g.3) FLSs without BM3D or contrast scaling; (h.1)-(h.3) FLSs with just the BM3D; and (i.1)-(i.3) FLSs with both BM3D and contrast scaling.

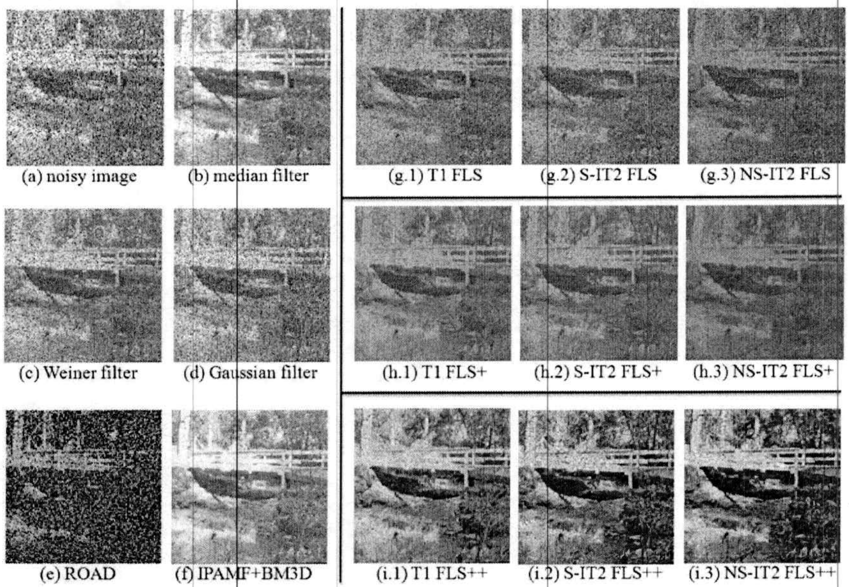


Figure 3.10 (a) Image "bridge" corrupted by 30% AWGN combined with 30% impulse noise; and its restored versions based on (b)-(f) non-fuzzy techniques; (g.1)-(g.3) FLSs without BM3D or contrast scaling; (h.1)-(h.3) FLSs with just the BM3D; and (i.1)-(i.3) FLSs with both BM3D and contrast scaling.

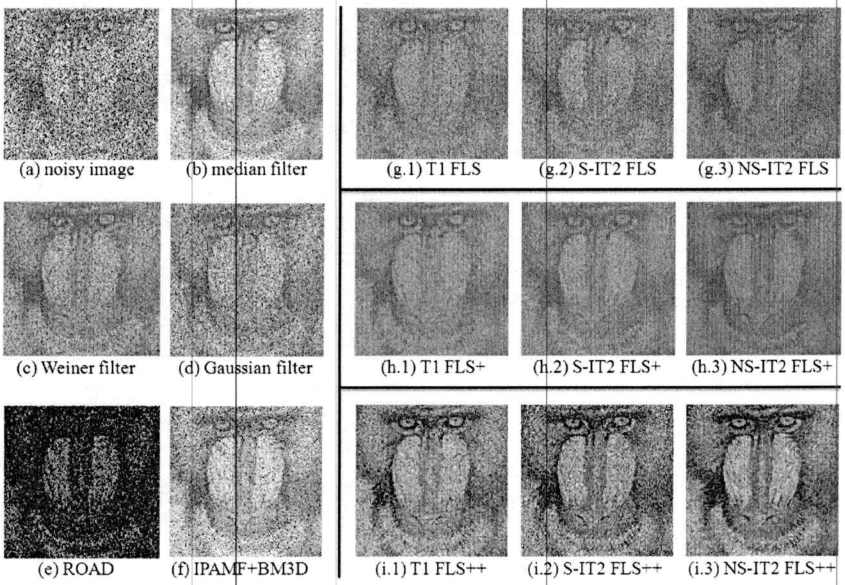
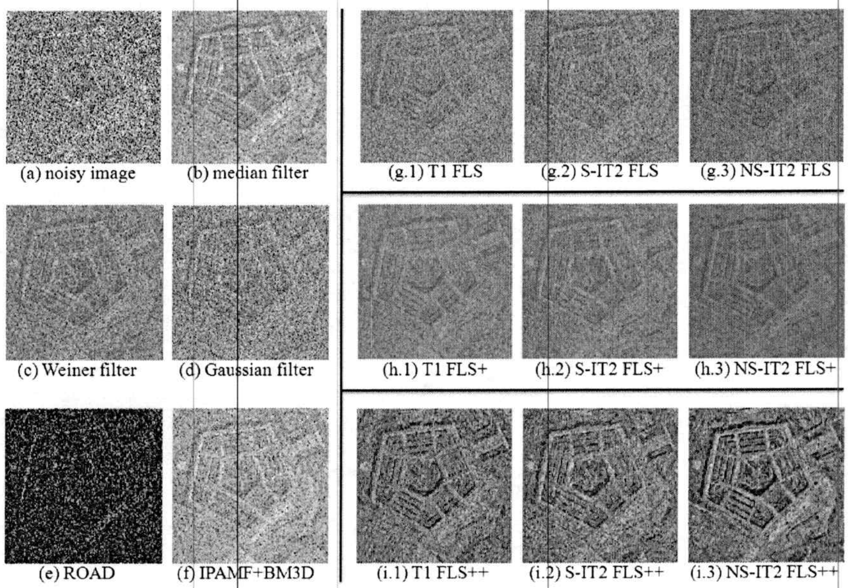


Figure 3.11 (a) Image "baboon" corrupted by 30% AWGN combined with 50% impulse noise; and its restored versions based on (b)-(f) non-fuzzy techniques; (g.1)-(g.3) FLSs without BM3D or contrast scaling; (h.1)-(h.3) FLSs with just the BM3D; and (i.1)-(i.3) FLSs with both BM3D and contrast scaling.

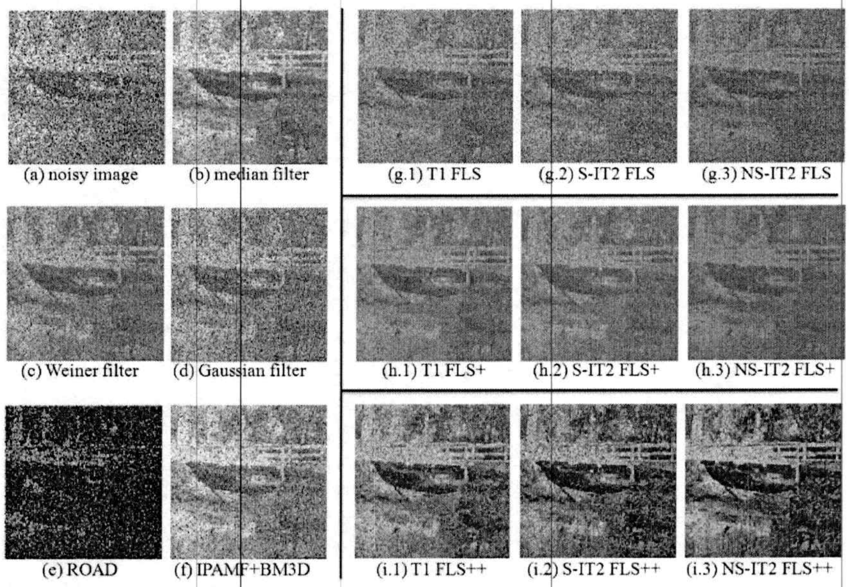


(a) noisy image (b) median filter (g.1) T1 FLS (g.2) S-IT2 FLS (g.3) NS-IT2 FLS

(c) Weiner filter (d) Gaussian filter (h.1) T1 FLS+ (h.2) S-IT2 FLS+ (h.3) NS-IT2 FLS+

(e) ROAD (f) IPAMF+BM3D (i.1) T1 FLS++ (i.2) S-IT2 FLS++ (i.3) NS-IT2 FLS++

Figure 3.12 (a) Image "pentagon" corrupted by 30% AWGN combined with 50% impulse noise; and its restored versions based on (b)-(f) non-fuzzy techniques; (g.1)-(g.3) FLSs without BM3D or contrast scaling; (h.1)-(h.3) FLSs with just the BM3D; and (i.1)-(i.3) FLSs with both BM3D and contrast scaling.



(a) noisy image (b) median filter (g.1) T1 FLS (g.2) S-IT2 FLS (g.3) NS-IT2 FLS

(c) Weiner filter (d) Gaussian filter (h.1) T1 FLS+ (h.2) S-IT2 FLS+ (h.3) NS-IT2 FLS+

(e) ROAD (f) IPAMF+BM3D (i.1) T1 FLS++ (i.2) S-IT2 FLS++ (i.3) NS-IT2 FLS++

Figure 3.13 (a) Image "bridge" corrupted by 30% AWGN combined with 50% impulse noise; and its restored versions based on (b)-(f) non-fuzzy techniques; (g.1)-(g.3) FLSs without BM3D or contrast scaling; (h.1)-(h.3) FLSs with just the BM3D; and (i.1)-(i.3) FLSs with both BM3D and contrast scaling.

image “Boat” has similar results to those of these three images, and the conference version of this chapter [227] already presented some of its results; therefore, images for “Boat” are not included here. The following observations can be made about these figures:

1. In Figs. 3.8 - 3.13, the noisy image and the restored outcomes by all non-fuzzy techniques are presented on the left-hand side of the vertical separation line. Observe that the IPAMF+BM3D method is definitely the winner in the non-fuzzy category for all these images.
2. On the right-hand side of the vertical separation line, the NS-IT2 system images always look better than their T1 and S-IT2 counterparts. This is easier to see for the higher noise level (30% AWGN combined with 50% impulse noise). It is also true that the NS-IT2 FLS+ images always look better than the NS-IT2 FLS images, and the NS-IT2++ images always look better than the NS-IT2 FLS+ images.
3. In the cases of lower noise level (30% AWGN combined with 30% impulse noise), the images for the NS-IT2 FLS look poorer than those of the IPAMF+BM3D, whereas the images for the NS-IT2 FLS+ look just as good as the images from the IPAMF+BM3D, except that the former appear to be darker than the latter, which is caused by the reduced contrast level.
4. In the cases of higher noise level (30% AWGN combined with 50% impulse noise), the images for the NS-IT2 FLS still look poorer than those of the IPAMF+BM3D, whereas the images from the NS-IT2 FLS+ clearly look better than images from the IPAMF+BM3D, although they still have relatively low contrasts. This indicates that the NS-IT2 systems are more robust in extremely noisy environments.

5. The NS-IT2 FLS++ images clearly look better than images from all other fuzzy and non-fuzzy techniques.

3.6 Summary

This chapter proposed a NS-IT2 FLS for removing Mixed Gaussian and Impulse Noises from black and white images; and advocated the use of a QPSO algorithm for determining the parameters of the FLS. Unlike some of the existing fuzzy techniques, our NS-IT2 FLS only needs to be tuned once, and then it becomes universally applicable to different images, which saves the trouble of looking for appropriate training sets for different kinds of noisy images. We have shown that such a NS-IT2 FLS is always able to achieve significantly smaller MSEs than the MSEs of all the non-fuzzy techniques.

This chapter also explained how to obtain an augmented version of the NS-IT2 FLS, denoted NS-IT2 FLS+, by appending to it a BM3D DCT filter, which helps clean up residual Gaussian noise. Not only does the NS-IT2 FLS+ always have smaller MSEs than all the non-fuzzy techniques and just the NS-IT2 FLS, but its images also look better than the images of all these techniques.

Finally, the chapter proposed the cascading of a standard linear contrast scaling filter with the NS-IT2 FLS+ to bring back the reduced contrast level of the de-noised images. Such a fully augmented system is denoted NS-IT2 FLS++, and has been shown to produce images with unparalleled visual quality. Although the NS-IT2 FLS++ images can sometimes have slightly larger MSEs than the NS-IT2 FLS and the NS-IT2 FLS images, these cases are very rare and the MSE differences are generally negligible.

In summary, the proposed NS-IT2 FLS++ allows us to achieve excellent MSE results as well as exceptionally good visual appearances.

Chapter 4 A Mixture Fuzzy Logic System for Forest-Fire-Size Prediction

4.1 Introduction

In the second application that focuses on the system design perspective — forest-fire-size prediction—, the data sets contain a small amount of unique and extreme events, which contain important information and, therefore, cannot simply be ignored. But mixing the extreme events with other regular events can cause trouble (overfitting) in the training stage of the system. This Chapter proposes a Mixture Fuzzy Logic System that consists of two components: 1) a fixed number of general rules modeled by Interval Type 2 Fuzzy Sets tuned by the regular events; and 2) a number of elite rules, each of which is modeled by Type-1 Fuzzy Sets and specifically designed using an unique and extreme event. We show that such a system, when applied to a benchmark forest fire data set, enables us to obtain better prediction of regular events and to also extract knowledge about the unique and extreme events..

4.2 Forest Fire Data Set

The benchmark data set (see Machine Learning Repository of University of California, Irvine: http://archive.ics.uci.edu/ml/datasets/Forest+Fires) was collected from the Montesinho natural park located in the northeast part of Portugal, which documents the *burnt area* and the corresponding spatial, temporal and meteorological measurements, each time a forest fire occurred. A total number of 517 fire incidents have been recorded. The elements from this data set are listed in Table 4.1. Note that this list does not include temporal data, because they are recorded in terms of month and day, and, therefore, are not suitable for some of the numerical processes introduced in this chapter.

After careful examinations of the data set, it is noticeable that, out of the 517 fire recordings, the *burnt area* has 247 zero entries, which means that these fires had sizes smaller than 0.01 hectare (ha), where one hectare is 10,000 square meters or 2.471 acres. As Fowler et al. [58] pointed out, "the fact that such a large proportion of the data indicates a burn area of size 0 affects a number of decisions in the development of the prediction system". As an effort to decide which elements in Table 4.1 should be used for optimal system design, Cortez and Morais tested (not exhaustively) several different subsets of these elements as inputs to their data-mining systems; but the element sets they tested seem to have little impact on the prediction results. In [58], Fowler et al. proposed to use four of the meteorological elements, namely, *temperature*, *relative humidity*, *wind speed* and *rain*, as inputs to their FLS. Although, intuitively, these elements could have direct impact on the fires, such a choice remains subjective and could be problematic, because, e.g., *rain* measurements only have eight nonzero entries, which is hardly enough for training purpose.

4.2.1 Element Selection

In this Chapter, we propose to first categorize the *burnt area* measurements into two classes, zero and nonzero, which allows us to apply *feature selection* techniques [51, 77-79, 95, 98, 142] from *pattern classification* to determine a subset of elements in Table 4.1 that "optimally" characterize the two-class *burnt area* measurements.

The technique employed in our study is called *minimal-Redundancy-Maximal-Relevance* (mRMR) [142]. It selects the elements that have the highest relevance to the target binary *burnt area*, where "relevance" is measured in terms of mutual information. Because it has long been recognized that "the combination of individually good features do not necessarily lead to good classification performance" [46, 47, 79, 142], the mRMR also ensures the redundant information among the selected elements is minimized.

A ranking of the elements in Table 4.1 (see the second row) was provided by the mRMR (The software that realizes the mRMR method can be found at: http://penglab.janelia.org/proj/mRMR/; its details are not of interest to this chapter.). As can be seen, e.g., *rain* is ranked as the least useful element, which contradicts our intuition. This could simply be due to the fact that the it hasn't rained too often in the park during those two years when the measurements were taken, and, therefore, the ranking could change if more fire incidents occurred during rain time and get recorded.

Based on mRMR, we select the first four elements, namely, *Duff Moisture Code* (DMC), *Relative Humidity* (RH), *Drought Code* (DC) and *Temperature* (Temp.), as data sources for our system. Note that this unique combination of elements was not tested by Cortez and Morais.

4.2.2 Unique and Extreme Events

Another important observation about the data set, which has significant impact on the system performance, is: out of the 517 *burnt area* measurements, 515 of them have values distributed over the interval [0, 278.5] (ha). We refer to a measurement like this and the

Table 4.1 Data elements and their ranking based on minimal-Redundancy-Maximum-Relevance (mRMR)

Burn Area	Duff Moist. Code	Relative Humidity	Drought Code	Temp.	Fine Fuel Moist. Code	Initial Spread Index	Wind Speed	x	y	Rain
	1	2	3	4	5	6	7	8	9	10

element values associated with it as a *regular event*. The other two measurements have relatively very large values, 746.3 and 1090.8 ha. We refer to a measurement like this and the element values associated with it as an *extreme event*. Note also that, these two extreme events are themselves quite different (1090.8-746.3=334.5 ha), making each of them also an unique event in the data set.

All previous forest-fire-size prediction techniques [41, 58] employed 30 Monte-Carlo simulations each with a ten-fold cross-validation, in which the objective function, $f_{obj}(\cdot)$, for the system was either the following RMSE or MAD measures:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{517}[y(k) - \hat{y}(k)]^2} \tag{4-1}$$

$$MAD = \frac{1}{N}\sum_{k=1}^{517}\left|y(k) - \hat{y}(k)\right| \tag{4-2}$$

where $y(k)$ and $\hat{y}(k)$ are, respectively, the actual and estimated values of the k^{th} *burnt area* measurement ($k \in \{1,2,\ldots,517\}$).

During each cross-validation process, all techniques could quickly become overfitted if the training data set contained one or both of the extreme events. Because each technique is finely tuned to reduce the large errors caused by the extreme events during training stage, its performance for predicting the remaining regular events became poor during testing stage.

No system, however, can afford to rule out these two extreme events from the data set; because, to do so would cause the system to completely lose its ability to predict an exceptionally destructive forest fire, which, as Fowler et al. [58] point out, "might have far more serious consequences than many small errors".

Therefore, we propose to distinguish between regular and extreme events in the training data set, and use them to design an IT2 FLS and a T1 FLS, respectively. Such an IT2 FLS provides us a set of *general rules* that are free from overfitting caused by extreme events and specialize in predicting regular events; and, on the other hand, the T1 FLS contains a set of *elite rules* that enable us to predict extreme events.

4.3 Mixture Fuzzy Logic System

As mentioned in Section 4.2, the regular events in the training data set are used to tune the above IT2 FLS, which provides us with a set of *general rules* that specializes in predicting regular-size fires. Meanwhile, the extreme events are used to design a T1 FLS that provides us with a set of *elite rules* for predicting significantly large fires.

4.3.1 Elite Rules for Predicting Extreme Events

Due to uniqueness of the extreme events, each *elite rule* is especially designed based on one such event and, e.g., has the following form:

Elite Rule 1: **IF** x_1 is F_1^1 and x_2 is F_2^1 ... and x_7 is F_7^1 and x_8 is F_8^1; **THEN** y is G^1.

where x_5, x_6, x_7, x_8 denote the following four remaining data elements in Table 4.1: *Fine Fuel Moisture Code*, *Initial Spread Index*, *Wind Speed* and *rain*, respectively. This because we discovered that a rarely large fire (extreme event) occurs only when all different conditions (except location) are satisfied to certain degree. Thus, all eight data elements need to be considered as antecedents when designing rules associated with extreme events.

Because there is only one extreme event available for each rule, we chose to model F_1^1, F_2^1,..., F_8^1 and G^1 as T1 FSs instead of IT2 FSs such that each *elite rule* has much fewer design parameters than does *general rule*. So that, we do not need to employ the tuning mechanism used by an IT2 FLS. Instead, one of the earliest rule-constructing principle is employed [122, pp. 112]: "*Let the data establish the center of the fuzzy sets that appear in the antecedent and consequent of the rules.*" More specifically, the MF for F_j^1 ($j = 1,2,\ldots,8$) is proposed to be:

$$\mu_j^1(x_j) = N(x_j', 0.1\times x_j'; x_j) \tag{4-3}$$

Note that y' and x_j' ($j = 1,2,\ldots,8$) are the actual values of *burnt area* and corresponding elements of an extreme event. Examples of two MFs of these eight antecedent T1 FSs,

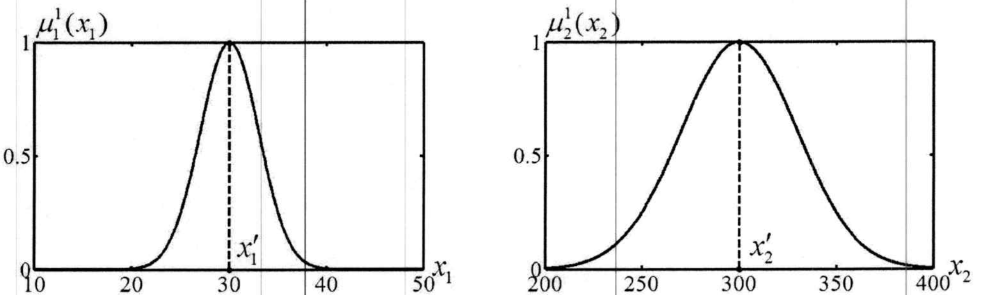
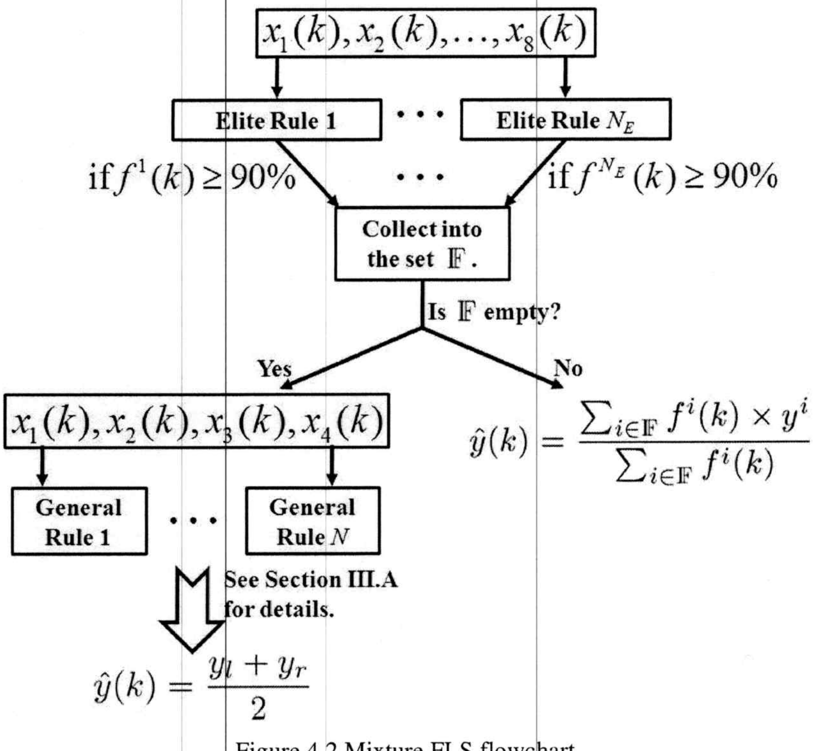


Figure 4.1 Examples of two MFs for the eight antecedent T1 FSs: $\mu_1^1(x_1)$ and $\mu_2^1(x_2)$ are centered at the actual values, x_1' and x_2', that are associated with an extreme event.

$\mu_1^1(x_1)$ and $\mu_2^1(x_2)$, are depicted in Fig. 4.1. As for the consequent of the rule, G^1 is set to be a T1 FS whose centroid equals y'.

4.3.2 Firing Criteria

A Mixture FLS is established by putting the two sets of rules (FLSs) together in parallel. However, unlike a single T1 or IT2 FLS where all rules are fired together to different levels, a slightly modified criterion becomes necessary to determine which set of rules is to be fired, because one needs to be more cautious about firing the *elite rules* due to the significant consequences they can bring about.



$x_1(k), x_2(k), \ldots, x_8(k)$
Elite Rule 1
Elite Rule N_E
if $f^1(k) \geq 90\%$
if $f^{N_E}(k) \geq 90\%$
Collect into the set $\mathbb{F}$.
Is $\mathbb{F}$ empty?
Yes
No
$x_1(k), x_2(k), x_3(k), x_4(k)$
$\hat{y}(k) = \frac{\sum_{i \in \mathbb{F}} f^i(k) \times y^i}{\sum_{i \in \mathbb{F}} f^i(k)}$
General Rule 1
General Rule N
See Section III.A for details.
$\hat{y}(k) = \frac{y_l + y_r}{2}$

Figure 4.2 Mixture FLS flowchart

Given a set of actual input values, $x_1(k)$, $x_2(k)$,..., $x_8(k)$ ($k \in \{1,2,\dots,517\}$), we first use them to compute the *firing level* of each *elite rule* , i.e. ($i = 1,2,\dots,N_E\}$):

$$f^i(k) = \min_{j\in\{1,2,\dots,8\}} \mu_j^i(x_j(k)) \tag{4-4}$$

where N_E is the total number of unique and extreme events (elite rules). If none of the firing levels are greater than or equal to 90%, i.e. $f^i(k) < 0.9$ ($\forall i \in [1, N_E]$), no *elite rule* is fired; and, $x_1(k)$, $x_2(k)$, $x_3(k)$ and $x_4(k)$ are fed to the *general rules* to obtain a prediction of the fire-size. The *firing threshold*, 90%, ensures that every condition necessary for a rarely large fire exists at least to the degree of 90%.

If one or more *elite rules* pass the firing threshold, the set of *general rules* is ignored, i.e., only the passing *elite rules* are fired. We denote the set of such rules as Z. They can be considered as a T1 FLS and defuzzified as [122]:

$$\hat{y}(k) = \frac{\sum_{i\in\square} f^i(k)\times y^i}{\sum_{i\in\square} f^i(k)} \tag{4-5}$$

where y^i is the *burnt area* value of the extreme event, based on which the i^{th} *elite rule* in Z is constructed. A flowchart of above procedure of the Mixture FLS is depicted in Fig. 4.2.

Table 4.2 30-time Monte-Carlo simulations of ten-fold cross-validation

Objective Function	Naïve Average Predictor	Naïve Zero Predictor	Multiple Regression	Decision Trees	Random Forests
RMSE	63.59	64.89	64.5	64.5	64.4
MAD	18.57	12.85	13.01	13.18	12.93
Objective Function	Neural Network	Support Vector Machine	Evolved T1 FLS	Mixture FLS (T1)	Mixture FLS (IT2)
RMSE	66.9	64.7	50.3	46.18	**45.7**
MAD	13.71	13.71	15.13	13.50	**12.32**

4.4 Experimental Results

The average RMSEs and MADs of the Mixture FLS and other techniques mentioned in Section 1.4, based on 30 Monte-Carlo simulations each with a ten-fold cross-validation, are summarized in Table 4.2. Note that we have also included in Table 4.2 the results of a Mixture FLS whose general rules are modeled by T1 FSs instead of IT2 FSs. Observe that *the Mixture FLS with IT2 FSs outperform all the other techniques under both measures.*

Comment: Prior to the introduction of the Mixture and Evolved FLSs, Cortez and Morais' SVM approach [41] was the winner of the test by a very narrow margin over the runner-up, the Naïve Zero Predictor, in terms of MAD measure, which is less sensitive to large errors; but, no technique could outperform the simple Naïve Average Predictor in terms of RMSE measure, which is partially the reason why Cortez and Morais wanted to also use MAD for evaluation.■

Among the 300 validation tests, there were cases where both extreme events were assigned to the training data set. We observed that, in such cases, the *elite rules* were never fired and the *general rules* worked very well on the testing data set which consisted of 100% regular events: the RMSEs and MADs of such cases actually got as low as 7.66 and 3.25, respectively.

The reason that the average RMSE of 45.7 and MAD of 12.32 are much higher than 7.66 and 3.25 is that there were also cases where both extreme events were assigned to the testing stage and no *elite rules* could be developed. In such cases, all techniques produced much worse results than their averages, because, as one can easily imagine, no technique could reasonably predict an extreme event if no prior information regarding it is provided.

This problem is caused by the nature of the data set, namely, the number of extreme events is too limited. To more thoroughly examine the validity of the *elite rules*, some more extreme events need to be collected and added to the existing data set to ensure that the extreme events will always be present in both training and testing stages.

4.5 Summary

Focusing on a particular kind of data set which has two subsets of events (regular and extreme), this chapter has proposed a Mixture FLS that distinguishes and uses them separately to build an IT2 and a T1 FLS. The IT2 FLS (*general rules*) is tuned by a QPSO algorithm based on measurements from the regular event set, and the T1 FLS (*elite rules*) is designed by a one-pass method based on limited measurements from the extreme event set. Once a set of input values is fed into the system, a firing criterion is first employed to select the set of rules to be used. Those rules are then fired and defuzzified.

When applied to a benchmark forest fire data set, this Mixture FLS outperformed all previous techniques with respect to average values of 300 cross-validation RMSEs and MADs. The Mixture FLS performs very well for predicting regular events; however, due to scarcity of the extreme events, their presence in both training and testing data sets cannot be guaranteed; therefore, the validity of the *elite rules* cannot not be fully determined. As more and more fire incidents become documented, it is expected that the number of extreme events in the data set will also increase to assist validation.

In the future, it is possible to further divide the data set into more than two groups based on numerical analysis such as mRMR, and, thus, build a Mixture FLS with multiple sets of rules. It is also possible that, as new events join the data set, a genuinely different group of events begins to emerge and, thus, demands a new set of rules for itself.

Chapter 5 Uncertainty Measures for General Type-2 Fuzzy Sets

In this and the following chapter, we will be looking at some fundamental research topics for the most advanced FSs that are currently under study — general T2 FSs. In particular, this section focuses on computing different uncertainty measures for such FSs.

Previously, five uncertainty measures have been defined for Interval Type-2 Fuzzy Sets (IT2 FSs): centroid, cardinality, fuzziness, variance and skewness. Based on a recently developed α-plane representation for a general T2 FS, we generalize these definitions to such T2 FSs, and, more importantly, derive a unified strategy for computing all different uncertainty measures with low complexity. Uncertainty measures of T2 FSs, with different shaped *Footprints of Uncertainty* and different kinds of secondary membership functions (MFs), are computed and are given as examples. Observations and summaries are made for these examples, and a *Summary Interval Uncertainty Measure* for a general T2 FS is proposed to simplify the interpretations. Comparative studies of uncertainty measures for Quasi-Type-2 (QT2), IT2 and T2 FSs are also performed to examine the feasibility of approximating T2 FSs using QT2 or even IT2 FSs.

5.1 Introduction

This section is about uncertainty measures for type-2 fuzzy sets (T2 FSs). In order to understand why one should be interested in such measures for T2 FSs, one must first recall some facts about uncertainty, uncertainty measures, and uncertainty measures for both type-1 (T1) and interval type-2 (IT2) FSs.

Zadeh [223] points out that uncertainty is an attribute of information, and introduced the general theory of uncertainty (GTU), because existing approaches to representation of uncertain information are inadequate for dealing with problems in which uncertain information is perception-based and is expressed in a natural language. He also states that, in GTU, uncertainty is linked to information through the concept of granular structure – a concept which plays a key role in human interaction with the real world [74, 218, 224].

It is necessary to quantify the uncertainty associated with fuzzy sets (FS) as they are used as granules in GTU, because, as Klir [92] points out that once uncertainty (and information) measures become well justified, they can very effectively be utilized for managing uncertainty and associated information. For example, they can be utilized for extrapolating evidence, assessing the strength of relationship between given groups of variables, assessing the influences of given input variables on given output variables, measuring the loss of information when a system is simplified, and the like.

Klir [92] and Harmanec [72] have developed three fundamental principles to guide the use of uncertainty measures under different circumstances:

1. The principle of minimum uncertainty, which states that solutions with the least loss of information should be selected, can be used in simplification and conflict resolution problems.
2. The principle of maximum uncertainty, which states that a conclusion should maximize the relevant uncertainty within constraints given by the verified premises, is widely used within classical probability framework [37, 38, 141].
3. The principle *of* uncertainty invariance, which states that the amount of uncertainty should be preserved in each transformation of uncertainty from one

mathematical framework to another, is widely studied in the context of probability-possibility transformation [62, 91, 93, 185].

Cross and Sudkamp [49] indicate that the quantification of the degree of uncertainty in a FS depends upon the type of uncertainty one is trying to measure and on the particular measure selected for that type of uncertainty.

Among many uncertainty measures proposed for T1 FSs, the most frequently-used are centroid, cardinality, fuzziness (entropy), variance and skewness, which provide us with very useful characteristics of such FSs. For example, as Karnik and Mendel [85] pointed out, the centroid of a T1 FS can be viewed as analogous (not equal!) to the mean of a probability density function. It is the most fundamental uncertainty measure for T1 FSs, and is used in the definitions of several other uncertainty measures, e.g., variance and skewness.

Regarding cardinality of a T1 FS, Dubois and Prade [52] state that cardinality is a natural tool for capturing the meaning of linguistic quantifiers [200, 213, 214, 217, 219-221] and provide satisfactory answers to queries pertaining to quantification, of the form "How many X's are A", "Are there more X's which are A than X's which are B", etc. Wygralak [194] pointed out that these queries occur in computing with words, communication with data bases and information/intelligent systems, modeling the meaning of imprecise quantifiers in natural language statements, decision-making in a fuzzy environment, analysis of grey tone images, clustering, etc.

Wenstop [183] showed how to use the centroid and the cardinality of T1 FSs to measure the distance between two T1 FSs, which allows one to find the FS that most resembles a target T1 FS A among a group of T1 FSs B_i ($i = 1,\ldots, N$). Bonissone [17, 18] proposed a two-step approach to solve the same problem by simultaneously utilizing centroid, cardinality, fuzziness and skewness. As for other related measures, Bustince et al. [23] have showed how to define and construct Decreasing-Increasing (DI)-subsethood measures for T1 FSs.

For more than a decade, IT2 FSs have been applied in many areas, such as: decision making [139, 161, 188, 198], time series forecasting [10, 122], survey processing [9, 108, 122], document retrieval [19], speech recognition [117, 226], noise cancellation [26, 28, 138], word modeling [108, 126, 189], clustering [152], control [8, 25, 27, 28, 55, 68, 101, 105, 114, 116, 160, 192, 193], wireless communication [104, 162], web-shopping [67], linguistic summarization of database [134, 135], etc. Because of their importance in so many applications, uncertainty [191, 199] and related measures [21, 22, 187] of IT2 FSs have become interests for researchers. Wu and Mendel [191] showed how the Wavy-Slice Representation Theorem (WS RT, also known as *embedded T2 FS representation* or the *Mendel-John Representation Theorem*) of a T2 FS could be used to extend the above T1 uncertainty measures to IT2 FSs. Their work also explained how to compute these IT2 uncertainty measures. They have also shown [128, 186, 187] how the cardinality and centroid of an IT2 FS can be used in Computing With Words (CWW) [215, 216].

Recently, there has been a growing interest in general T2 FSs and fuzzy logic systems (FLSs) [42-45, 65, 66, 86, 106, 118, 122, 125, 169, 170, 174, 180, 181, 222]. Such FSs have more design degrees of freedom than do IT2 FSs, consequently, a general T2 FLS has the potential to out-perform an IT2 FLS. Just as an IT2 FLS involves type-reduction (TR) so does a general T2 FLS. Centroid TR is one popular form of TR; it requires computing the centroid of a T2 FS, which explains why this uncertainty measure for a general T2 FS has already been the subject of research.

Until very recently, there was no practical way to compute the centroid of a general T2 FS, because this computation requires computing the centroids of all of the embedded T2 FSs of the general T2 FS. Enumerating all such sets is difficult and, depending upon how finely one discretizes the primary and secondary memberships of a T2 FS, there can be an extremely large number of such embedded T2 FSs.

To get around this, Greenfield et al. [66] proposed to compute the centroid of a small number of randomly selected embedded T2 FSs, instead of computing the centroids of all the embedded T2 FSs. Coupland [42] proposed to utilize the x-coordinate of the geometric centroid of the 3D MF of the T2 FS. John and Czarnecki [80] and Lucas el al. [111] proposed to use the centroids of all the vertical slices of the T2 FS.

Even more recently, an α-plane *RT* (also called Horizontal Slice or z-slice *RT*) of a T2 FS was introduced by several research groups independently, i.e., Liu [106], Wagner and Hagras [181], Tahayori et al [174], and Chen and Kawase [32] (the term α-Plane *representation* was first coined by Liu [106]). By applying the α-plane *RT*, Liu [106] showed that the centroid of a T2 FS can be obtained by taking the union of the centroids of all the α-Planes of that T2 FS. Each α-plane centroid can be computed by the Karnik-Mendel (KM) or Enhanced KM (EKM) algorithms [86, 122, 124, 190].

This section generalizes Liu's centroid calculation to other T2 FS uncertainty measures, namely, cardinality, fuzziness, variance and skewness for general T2 FSs. Just as such uncertainty measures are used in T1 and IT2 applications, as explained above, we expect them also to be used in general T2 FS applications.

5.2 Type-2 Fuzzy Logic Set

This section involves many important concepts from T2 FS theory. To help readers quickly familiarize themselves with the notations used in this chapter, the most fundamental symbols and expressions for a T2 FS $\tilde{A}$ are summarized in Table I, and a more detailed review follows next.

5.2.1 Earlier Representations of a general T2 FS

A general T2 FS $\tilde{A}$ can be described by its vertical-slice representation, as:

$$\tilde{A} = \int_{\forall x \in X} \left[\int_{\forall u \in U} f_x(u)/u \right] / x \tag{5-1}$$

In (5-1), $x \in X$ is the primary variable; $u \in U = [0,1]$ is the secondary variable; and, $f_x(u)$ is the secondary MF at x. In Fig. 1(a), examples of secondary *MF*s of $\tilde{A}$ are depicted by the dashed curves for three different primary variable values, x_1, x_2 and x_3.

Definition 1. An embedded T2 FS of $\tilde{A}$, $\tilde{A}_e$, is

$$\tilde{A}_e = \int_{x \in X} [f_x(u)/u]/x, \quad u \in J_x \tag{5-2}$$

where J_x is the interval between the *Lower* and *Upper Membership Functions* (see Definition 8) of the *Footprints of Uncertainty* (see Definition 5) of $\tilde{A}$ at x [see Fig. 1(a)]. $\tilde{A}_e$ is a T2 FS that only has one primary membership, u', at each x', and only has one secondary membership, $f_{x'}(u')$, at that u'.■

Table 5.1 notations, expressions and concepts for a T2 FS $\tilde{A}$

Item	Explanation
$\tilde{A}_\alpha$	α-plane of $\tilde{A}$; $\tilde{A}_\alpha = \int_{\forall x \in X} \int_{\forall u \in U} \{(x,u) \mid f_x(u) \geq \alpha\}$
A_e	Embedded T1 FS of $\tilde{A}$; $A_e = \int_{x \in X} u/x, \ \ u \in J_x$
$\tilde{A}_e$	Embedded T2 FS of $\tilde{A}$; $\tilde{A}_e = \int_{x \in X} [f_x(u)/u]/x, \ \ u \in J_x$
$A_e(\alpha)$	Embedded T1 FS of $\tilde{A}_\alpha$; $A_e(\alpha) = \int_{x \in X} u/x, \ \ u \in [s_L(x \mid \alpha), s_R(x \mid \alpha)]$
α-plane representation of $\tilde{A}$	$\tilde{A} = \bigcup_{\alpha \in [0,1]} \alpha/\tilde{A}_\alpha$
$FOU(\tilde{A})$	Footprint of Uncertainty of $\tilde{A}$; $FOU(\tilde{A}) = \tilde{A}_0$
$f_x(u)$	Secondary membership of $\tilde{A}$
J_x	Interval between the Lower and Upper Membership Functions of $FOU(\tilde{A})$ at x; $J_x = [s_L(x \mid 0), s_R(x \mid 0)]$
$R_{\tilde{A}_\alpha}$	α-level T2 FS of $\tilde{A}$; $R_{\tilde{A}_\alpha} = \alpha/\tilde{A}_\alpha$
$s_L(x \mid \alpha)$ and $s_R(x \mid \alpha)$	Lower and Upper Membership Functions (LMF and UMF) of $\tilde{A}_\alpha$
u (continuous) or u_i (discrete) $\in U$	Secondary variable of $\tilde{A}$
Vertical-Slice representation of $\tilde{A}$	$\tilde{A} = \int_{\forall x \in X} \left[\int_{\forall u \in U} f_x(u)/u \right] / x$
Wavy-Slice representation of $\tilde{A}$	$\tilde{A} = \bigcup_{\forall \tilde{A}_e} \tilde{A}_e$
x (continuous) or x_i (discrete) $\in X$	Primary variable of $\tilde{A}$

Mendel and John [123] have provided the following important *WS RT* for $\tilde{A}$:

$$\tilde{A} = \bigcup_{\forall \tilde{A}_e} \tilde{A}_e \tag{5-3}$$

i.e., a general T2 FS can be represented by the union of all its embedded T2 FSs.

Definition 2. An embedded T1 FS of $\tilde{A}$, A_e, is

$$A_e = \int_{x \in X} u/x, \ \ u \in J_x \blacksquare \tag{5-4}$$

The embedded T1 FS A_e that corresponds to an embedded T2 FS $\tilde{A}_e$ contains the primary memberships of that $\tilde{A}_e$.

Note that when applying the *WS RT* to an IT2 FS, $\tilde{A}$, (5-3) can be simplified to:

$$\tilde{A} = 1 \Big/ \bigcup_{\forall A_e} A_e \tag{5-5}$$

This is because the secondary memberships of $\tilde{A}$ are always 1 for an IT2 FS, so only the primary memberships are needed to specify such a T2 FS.

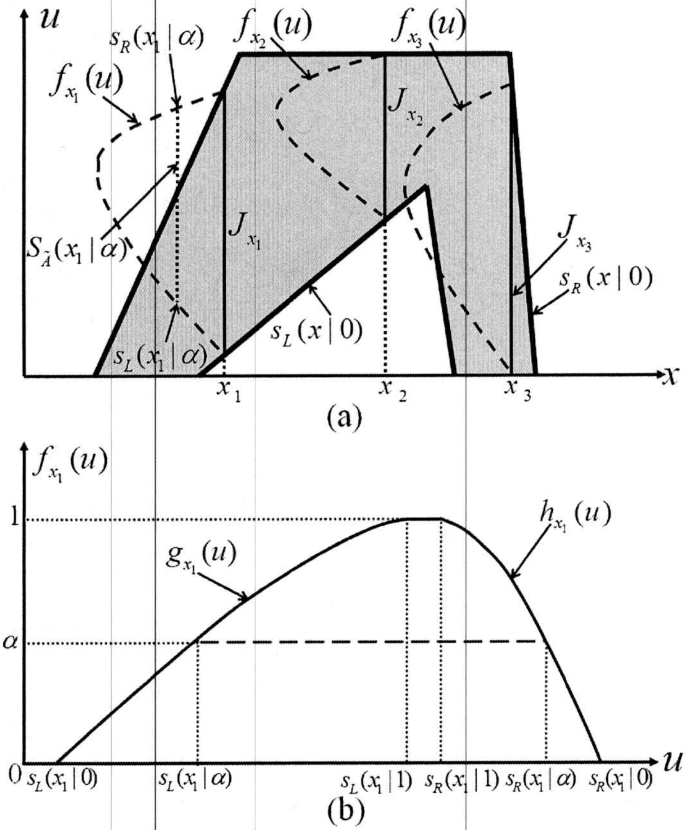


Figure 5.1 (a) *FOU* (shaded area) and secondary *MF*s (dashed curves) of a general T2 FS, (b) Secondary *MF* for the T2 FS from (a) at $x = x_1$

5.2.2 Alpha-Plane RT for a T2 FS

The α-plane of a T2 FS is analogous to an α-cut of a T1 FS.

Definition 3. An α-plane for a general T2 FS $\tilde{A}$, denoted $\tilde{A}_\alpha$, is the union of all primary memberships of $\tilde{A}$ whose secondary grades are greater than or equal to α ($0 \le \alpha \le 1$):

$$\tilde{A}_\alpha = \int_{\forall x \in X} \int_{\forall u \in U} \{(x,u) \mid f_x(u) \ge \alpha\} = \bigcup_{\forall A_e(\alpha)} A_e(\alpha) \tag{5-6}$$

where $A_e(\alpha)$ is an embedded T1 FS of $\tilde{A}_\alpha$.■

Definition 4. $S_{\tilde{A}}(x|\alpha)$ denotes an α-cut of the secondary MF $f_x(u)$, i.e.,

$$S_{\tilde{A}}(x|\alpha) = [s_L(x|\alpha), s_R(x|\alpha)]■ \tag{5-7}$$

An example of $S_{\tilde{A}}(x|\alpha)$ at x_1, raised to level α, is depicted in Fig. 5.1(a).

Definition 5. The 2D domain of $\tilde{A}$, called the *Footprint of Uncertainty* of $\tilde{A}$, is denoted $FOU(\tilde{A})$ and is the $\alpha = 0$ plane, i.e.

$$FOU(\tilde{A}) = \tilde{A}_0■ \tag{5-8}$$

An example of $FOU(\tilde{A})$ is the shaded area in Fig. 1(a).

Definition 6. Let $I_{\tilde{A}_\alpha}(x, u|\alpha)$ be a 3D indicator function for α-plane $\tilde{A}_\alpha$, where:

$$I_{\tilde{A}_\alpha}(x,u \mid \alpha) = \begin{cases} 1 & (x,u) \in \tilde{A}_\alpha \\ 0 & (x,u) \notin \tilde{A}_\alpha \end{cases} = \begin{cases} 1, \forall x \in X \text{ and } \forall u \in [s_L(x \mid \alpha), s_R(x \mid \alpha)] \\ 0, \qquad \text{otherwise} \end{cases} \quad ■(5\text{-}9)$$

Definition 7. An α-level T2 FS, $R_{\tilde{A}_\alpha}$, is [120]:

$$R_{\tilde{A}_\alpha} = \alpha I_{\tilde{A}_\alpha}(x, u|\alpha) = \alpha/\tilde{A}_\alpha \blacksquare \tag{5-10}$$

Definition 8. $\tilde{A}_\alpha$ is bounded from both above and below. The upper bound is called *Upper Membership Function* (*UMF*) for $\tilde{A}_\alpha$, and the lower bound is called *Lower Membership Function* (*LMF*) for $\tilde{A}_\alpha$.■

There are several different notations for *UMF* and *LMF* in the literature. Observe from (9) that $s_L(x|\alpha)$ and $s_R(x|\alpha)$ $\forall x \in X$ denote the *LMF* and *UMF* for $\tilde{A}_\alpha$, respectively, and are used by us in the rest of this dissertation, where $s_L(x|\alpha) = inf\{u|u \in [0,1], f_x(u) \geq \alpha\}$ and $s_R(x|\alpha) = sup\{u|u \in [0,1], f_x(u) \geq \alpha\}$. Consequently, $s_L(x|0)$ and $s_R(x|0)$ $\forall x \in X$ denote the *LMF* and *UMF* for $FOU(\tilde{A})$ [e.g., see Fig. 1(a)], and $J_x = [s_L(x|0), s_R(x|0)]$.

Fig. 5.1(b) depicts the secondary *MF* for the T2 FS $\tilde{A}$ given in Fig. 5.1(a) at $x = x_1$. In this figure, the *LMF* and *UMF* values for $FOU(\tilde{A})$ [$s_L(x_1|0)$ and $s_R(x_1|0)$], the *LMF* and *UMF* values for $\tilde{A}_\alpha$ at a given α [$s_L(x_1|\alpha)$ and $s_R(x_1|\alpha)$], and the *LMF* and *UMF* values for $\tilde{A}_1$ [$s_L(x_1|1)$ and $s_R(x_1|1)$], are labeled respectively.

Liu's [106] α-plane RT can be expressed in different ways, as:

$$\tilde{A} = \bigcup_{\alpha\in[0,1]} R_{\tilde{A}_\alpha} = \bigcup_{\alpha\in[0,1]} \alpha/\tilde{A}_\alpha = \bigcup_{\alpha\in[0,1]} \left\{ \alpha \Big/ \bigcup_{\forall A_e(\alpha)} A_e(\alpha) \right\} \tag{5-11}$$

This RT is very useful because, since each $\tilde{A}_\alpha$ can be viewed as the *FOU* of an α-level T2 FS — a special IT2 FS —, operations involving T2 FSs can be performed by using existing techniques that have already been developed for IT2 FSs.

5.2.3 Assumptions on the Secondary MFs

In this dissertation, it is assumed that:

$$\tilde{A}_{\alpha_1} \subseteq \tilde{A}_{\alpha_2},\ \forall \alpha_1, \alpha_2 \in [0,1] \text{ and } \alpha_1 > \alpha_2 \qquad (5\text{-}12)$$

Equation (5-12) can be interpreted in terms of secondary *MF*s, i.e. each secondary *MF* of $\tilde{A}$ must be a convex T1 FS [94], i.e. it must satisfy the following structure:

$$f_x(u) = \begin{cases} g_x(u) & u \in [s_L(x \mid 0), s_L(x \mid 1)) \\ 1 & u \in [s_L(x \mid 1), s_R(x \mid 1)] \\ h_x(u) & u \in (s_R(x \mid 1), s_R(x \mid 0)] \\ 0 \quad u \in (-\infty, s_L(x \mid 0)) \cup (s_R(x \mid 0), +\infty) & \end{cases} \qquad (5\text{-}13)$$

In (5-13), $g_x(u)$ and $h_x(u) \in [0,1]$ are monotonically non-decreasing and non-increasing, respectively. The secondary MF depicted in Fig. 5.1(b) is an example of $f_x(u)$ that satisfies 5-(13). Practical examples of secondary MFs that satisfy (5-13) are triangles and trapezoids.

Table 5.2 Definitions of five T1 uncertainty measures

Name	Formula	References
Centroid	$c(A) = \frac{\sum_{i=1}^{N} x_i\, u_i}{\sum_{i=1}^{N} u_i}$	[85, 122]
Cardinality[a]	$p(A) = \frac{\lvert X \rvert}{N} \sum_{i=1}^{N} u_i$	[112]
Fuzziness (Yager's)[b]	$f_Y(A) = 1 - \frac{\left[\sum_{i=1}^{N} \lvert 2u_i - 1 \rvert^r\right]^{1/r}}{N^{1/r}}$	[197]
Variance	$v(A) = \frac{\sum_{i=1}^{N} [x_i - c(A)]^2\, u_i}{\sum_{i=1}^{N} u_i}$	[100]
Skewness	$s(A) = \frac{\sum_{i=1}^{N} [x_i - c(A)]^3\, u_i}{\sum_{i=1}^{N} u_i}$	[191]

a. $|X| = x_N - x_1$ is the length of the universe of discourse used in the computation.

b. r is a positive constant.

Table 5.3 Definitions of the five T2 uncertainty measures in (5-14) and (5-15)

Name & Symbol	$\xi =$
Centroid, $C_{\tilde{A}}(\xi)$	$\frac{\sum_{i=1}^{N} x_i\, u_i(A_e)}{\sum_{i=1}^{N} u_i(A_e)}$
Cardinality, $P_{\tilde{A}}(\xi)$	$\frac{\lvert X\rvert}{N}\sum_{i=1}^{N} u_i(A_e)$
Fuzziness (Yager's), $F_{\tilde{A}}(\xi)$	$1-\frac{\left[\sum_{i=1}^{N}\lvert 2u_i(A_e)-1\rvert^r\right]^{1/r}}{N^{1/r}}$
Variance, $V_{\tilde{A}}(\xi)$	$\frac{\sum_{i=1}^{N}[x_i-c(A_e)]^2\, u_i(A_e)}{\sum_{i=1}^{N} u_i(A_e)}$
Skewness, $S_{\tilde{A}}(\xi)$	$\frac{\sum_{i=1}^{N}[x_i-c(A_e)]^3\, u_i(A_e)}{\sum_{i=1}^{N} u_i(A_e)}$

5.3 Uncertainty Measures for General T2 FSs

In this section, we define the uncertainty measures for a general T2 FS, show how they can be computed by taking the union of the uncertainty measures of all the α-level T2 FSs, and review existing algorithms for computing uncertainty measures for an IT2 FS.

5.3.1 Definitions of Uncertainty Measures for a General T2 FS

The uncertainty measure of a general T2 FS is a T1 FS, $M_{\tilde{A}}$, whose *MF* is $M_{\tilde{A}}(\xi), \forall \xi \in X$. The following general two-step procedure is proposed for defining $M_{\tilde{A}}(\xi)$, because it was already used by Karnik and Mendel [85, 122] to define the centroid of a general T2 FS:

1. Choose a well-established uncertainty measure for a T1 FS *A*, *m*(*A*). The five T1 uncertainty measures (centroid, cardinality, fuzziness, variance and skewness) used in this chapter follow the same definitions in [191], and are summarized in Table II (Because the actual computations of the uncertainty measures require

discrete universes of discourse, all of the definitions in this chapter assume such discretizations.).

2. Use the *WS RT* to define $M_{\tilde{A}}$, with *MF* $M_{\tilde{A}}(\xi)$ $(\forall \xi \in X)$ as

$$M_{\tilde{A}}(\xi) \equiv \bigcup_{\forall \tilde{A}_e} m(\tilde{A}_e) \tag{5-14}$$

where the uncertainty measure of $\tilde{A}_e$, $m(\tilde{A}_e)$, is defined as

$$m(\tilde{A}_e) \equiv \min_{(x,u)\in A_e} \{f_x(u)\} \big/ [\xi = m(A_e)] \tag{5-15}$$

in which A_e is the corresponding embedded T1 FS of $\tilde{A}_e$. The five corresponding T2 uncertainty measures are summarized in Table III.

To describe the T1 FS $M_{\tilde{A}}$, we use its centroid, $c(M_{\tilde{A}}(\xi))$, and standard deviation, $SD(M_{\tilde{A}}(\xi))$, where [(5-16) and (5-17) are for discrete ξ. For continuous ξ, replace the summations in these equations by integrals]

$$c(M_{\tilde{A}}(\xi)) = \frac{\sum_{i=1}^{N} \xi_i M_{\tilde{A}}(\xi_i)}{\sum_{i=1}^{N} M_{\tilde{A}}(\xi_i)}; \tag{5-16}$$

$$SD(M_{\tilde{A}}(\xi)) = \sqrt{\frac{\sum_{i=1}^{N} (\xi_i - c(M_{\tilde{A}}(\xi)))^2 M_{\tilde{A}}(\xi_i)}{\sum_{i=1}^{N} M_{\tilde{A}}(\xi_i)}} \tag{5-17}$$

5.3.2 Computing Uncertainty Measures for a General T2 FS

Direct computations of the T2 uncertainty measures are not practical because for even reasonable accuracy they require enumerations of an extremely large number of embedded T2 FSs. Using the α-plane RT, Liu [106] has already shown that $C_{\tilde{A}}(\xi)$ $(\forall \xi \in X)$ can be computed by taking the union of the centroids of the α-planes of $\tilde{A}$, $C_{\tilde{A}_\alpha}(\xi)$, i.e.

$$C_{\tilde{A}}(\xi) = \bigcup_{\alpha \in [0,1]} \alpha / C_{\tilde{A}_\alpha}(\xi) \tag{5-18}$$

A generalization of (5-18) to other uncertainty measures is given in the following:

Theorem 1. Let $m(A)$ be a given uncertainty measure for a T1 FS A, $M_{\tilde{A}}(\xi)$ $(\forall \xi \in X)$ be the corresponding uncertainty measure for T2 FS $\tilde{A}$ [see (5-14)] and $M_{\tilde{A}_\alpha}(\xi)$ $(\forall \xi \in X)$ be the corresponding uncertainty measure for $\tilde{A}_\alpha$, where

$$M_{\tilde{A}_\alpha}(\xi) \equiv \bigcup_{\forall A_e(\alpha)} [\xi = m(A_e(\alpha))] = [m_l(\tilde{A}_\alpha), m_r(\tilde{A}_\alpha)] \tag{5-19}$$

and $m_l(\tilde{A}_\alpha)$ and $m_r(\tilde{A}_\alpha)$ are the smallest and largest values of $m(A_e(\alpha))$, respectively. Then,

$$M_{\tilde{A}}(\xi) = \bigcup_{\alpha \in [0,1]} \alpha / M_{\tilde{A}_\alpha}(\xi) = \bigcup_{\alpha \in [0,1]} \alpha / [m_l(\tilde{A}_\alpha), m_r(\tilde{A}_\alpha)] \; \blacksquare \tag{5-20}$$

Proof:

The theorem and its proof are given for continuous universes of discourse, but they can easily be specialized to discrete universes of discourse. By combining (5-14) and (5-15), it follows that:

$$M_{\tilde{A}}(\xi) = \bigcup_{\forall \tilde{A}_e} m(\tilde{A}_e) = \bigcup_{\forall \tilde{A}_e} \min_{(x,u)\in A_e} \{f_x(u)\} \Big/ m(A_e) \tag{5-21}$$

Because $M_{\tilde{A}}(\xi)$ is a T1 FS, it can be described by its T1 α-cut representation [94], as:

$$M_{\tilde{A}}(\xi) = \bigcup_{\alpha\in[0,1]} \alpha / M_{\tilde{A}}(\xi \mid \alpha),\ \forall \xi \in X \tag{5-22}$$

where $M_{\tilde{A}}(\xi|\alpha)$ is the α-cut of $M_{\tilde{A}}(\xi)$, i.e. $(\forall \xi \in X)$

$$M_{\tilde{A}}(\xi \mid \alpha) = \left\{ \xi \mid M_{\tilde{A}}(\xi) \ge \alpha \right\} \tag{5-23}$$

Substituting (5-21) into (5-23), it follows that:

$$M_{\tilde{A}}(\xi \mid \alpha) = \bigcup_{\forall \tilde{A}_e} \left\{ m(A_e) \mid \min_{(x,u)\in A_e} \{f_x(u)\} \ge \alpha \right\} \tag{5-24}$$

Note that

$$\min_{(x,u)\in A_e} \{f_x(u)\} \ge \alpha \tag{5-25}$$

means that

$$f_x(u) \ge \alpha \quad \forall (x,u) \in A_e\,; \tag{5-26}$$

hence, the embedded T1 FS A_e must belong to $\tilde{A}_\alpha$. Consequently, it follows that:

$$M_{\tilde{A}}(\xi \mid \alpha) = \bigcup_{\forall \tilde{A}_e} \left\{ m(A_e) \mid A_e \in \tilde{A}_\alpha \right\} = \bigcup_{\forall A_e \in \tilde{A}_\alpha} m(A_e) = \bigcup_{\forall A_e(\alpha)} m(A_e(\alpha)) \tag{5-27}$$

Note that the last line of (5-27) is the definition of the uncertainty measure for $\tilde{A}_\alpha$ which is given in (5-19), i.e.

$$M_{\tilde{A}}(\xi \mid \alpha) = M_{\tilde{A}_\alpha}(\xi) \quad \forall \xi \in X \tag{5-28}$$

Substituting (5-28) into (5-22), it follows that

$$M_{\tilde{A}}(\xi) = \bigcup_{\alpha \in [0,1]} \alpha \big/ M_{\tilde{A}_\alpha}(\xi), \ \forall \xi \in X \tag{5-29}$$

which is the middle part of (5-20).

Next, consider the α-level T2 FS $\tilde{A}_\alpha$, whose uncertainty measure, $M_{\tilde{A}_\alpha}(\xi)$ ($\forall \xi \in X$), is defined by (5-28) and the last line of (5-27). Observe that (5-27) directly uses the uncertainty measure of the embedded T1 FS of $\tilde{A}_\alpha$, $m(A_e(\alpha))$. Since each $m(A_e(\alpha))$ is a positive real number, the union of all the numbers has a smallest and a largest value, $m_l(\tilde{A}_\alpha)$ and $m_r(\tilde{A}_\alpha)$, respectively; hence

$$M_{\tilde{A}_\alpha}(\xi) = [m_l(\tilde{A}_\alpha), m_r(\tilde{A}_\alpha)] \tag{5-30}$$

Substituting (5-30) into (5-29) gives:

$$M_{\tilde{A}}(\xi) = \bigcup_{\alpha \in [0,1]} \alpha \big/ [m_l(\tilde{A}_\alpha), m_r(\tilde{A}_\alpha)], \ \forall \xi \in X \tag{5-31}$$

which is the right-hand side of (5-20).■

Equation (5-20) lets us compute our previously defined T2 FS uncertainty measures with reasonable computational costs, because each $M_{\tilde{A}_\alpha}(\xi)$ can be computed using already-established algorithms for IT2 FSs and all the $M_{\tilde{A}_\alpha}(\xi)$ can be computed in parallel. If, for example, k values of α are used, then $M_{\tilde{A}_{\alpha 1}}(\xi)$, …, $M_{\tilde{A}_{\alpha k}}(\xi)$ require k parallel processors. Of course, greater accuracy can be obtained by using a larger number of α-planes.

Our general procedure for computing an uncertainty measure for a general T2 FS is:

1. Decide on how many α-planes will be used, where $\alpha \in [0,1]$. Call that number k; its choice will depend on the accuracy that is required. Regardless of k, $\alpha = 0$ and $\alpha = 1$ must always be used.
2. For each α, compute $\tilde{A}_\alpha$.
3. Compute the uncertainty measure of $\tilde{A}_\alpha$, $M_{\tilde{A}_\alpha}(\xi) = [m_l(\tilde{A}_\alpha), m_r(\tilde{A}_\alpha)]$, using already-existing algorithms [191]. The accuracy of this step will depend upon the discretization of the primary variable, x.
4. Repeat Steps 2 and 3 for the k values of α chosen in Step 1.
5. Collect all of the k $M_{\tilde{A}_\alpha}(\xi)$ by using (5-20), to obtain $M_{\tilde{A}}(\xi)$.

It is worth noting the following:

Theorem 2. Each uncertainty measure of a general T2 FS with secondary MFs that are convex T1 FSs, as in (5-13), is also a convex T1 FS whose MF also satisfies (5-13).

Proof:

Given a general T2 FS $\tilde{A}$ whose secondary *MF*s satisfy (5-13), it is clear that

$$s_L(x \mid \alpha_2) > s_L(x \mid \alpha_1) \quad \forall \alpha_2 > \alpha_1 \tag{5-32}$$

$$s_R(x|\alpha_2) < s_R(x|\alpha_1) \quad \forall \alpha_2 > \alpha_1\,; \tag{5-33}$$

hence,

$$\tilde{A}_{\alpha_2} \subset \tilde{A}_{\alpha_1} \tag{5-34}$$

Using (5-19) for $M_{\tilde{A}_{\alpha_1}}(\xi)$, it follows that:

$$\begin{aligned} M_{\tilde{A}_{\alpha_1}}(\xi) &= \bigcup_{\forall A_e(\alpha_1)} m(A_e(\alpha_1)) = \bigcup_{\forall A_e(\alpha_1)\in \tilde{A}_{\alpha_2}} m(A_e(\alpha_1)) + \bigcup_{\forall A_e(\alpha_1)\notin \tilde{A}_{\alpha_2}} m(A_e(\alpha_1)) \\ &= M_{\tilde{A}_{\alpha_2}}(\xi) + \bigcup_{\forall A_e(\alpha_1)\notin \tilde{A}_{\alpha_2}} m(A_e(\alpha_1)) \end{aligned} \tag{5-35}$$

where the second line has made use of (5-34). Consequently, the last line of (5-35) indicates:

$$M_{\tilde{A}_{\alpha_2}}(\xi) \subset M_{\tilde{A}_{\alpha_1}}(\xi) \tag{5-36}$$

Substituting (5-30) into (5-36), we see that:

$$[m_l(\tilde{A}_{\alpha_2}), m_r(\tilde{A}_{\alpha_2})] \subset [m_l(\tilde{A}_{\alpha_1}), m_r(\tilde{A}_{\alpha_1})] \tag{5-37}$$

from which it follows that:

$$m_l(\tilde{A}_{\alpha_2}) > m_l(\tilde{A}_{\alpha_1}) \quad \forall \alpha_2 > \alpha_1 \tag{5-38}$$

$$m_r(\tilde{A}_{\alpha_2}) < m_l(\tilde{A}_{\alpha_1}) \quad \forall \alpha_2 > \alpha_1 \tag{5-39}$$

Comparing (5-38) and (5-39) to (5-12) and (5-33), respectively, shows that $M_{\tilde{A}}(\xi)$ must also satisfy (5-13).■

Definitions of $M_{\tilde{A}_\alpha}(\xi)$ and algorithms for computing its left and right ends, $m_l(\tilde{A}_\alpha)$ and $m_{r'}(\tilde{A}_\alpha)$, are summarized in Table 5.4 for the five uncertainty measures.

5.4 Examples

In this section, uncertainty measures for different T2 FSs are computed using the procedures described above. The specific T2 FSs were selected in the following way:

1. Three representative *FOUs* were selected from the 32-word codebook in [109] each of which formed $FOU(\tilde{A})$. The three words used are *Tiny*, *Medium* and *Very High Amount*. Their *FOU*s are depicted in Fig. 5.2 and the parameters of their *LMF*s and *UMF*s are given in Table V. These five *FOU*s were chosen because they have quite different shapes, successively cover different spans of the primary variable's domain, [0, 10]. Each word is a separate example.
2. For each word, two classes of secondary *MF*s were used, namely triangles and trapezoids. The triangle secondary *MF*s have base equal to $s_R(x|0) - s_L(x|0)$ and apex location, *Apex*(*x*), parameterized as [125] (*w* = 0, 0.5, 1) :

$$Apex(x) = s_L(x\,|\,0) + w\left[s_R(x\,|\,0) - s_L(x\,|\,0)\right] \tag{5-40}$$

An example of the triangle secondary *MF*s for the word *Medium* is shown in Fig. 5.3(b) at *x* = 5.8 for the three values of *w*. The trapezoidal secondary *MF*s have base equal to $s_R(x|0) - s_L(x|0)$ and top defined by the locations of the left and right end points, $EP_l(x)$ and $EP_{r'}(x)$, both of which are parameterized as [125] (*w* = 0, 0.5, 1) :

$$EP_l(x) = s_L(x \mid 0) + 0.6w\left[s_R(x \mid 0) - s_L(x \mid 0)\right] \tag{5-41}$$

$$EP_r(x) = s_R(x \mid 0) - 0.6(1-w)\left[s_R(x \mid 0) - s_L(x \mid 0)\right] \tag{5-42}$$

An example of the trapezoidal secondary *MFs* for the word *Medium* is shown in Fig. 5.3(c) at $x = 5.8$ for the three values of w.

So, for each of the three words, there are two kinds of secondary *MF*s, each with three cases; hence, our results are for $3 \times 2 \times 3 = 18$ general T2 FSs.

How many discretized values of α are needed to produce a close enough approximation to the centroid has been studied in [106, 125]. Based on those studies, we discretized $\alpha \in [0,1]$ into 21 values, for which $\alpha = 0, 0.05, 0.1, \ldots, 0.95, 1$.

Table 5.4 Algorithms for computing the five uncertainty measures of $\tilde{A}_\alpha$, $M_{\tilde{A}_\alpha}(\xi) = \left[m_l(\tilde{A}_\alpha), m_r(\tilde{A}_\alpha)\right]$

Definitions of $M_{\tilde{A}_\alpha}(\xi)$	algorithms for $m_l(\tilde{A}_\alpha)$	algorithms for $m_r(\tilde{A}_\alpha)$
$C_{\tilde{A}_\alpha}(\xi) = \bigcup_{\forall A_e(\alpha)} \left[\xi = \frac{\sum_{i=1}^{N} x_i u_i(A_e(\alpha))}{\sum_{i=1}^{N} u_i(A_e(\alpha))}\right]$	$c_l(\tilde{A}_\alpha)$, KM/EKM	$c_r(\tilde{A}_\alpha)$, KM/EKM
$P_{\tilde{A}_\alpha}(\xi) = \bigcup_{\forall A_e(\alpha)} \left[\xi = \frac{\lvert X\rvert}{N} \sum_{i=1}^{N} u_i(A_e(\alpha))\right]$	$p_l(\tilde{A}_\alpha) = p(s_L(x \vert \alpha))$[a]	$p_r(\tilde{A}_\alpha) = p(s_R(x \vert \alpha))$
$F_{\tilde{A}_\alpha}(\xi) = \bigcup_{\forall A_e(\alpha)} \left[\xi = 1 - \frac{\left[\sum_{i=1}^{N} \lvert 2u_i(A_e(\alpha)) - 1\rvert^r\right]^{1/r}}{N^{1/r}}\right]$	$f_l(\tilde{A}_\alpha) = f_Y(A_{e1}(\alpha))$[b]	$f_r(\tilde{A}_\alpha) = f_Y(A_{e2}(\alpha))$[c]
$V_{\tilde{A}_\alpha}(\xi) = \bigcup_{\forall A_e(\alpha)} \left[\xi = \frac{\sum_{i=1}^{N} [x_i - c(C_{\tilde{A}}(\xi))]^2 u_i(A_e(\alpha))}{\sum_{i=1}^{N} u_i(A_e(\alpha))}\right]$	$v_l(\tilde{A}_\alpha)$, KM/EKM	$v_r(\tilde{A}_\alpha)$, KM/EKM
$S_{\tilde{A}_\alpha}(\xi) = \bigcup_{\forall A_e(\alpha)} \left[\xi = \frac{\sum_{i=1}^{N} [x_i - c(C_{\tilde{A}}(\xi))]^3 u_i(A_e(\alpha))}{\sum_{i=1}^{N} u_i(A_e(\alpha))}\right]$	$s_l(\tilde{A}_\alpha)$, KM/EKM	$s_r(\tilde{A}_\alpha)$, KM/EKM

a. $p(\bullet)$ is defined in Table 5.2.

b. $f_Y(\bullet)$ is defined in Table 5.2, and, $A_{e1}(\alpha)$ is a T1 FS defined as $(\forall x \in X)$: $u(A_{e1}(\alpha)) = s_R(x|\alpha)$ if $s_R(x|\alpha)$ is further away from 0.5 than $s_L(x|\alpha)$, or $u(A_{e1}(\alpha)) = s_L(x|\alpha)$ otherwise.

c. $A_{e2}(\alpha)$ is a T1 FS defined as $(\forall x \in X)$: $u(A_{e2}(\alpha)) = s_R(x|\alpha)$ if both $s_L(x|\alpha)$ and $s_R(x|\alpha)$ are below 0.5, or $u(A_{e2}(\alpha)) = s_L(x|\alpha)$ if both $s_L(x|\alpha)$ and $s_R(x|\alpha)$ are above 0.5, or $u(A_{e2}(\alpha)) =$ 0.5 otherwise.

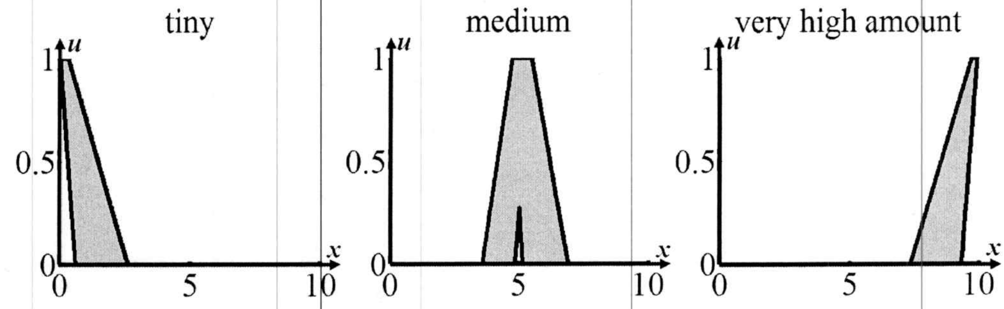


Figure 5.2 Three *FOU*s from a sub-vocabulary selected from the 32-word codebook in [109]

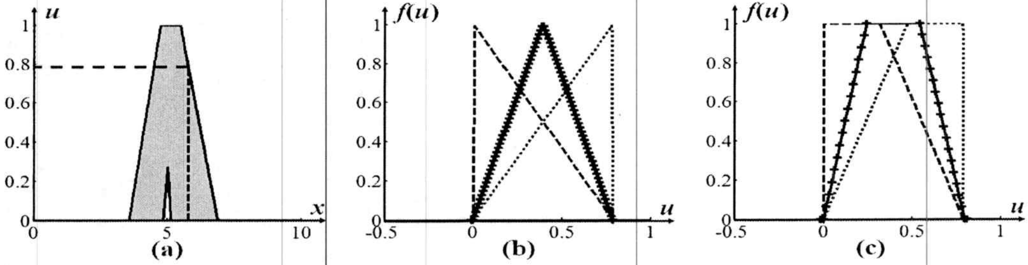


Figure 5.3 (a) *FOU* for the word Medium, (b) five triangular secondary *MF*s parameterized by w [see (5-40), and w = 0, 0.5, 1] when x = 5.8, and (c) five trapezoidal secondary *MF*s parameterized by w [see (5-41) and (5-42)] when x = 5.8.

The uncertainty measures for the three words with triangle and trapezoidal secondary *MF*s are depicted in Figs. 5.4 and 5.5, respectively. In these figures, each row displays the five uncertainty measures for the word, i.e. centroid, cardinality, fuzziness, variance and skewness going from left to right. Each plot has three T1 FSs, which correspond to the uncertainty measures of the word for the three triangle (trapezoidal) secondary *MF*s.

Observe from Fig. 5.4, that for all the words, the *MF* of the centroid, $C_{\tilde{A}}(\xi)$, becomes more concentrated as w increases from 0 to 1. "More concentrated" means that $SD(C_{\tilde{A}}(\xi))$ decreases as w increases from 0 to 1. For the somewhat symmetrical *FOU* of *Medium*, $C_{\tilde{A}}(\xi)$ is also somewhat symmetrical for all w values. For the very small sounding word, *Tiny*,

$C_{\tilde{A}}(\xi)$ skews from left to right as w increases from 0 to 1. For the very large sounding word, *Very High Amount*, $C_{\tilde{A}}(\xi)$ skews from right to left as w increases from 0 to 1.

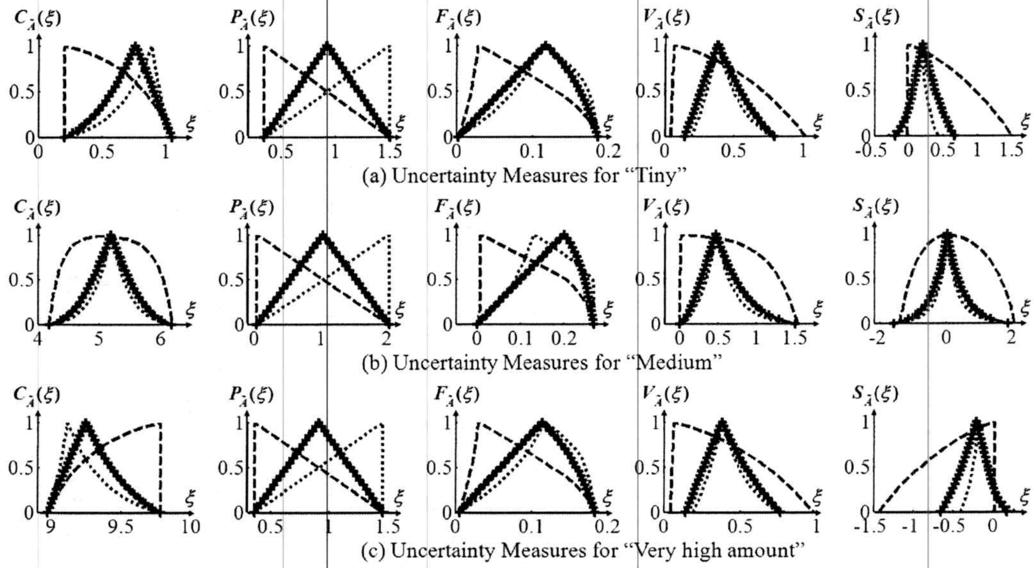
Observe, also, from Fig. 5.4, that for all the words, the *MF* of cardinality, $P_{\tilde{A}}(\xi)$, has exactly the same shape as the secondary *MF*s, e.g. if the secondary *MF*s are triangles parameterized using w = 0.5, $P_{\tilde{A}}(\xi)$ is also the same kind of triangle. This is because the α-cut of $P_{\tilde{A}}(\xi)$, $[p_l(\tilde{A}_\alpha), p_r(\tilde{A}_\alpha)]$, only depends on the *LMF* and *UMF* of $\tilde{A}_\alpha$, $s_L(x|\alpha)$ and $s_R(x|\alpha)$, respectively (see Table IV), and, $s_L(x|\alpha)$ and $s_R(x|\alpha)$ are fully determined by the secondary *MF*s.

Additionally, for all the words, the *MF* of fuzziness, $F_{\tilde{A}}(\xi)$, skews to the left when w = 0, but there is no fixed trend for other values of w.

Finally, for all the words, the MFs of variance and skewness, $V_{\tilde{A}}(\xi)$ and $S_{\tilde{A}}(\xi)$, both become more concentrated as w goes from 0 to 1, just like $C_{\tilde{A}}(\xi)$, because $C_{\tilde{A}_\alpha}$, $V_{\tilde{A}_\alpha}$ and $S_{\tilde{A}_\alpha}$ all have similar mathematical forms (see Table 5.4).

Table 5.5 *FOU* data for three words: each LMF and UMF is represented as a trapezoid (a, b, c, d), where a, b, c and d are the locations of the left end of its base, the left end of its top, the right end of its top and the right end of its base, respectively. The fifth parameter for the LMF is its height.

Word	*LMF*	*UMF*
Tiny	(0, 0, 0.05, 0.63, 1)	(0, 0, 0.36, 2.63)
Medium	(4.86, 5.03, 5.03, 5.14, 0.27)	(3.59, 4.75, 5.50, 6.91)
Very high amount	(9.34, 9.95, 10, 10, 1)	(7.37, 9.73, 10, 10)



(a) Uncertainty Measures for "Tiny"

(b) Uncertainty Measures for "Medium"

(c) Uncertainty Measures for "Very high amount"

Figure 5.4 uncertainty measures for three words and their three triangle secondary *MF*s parameterized by w [see (5-40), and w = 0, 0.5, 1]. Note that the numerical labels for the horizontal axes differ from plot to plot.

Table 5.6 $|SI[M_{\tilde{A}}(\xi)]|$ values for the three words with triangle and trapezoidal secondary *MF*s and for the three values of w, and $|M_{\tilde{A}}(\xi)|$ for the three words with interval secondary *MF*s

		Tiny			***Medium***			***Very High Amount***		
		Secondary MFs			Secondary MFs			Secondary MFs		
	w	Tri.	Trap.	Interval	Tri.	Trap.	Interval	Tri.	Trap.	Interval
Centroid	0	0.538	0.682		1.461	1.792		0.513	0.643	
	.5	0.282	0.430	0.843	0.531	0.823	1.997	0.275	0.421	0.811
	1	0.227	0.343		0.380	0.604		0.223	0.340	
Cardinality	0	0.543	0.735		0.939	1.272		0.514	0.697	
	.5	0.468	0.708	1.153	0.811	1.226	1.994	0.444	0.671	1.093
	1	0.543	0.720		0.939	1.246		0.514	0.683	
Fuzziness	0	0.096	0.133		0.164	0.216		0.091	0.127	
	.5	0.082	0.121	0.184	0.121	0.174	0.267	0.079	0.117	0.178
	1	0.079	0.116		0.136	0.193		0.076	0.111	
Variance	0	0.563	0.636		1.143	1.371		0.536	0.585	
	.5	0.237	0.372	0.742	0.399	0.618	1.509	0.230	0.356	0.712
	1	0.171	0.280		0.284	0.453		0.166	0.270	
Skewness	0	0.866	0.938		2.230	2.936		0.822	0.838	
	.5	0.278	0.451	1.058	0.613	0.981	3.474	0.274	0.430	1.017
	1	0.182	0.311		0.414	0.675		0.180	0.302	

Observations from Fig. 5.5 are very similar to those from Fig. 5.4, and are therefore left to the readers.

It should by now be clear to the reader that it requires some effort to extract useful summarizations and comparisons from the plots of the *MF*s of the uncertainty measures, and, some of them are quite subjective. Next, we turn to more quantitative analyses.

5.5 Quantitative Interpretations and Summaries of Uncertain Measures

The *MF* of a generic uncertainty measure for a general T2 FS, whose secondary MFs satisfy (13), is depicted by the solid curve in Fig. 6, whereas the *MF* for the same generic uncertainty measure for an IT2 FS, that has the same *FOU* as the general T2 FS, is a rectangular function and is depicted by the dashed lines in Fig. 6.

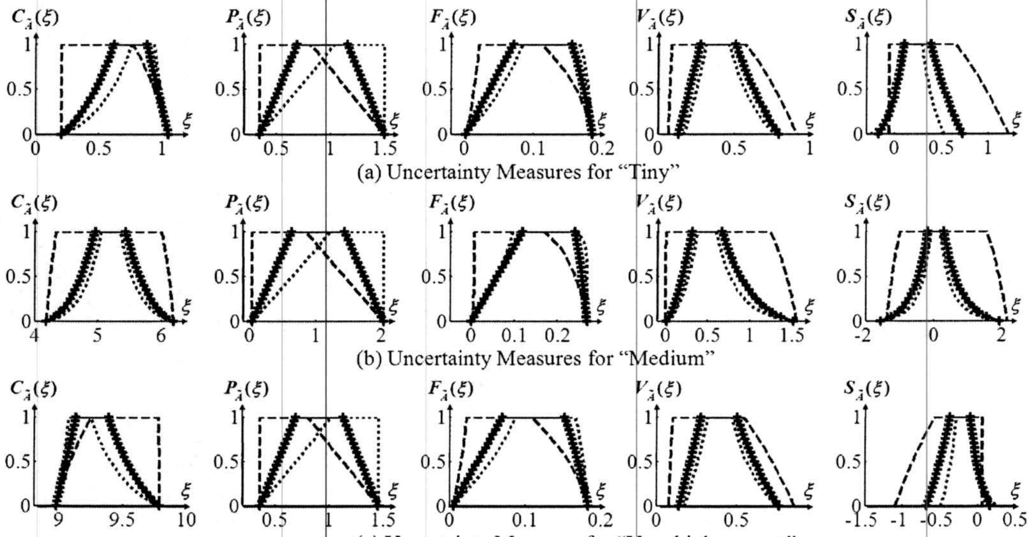
Theorem 3. If the secondary MF of a T2 FS $\tilde{A}$ is contained within the secondary *MF* of another T2 FS $\tilde{B}$ at $\forall x \in X$, then an uncertainty measure for $\tilde{A}$, $M_{\tilde{A}}(\xi)$, is always contained within the uncertainty measure for $\tilde{B}$, $M_{\tilde{B}}(\xi)$.■

If the secondary *MF* of a T2 FS $\tilde{A}$ is contained within the secondary *MF* of another T2 FS $\tilde{B}$ at $\forall x \in X$, then it follows from (5-7) that ($\forall \alpha \in [0,1]$):

Proof:

$$s_{\tilde{A},L}(x \mid \alpha) \geq s_{\tilde{B},L}(x \mid \alpha) \tag{5-43}$$

$$s_{\tilde{A},R}(x \mid \alpha) \leq s_{\tilde{B},R}(x \mid \alpha) \tag{5-44}$$



(a) Uncertainty Measures for "Tiny"

(b) Uncertainty Measures for "Medium"

(c) Uncertainty Measures for "Very high amount"

Figure 5.5 Five uncertainty measures for three words and their three trapezoidal secondary *MF*s parameterized by *w* [see (5-40), and *w* = 0, 0.5, 1]. Note that the numerical labels for the horizontal axes differ from plot to plot.

Table 5.7 Centroids of five uncertainty measures of the T2 FSs, and their Percentage Differences from their IT2 and QT2 counterparts [100(IT2-T2)/T2 and 100(QT2-T2)/T2, respectively], for three words, triangular secondary MFs and three values of *w*

		Tiny			***Medium***			***Very High Amount***		
	w	T2	100(IT2-T2)/T2	100(QT2-T2)/T2	T2	100(IT2-T2)/T2	100(QT2-T2)/T2	T2	100(IT2-T2)/T2	100(QT2-T2)/T2
Centroid	0	0.429	46.20%	13.65%	5.147	0.803%	-0.355%	9.573	-2.019%	-0.620%
	0.5	0.747	-16.12%	-10.18%	5.197	-0.160%	-0.110%	9.264	1.252%	0.784%
	1	0.849	-26.17%	-15.95%	5.200	-0.212%	-0.138%	9.159	2.411%	1.459%
Cardinality	0	0.536	71.855%	36.008%	0.369	180.563%	90.455%	0.541	67.486%	33.820%
	0.5	0.922	-0.06%	-0.083%	1.035	-0.057%	-0.079%	0.907	-0.063%	-0.087%
	1	1.307	-29.5%	-14.82%	1.702	-39.163%	-19.651%	1.272	-28.739%	-14.44%
Fuzziness	0	0.059	60.398%	23.319%	0.068	95.217%	33.821%	0.059	60.395%	24.199%
	0.5	0.114	-16.99%	-10.20%	0.182	-26.712%	-14.730%	0.111	-15.222%	-9.231%
	1	0.121	-21.76%	-16.61%	0.156	-14.419%	-15.591%	0.123	-23.114%	-16.75%
Variance	0	0.284	66.061%	30.739%	0.525	45.424%	0.498%	0.271	68.375%	31.844%
	0.5	0.397	18.962%	11.430%	0.499	52.951%	33.647%	0.393	16.435%	9.897%
	1	0.394	19.986%	13.755%	0.494	54.499%	35.528%	0.394	16.258%	11.569%
Skewness	0	0.312	-116.1%	33.490%	0.197	-496.6%	36.428%	-0.30	56.214%	34.580%
	0.5	0.215	-123.3%	56.293%	0.039	-2091.5%	214.228%	-0.22	111.137%	51.172%
	1	0.157	-132.0%	94.452%	0.031	-2614.8%	285.614%	-0.16	188.114%	87.722%

Table 5.8 Centroids of five uncertainty measures of the T2 FSs, and their Percentage Differences from their IT2 and QT2 counterparts [100(IT2-T2)/T2 and 100(QT2-T2)/T2, respectively], for three words, trapezoidal secondary MFs and three values of *w*

		Tiny			***Medium***			***Very High Amount***		
	w	T2	100(IT2-T2)/T2	100(QT2-T2)/T2	T2	100(IT2-T2)/T2	100(QT2-T2)/T2	T2	100(IT2-T2)/T2	100(QT2-T2)/T2
Centroid	0	0.528	18.746%	6.216%	5.136	1.026%	0.833%	9.450	-0.749%	-0.061%
	0.5	0.722	-13.163%	-6.179%	5.186	0.063%	0.123%	9.277	1.111%	0.587%
	1	0.805	-22.121%	-11.418%	5.192	-0.059%	0.018%	9.196	1.998%	1.066%
Cardinality	0	0.678	35.756%	13.924%	0.615	68.192%	26.536%	0.676	34.018%	13.249%
	0.5	0.910	1.207%	1.207%	1.015	1.913%	1.913%	0.896	1.160%	1.160%
	1	1.142	-19.275%	-6.299%	1.416	-26.859%	-8.753%	1.115	-18.709%	-6.117%
Fuzziness	0	0.079	20.398%	4.819%	0.108	23.389%	3.680%	0.078	21.618%	5.579%
	0.5	0.106	-11.117%	-3.131%	0.162	-17.675%	-5.292%	0.105	-9.943%	-2.771%
	1	0.118	-19.701%	-7.872%	0.164	-18.220%	-7.992%	0.118	-19.954%	-7.831%
Variance	0	0.383	23.284%	9.126%	0.657	16.096%	5.605%	0.367	24.536%	9.750%
	0.5	0.407	15.907%	7.914%	0.523	45.947%	25.513%	0.401	14.020%	7.082%
	1	0.404	17.062%	9.974%	0.506	50.988%	30.549%	0.401	14.022%	8.479%
Skewness	0	0.385	-113.03%	6.388%	0.253	-408.01%	-27.256%	-0.37	25.012%	6.598%
	0.5	0.255	-119.6%	28.507%	0.064	-1326.6%	89.576%	-0.26	81.023%	26.845%
	1	0.195	-125.8%	54.733%	0.046	-1808.3%	158.174%	-0.20	135.60%	51.995%

where $s_{\tilde{A},L}(x|\alpha)$, $s_{\tilde{A},R}(x|\alpha)$ and $s_{\tilde{B},L}(x|\alpha)$, $s_{\tilde{B},R}(x|\alpha)$ denote the *LMF* and *UMF* of $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, respectively. Based on (5-43) and (5-44), we have ($\forall\alpha \in [0,1]$):

$$\tilde{A}_\alpha \subseteq \tilde{B}_\alpha \tag{5-45}$$

Using (5-19) for $M_{\tilde{B}_\alpha}(\xi)$, it follows that:

$$\begin{aligned} M_{\tilde{B}_\alpha}(\xi) &= \bigcup_{\forall B_e(\alpha)} m(B_e(\alpha)) \\ &= \bigcup_{\forall B_e(\alpha)\in\tilde{A}_\alpha} m(B_e(\alpha)) + \bigcup_{\forall B_e(\alpha)\notin\tilde{A}_\alpha} m(B_e(\alpha)) \\ &= M_{\tilde{A}_\alpha}(\xi) + \bigcup_{\forall B_e(\alpha)\notin\tilde{A}_\alpha} m(B_e(\alpha)) \end{aligned} \tag{5-46}$$

where the second line has made use of (5-45). Consequently, the last line of (5-46) indicates:

$$M_{\tilde{A}_\alpha}(\xi) \subseteq M_{\tilde{B}_\alpha}(\xi) \tag{5-47}$$

Based on (5-28), (5-47) is equivalent to ($\forall\alpha \in [0,1]$):

$$M_{\tilde{A}_\alpha}(\xi|\alpha) \subseteq M_{\tilde{B}_\alpha}(\xi|\alpha) \tag{5-48}$$

(5-48) shows that each α-cut of the uncertainty measure for $\tilde{A}$ is always contained within the α-cut of the uncertainty measure for $\tilde{B}$. As a result, the uncertainty measure for $\tilde{A}$, $M_{\tilde{A}}(\xi)$, is always contained within the uncertainty measure for $\tilde{B}$, $M_{\tilde{B}}(\xi)$.■

Note that Fig. 5.6 shows a special case of Theorem 3, because the secondary *MF*s of a general T2 FS that satisfy (5-13) are always contained within the rectangular secondary *MF*s of the IT2 FS that has the same *FOU* as the general T2 FS.

One widely accepted interpretation for an uncertainty measure of an IT2 FS is that it is a uniformly weighted interval for which every uncertainty-measure-value between its left and right end points is weighted equally. On the other hand, the uncertainty measure for the general T2 FS has its values weighted differently over its support, which makes its interpretation not so straightforward. We, therefore, introduce the following summary measure for the T1 FS $M_{\tilde{A}}(\xi)$:

Definition 9. A *Summary Interval Uncertainty Measure* (*SIUM*), $SI[M_{\tilde{A}}(\xi)]$, for a general T2 FS $\tilde{A}$ is:

$$SI[M_{\tilde{A}}(\xi)] \equiv [c(M_{\tilde{A}}(\xi)) - SD(M_{\tilde{A}}(\xi)), c(M_{\tilde{A}}(\xi)) + SD(M_{\tilde{A}}(\xi))] \;■ \tag{5-49}$$

It follows from (5-49) that the length of $SI[M_{\tilde{A}}(\xi)]$ is:

$$|SI[M_{\tilde{A}}(\xi)]| = 2SD(M_{\tilde{A}}(\xi)) \tag{5-50}$$

The larger this value, which is determined only by $SD(M_{\tilde{A}}(\xi))$, the more uncertain is $\tilde{A}$ as measured by $M_{\tilde{A}}(\xi)$. The $|SI[M_{\tilde{A}}(\xi)]|$ values of the five uncertainty measures for the three words with triangle and trapezoidal secondary *MF*s and for the three values of w are summarized in Table VI. Observe that the fatter the *FOU* is (see Fig. 2), i.e., the more area it contains, the larger are the $|SI[M_{\tilde{A}}(\xi)]|$ values.

It is instructive to compare these *SIUMs* against the corresponding uncertainty measures for an IT2 FS that has the same *FOU* as the general T2 FS; therefore, the lengths of the five uncertainty measures, $|M_{\tilde{A}}(\xi)|$, for the three words with interval secondary *MF*s are also summarized in Table VI. Observe that:

$$|SI[M_{\text{word,triangle},w}(\xi)]| \;<\; |SI[M_{\text{word,trapezoid},w}(\xi)]| \;<\; |M_{\text{word,IT2}}(\xi)| \tag{5-51}$$

Comparing (5-40), (5-41) and (5-42), it is straightforward to show that, for the same value of w,

$$EP_l(x) \le Apex(x) \le EP_r(x) \tag{5-52}$$

which indicates that, at each x, for the same *FOU*, the triangle secondary MF is contained within the trapezoidal secondary MF, and, of course, both are contained within the rectangular *MF*. Then it follows from Theorem 3 that $M_{\text{word,triangle},w}(\xi)$ is contained within $M_{\text{word,trapezoid},w}(\xi)$, and both are contained within $M_{\text{word,IT2}}(\xi)$. It is intuitive that the inner MF has a smaller *SD* value than the outer MF, and, therefore, (5-51) follows.

5.6 Comparative Studies of T2, IT2 and QT2 FLSs

Mendel et al. [125] have proposed to use Quasi-Type-2 (QT2) FLSs in the place of full-blown T2 FLSs to reduce computational expenses. It has been shown in [125] that a QT2 FLS provides a good approximation when used to compute the centroid of a full-blown T2 FS with triangle secondary *MF*s. Additionally, a chaotic time series forecaster designed using a QT2 Fuzzy Logic System (FLS) outperformed its IT2 counterpart.

Similarly, while a general T2 FS can be viewed as a very large collection of α-level T2 FSs, one FS for each value of α, Hamrawi and Coupland [70] proposed the concept of QT2 FSs.

Definition 10. A QT2 FS, $\tilde{A}_Q$, is the union of only two α-level T2 FSs, i.e. $\tilde{A}_0$ and $\tilde{A}_1$:

$$\tilde{A}_Q = \tilde{A}_0 \cup \tilde{A}_1 \blacksquare \tag{5-53}$$

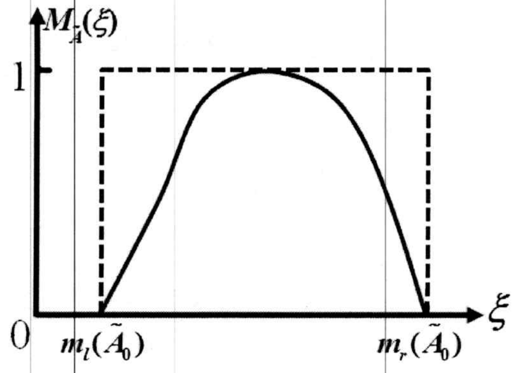


Figure 5.6 MF of a generic uncertainty measure for a general T2 FS (solid curve) and the MF of that generic uncertainty measure for an IT2 FS (dashed line) that has the same FOU as the general T2 FS.

In this section, comparative studies are performed to see whether QT2 or even IT2 FSs can be used to provide good enough approximations to all of our uncertainty measures for general T2 FSs.

Computing the uncertainty measure for a QT2 FS involves the following operations: 1) Compute the uncertainty measure for $\tilde{A}_0$, $M_{\tilde{A}_0}(\xi)$; 2) Compute the uncertainty measure for $\tilde{A}_1$, $M_{\tilde{A}_1}(\xi)$; and 3) Connect the end-points of $M_{\tilde{A}_0}(\xi)$ and $M_{\tilde{A}_1}(\xi)$.

Table 5.7 provides comparative results for the three words, triangle secondary *MF*s and the three values of *w*. In Table 5.7, each word has three columns, labeled T2, 100(IT2-T2)/T2 and 100(QT2-T2)/T2. The first column provides the centroids of the uncertainty measures for the T2 FSs, and the second and third columns provide the percentage differences between the centroids of the uncertainty measures for IT2 and T2 FSs, and QT2 and T2 FSs, respectively.

Table 5.8 provides the same kind of comparative results for the three words, trapezoidal secondary *MF*s and the three values of *w*.

A careful examination of Tables 5.7 and 5.8 reveals that, *except for the centroid for medium and very high amount, replacing a general T2 FS by either an IT2 FS or a QT2 FS gives large errors*; hence, we do not advocate such replacement for computing the uncertainty measures for general T2 FSs.

5.7 Summary

This section has extended five well-known T1 FS uncertainty measures (centroid, cardinality, fuzziness, variance and skewness) to general T2 FSs by using the WS RT. It has also shown how Liu's α-plane RT can be used to compute the five uncertainty measures of a general T2 FS by using existing algorithms for computing the comparable uncertainty measures for an IT2 FS. This is done one α-plane at a time.

Examples have been given that showed the five uncertainty measures for T2 FSs with different *FOU*s and triangle or trapezoidal secondary *MF*s. A *Summary Interval Uncertainty Measure* (*SIUM*) has been proposed for interpretive purposes that simplifies the uncertainty measure for a general T2 FS from a T1 FS to an interval. Our results have shown how the secondary MFs of a T2 FS can affect each uncertainty measure, namely *if the secondary MF of a T2 FS is contained within the secondary MF of another T2 FS, then the SIUM of the former T2 FS is narrower than the SIUM of the latter.*

Comparative studies of uncertainty measures for T2, IT2 and QT2 FSs have also been performed and showed that neither IT2 nor QT2 FSs can provide good enough approximations for the uncertainty measures for T2 FSs, except for the centroid of a small number of cases. Consequently, one should not replace a general T2 FS by either an IT2 or QT2 FS for such computations.

The approach used in this chapter can also be applied to define and compute similarity, ranking, subsethood and other important measures of general T2 FSs.

Chapter 6 Computing the Centroid of Type-2 Fuzzy Sets by means of Centroid-Flow Algorithm

In the previous chapter, we showed that the centroid of a general type-2 fuzzy set (T2 FS) $\tilde{A}$ can be obtained by taking the union of the centroids of all the α-planes (each raised to level α) of $\tilde{A}$. Karnik-Mendel (KM) or the Enhanced Karnik-Mendel (EKM) algorithms are used for computing the centroid of each α-plane. The iterative features in KM/EKM algorithms can be time-consuming, especially when the algorithms have to be repeated for many α - planes. This chapter proposes a new method named *Centroid Flow (CF) algorithm* to compute the centroid of $\tilde{A}$ without having to apply KM/EKM algorithms for every α – plane. Extensive simulations have shown that the CF algorithm can reduce the computation time by 75% to 80% and 50% to 75% compared to KM and EKM algorithms, respectively, and still maintains satisfactory computation accuracy for various T2 FSs when the primary variable x and α- plane are discretized finely enough.

6.1 Introduction

As studied in Chapter 5, a general Type-2 Fuzzy Set (T2 FS) is a T2 FS whose secondary memberships can take any value between [0,1]. Compared to an Interval T2 (IT2) FS whose secondary memberships all equal 1, a general T2 FS has more design degrees of freedom and, consequently, it is receiving more and more attention by T2 FS researchers (e.g., [42-45, 65, 66, 86, 106, 169, 170, 174, 181, 229]).

An important calculation for a general T2 FS is its *centroid*, because the centroid provides a measure of the uncertainty of such a FS, and the centroid may have to be computed during type-reduction for a general T2 Fuzzy Logic System (FLS) [87, 122]. The centroid of a general T2 FS $\tilde{A}$, developed by Karnik and Mendel [85, 122] is the union of the centroids of all the embedded T2 FSs of $\tilde{A}$. Unfortunately, this exhaustive and direct computation of the centroid is practically impossible, because it would require explicit enumeration of an extremely large number of embedded T2 FSs.

Different practical methods have been proposed to compute or approximate the actual centroid of $\tilde{A}$. Greenfield, et al. [66] proposed to randomly sample a number of embedded T2 FSs, and provided some examples that demonstrated that the number of randomly selected embedded sets only marginally affects the mean value of the centroid of $\tilde{A}$. Excellent results were obtained for as few as 10 randomly selected embedded T2 FSs, although it is still theoretically unknown why such a small number of randomly chosen embedded T2 FSs can lead to such good "defuzzified" results. This method does not provide the complete membership function (MF) of the centroid.

Coupland [42] recommends the use of the *x*-coordinate of the geometric centroid of the 3D MF of $\tilde{A}$. His approach goes directly from $\tilde{A}$ to a crisp number rather than to a T1 FS; hence, it does not provide a measure of the uncertainty in $\tilde{A}$. Additionally, when the T2 FS reduces to a T1 FS (when all uncertainty disappears), this *x*-coordinate does not reduce to the correct centroid of the T1 FS.

John and Czarnecki [9] and Lucas, et al. [111] have proposed to compute the centroid of $\tilde{A}$ as the centroids of all vertical slices (each of which is a T1 FS). This ad-hoc method leads to a centroid whose domain of non-zero centroid values always equals the domain of non-zero values of the primary variable of $\tilde{A}$, regardless of the rest of the geometry of the T2 FS. This does not seem to be a correct measure of the uncertainty in $\tilde{A}$, because the domain of non-zero values of the centroid of $\tilde{A}$ should depend upon the rest of the geometry of the T2 FS.

Most recently, Liu [106] has shown how to use an α-plane Representation Theorem (*RT*) (also called *Horizontal Slice RT*) of a T2 FS to compute the centroid of $\tilde{A}$. He has shown that the centroid of $\tilde{A}$ is the union of the centroids of all the α-level T2 FSs of $\tilde{A}$, where an α-level T2 FS is an α-plane raised to level α. In other words, it is similar to an IT2 FS, except that the secondary MFs all equal one for an IT2 FS, whereas they all equal α for an α-level T2 FS. The Karnik-Mendel (KM) algorithms [85, 86, 124], or the Enhanced KM (EKM) algorithms [190] are used to compute the centroid of each α-level T2 FS. Mendel and Liu [124] have shown that KM algorithms converge monotonically and super-exponentially fast, but several (usually two to six) iterations are always needed before convergence is reached. Consequently, when α is finely discretized, the iterative nature of KM (EKM) algorithms may become a computational burden that might hinder this method from being widely used.

This section introduces a new method for computing the centroid of a T2 FS, one that only utilizes KM (EKM) algorithms for computing the centroid of the $\alpha = 0$ α-plane, and then approximates the centroid of each of the remaining $\alpha > 0$ α-planes using the centroid of its previous α-plane and the structure of the secondary MFs. This method avoids using KM (EKM) algorithms for every α-plane of $\tilde{A}$. Examples in Section VII show that the centroid of $\tilde{A}$ can be obtained, with satisfactory accuracy and significantly reduced computation time, using the new algorithm, which we call *Centroid Flow* (CF) *algorithm*, because it lets the calculations "flow" from one α-plane to the next.

6.2 KM Algorithms and Their Properties

In this section, we review the continuous versions of KM algorithms [125, 127, 129] and two of their important properties. Note that the EKM algorithms are a more efficient way to process the KM calculations, and, the continuous version of the EKM algorithms are the same as the continuous version of the KM algorithms. The reason we are focusing on the continuous versions of these algorithms is because the CF algorithm is derived by using the two properties. Although the continuous KM algorithms are not always implemented on digital computers, they have been used in the past to gain a better understanding about convergence of the (discrete) KM algorithms [125], as well as properties about the centroid [127, 129], and, the two properties still hold as long as *x* is discretized finely enough.

The continuous KM algorithm for computing $c_l(\tilde{A}_\alpha)$ is:

1. Compute the initial value, $c_l(\tilde{A}_\alpha)_0$, for $c_l(\tilde{A}_\alpha)$, as:

$$c_l(\tilde{A}_\alpha)_0 = \frac{\int_{-\infty}^{+\infty} xs_R(x\,|\,\alpha)dx + \int_{-\infty}^{+\infty} xs_L(x\,|\,\alpha)dx}{\int_{-\infty}^{+\infty} s_R(x\,|\,\alpha)dx + \int_{-\infty}^{+\infty} s_L(x\,|\,\alpha)dx}; \tag{6-1}$$

then set *j*=1 and

$$b_1 = c_l(\tilde{A}_\alpha)_0. \tag{6-2}$$

2. Compute $c_l(\tilde{A}_\alpha)_j$ as

$$c_l(\tilde{A}_\alpha)_j = \frac{\int_{-\infty}^{b_j} xs_R(x \mid \alpha)dx + \int_{b_j}^{+\infty} xs_L(x \mid \alpha)dx}{\int_{-\infty}^{b_j} s_R(x \mid \alpha)dx + \int_{b_j}^{+\infty} s_L(x \mid \alpha)dx} \tag{6-3}$$

3. If convergence occurs, STOP; otherwise, go to step 4.

4. Set

$$b_{j+1} = c_l(\tilde{A}_\alpha)_j \tag{6-4}$$

5. Set $j=j+1$, and go to Step 2.

Because the continuous version of the KM algorithm for computing $c_r(\tilde{A}_\alpha)$ is very similar to the above algorithm, it is not stated here.

The two important properties of the continuous version of the KM algorithms, that are made very heavy use of in the sequel, are:

Theorem 1. [125, 127, 129] (a) For $c_l(\tilde{A}_\alpha)_j$ defined in (6-3),

$$\arg\min_{b_j}\{c_l(\tilde{A}_\alpha)_j\} = c_l(\tilde{A}_\alpha), \tag{6-5}$$

i.e.

$$c_l(\tilde{A}_\alpha) = \frac{\int_{-\infty}^{c_l(\tilde{A}_\alpha)} xs_R(x \mid \alpha)dx + \int_{c_l(\alpha)}^{+\infty} xs_L(x \mid \alpha)dx}{\int_{-\infty}^{c_l(\alpha)} s_R(x \mid \alpha)dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} s_L(x \mid \alpha)dx} \tag{6-6}$$

(b): For $c_r(\tilde{A}_\alpha)_j$ defined as

$$c_r(\tilde{A}_\alpha)_j = \frac{\int_{-\infty}^{b_j} x s_L(x \mid \alpha)dx + \int_{b_j}^{+\infty} x s_R(x \mid \alpha)dx}{\int_{-\infty}^{b_j} s_L(x \mid \alpha)dx + \int_{b_j}^{+\infty} s_R(x \mid \alpha)dx}, \tag{6-7}$$

$$\arg\max_{b_j}\{c_r(\tilde{A}_\alpha)_j\} = c_r(\tilde{A}_\alpha), \tag{6-8}$$

i.e.

$$c_r(\tilde{A}_\alpha) = \frac{\int_{-\infty}^{c_r(\tilde{A}_\alpha)} x s_L(x \mid \alpha)dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} x s_R(x \mid \alpha)dx}{\int_{-\infty}^{c_r(\tilde{A}_\alpha)} s_L(x \mid \alpha)dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} s_R(x \mid \alpha)dx} \tag{6-9}$$

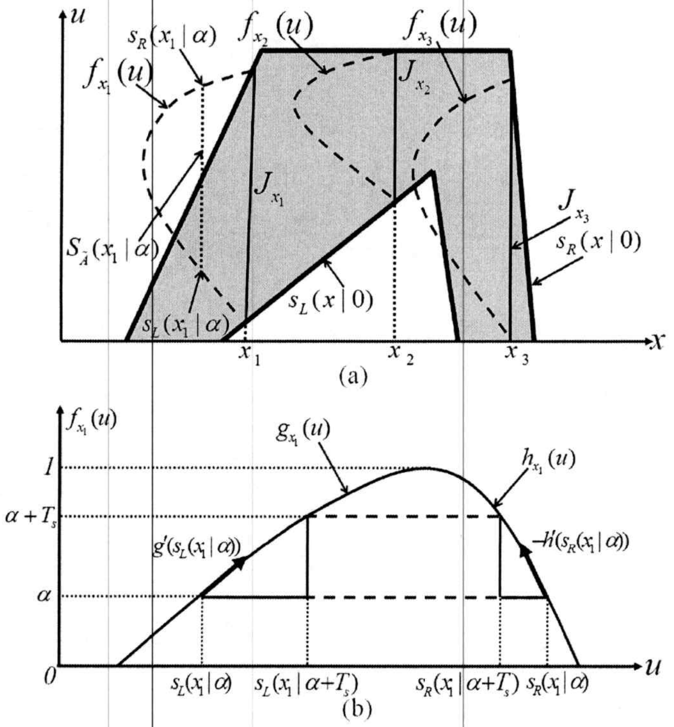


Figure 6.1 (a) *FOU* (shaded area) and secondary *MF*s (dashed curves) of a general T2 FS, (b) Secondary *MF* for the T2 FS from (a) at $x = x_1$.

where (6-6) shows that b_j that minimizes $c_l(\tilde{A}_\alpha)_j$ is itself the minimum value, $c_l(\tilde{A}_\alpha)$; and (6-9) shows that b_j that maximizes $c_r(\tilde{A}_\alpha)_j$ is itself the maximum value, $c_r(\tilde{A}_\alpha)$.

6.3 Alpha-Plane Connection Equations

Our CF algorithm depends on being able to convert one α-plane into a nearby α-plane. In this section, *connection equations* are derived that let us connect an α-plane to a nearby $(\alpha + T_s)$-plane, where T_s, as mentioned above, is a small increment. Note that we do not use $d\alpha$ for T_s, because the derivations of the CF algorithms involve integrals with respect to primary variable x, and include both dx and T_s. Calling T_s, $d\alpha$, could cause some confusion, because there are no integral with respect to α. It is assumed that $g_x(u)$ and $h_x(u)$ in (5-13) are both piecewise linear in the interval $[\alpha, \alpha + T_s]$, where T_s is small enough so that the gradients of $g_x(u)$ and $h_x(u)$ with respect to α, $g'_x(u)$ and $h'_x(u)$, are approximately constant in that interval.

From geometry, $g'_x(u)$ and $h'_x(u)$ in $[\alpha, \alpha + T_s]$ can be expressed as [see Fig. 6.1(b)]:

$$g'(s_L(x\,|\,\alpha)) \approx \frac{(\alpha+T_s)-\alpha}{s_L(x\,|\,\alpha+T_s)-s_L(x\,|\,\alpha)} \tag{6-10}$$

$$-h'(s_r(x\,|\,\alpha)) \approx \frac{(\alpha+T_s)-\alpha}{s_R(x\,|\,\alpha)-s_R(x\,|\,\alpha+T_s)} \tag{6-11}$$

where $g'(s_L(x\,|\,\alpha))$ is the gradient of $g_x(u)$ at $u = s_L(x\,|\,\alpha)$, and $h'(s_R(x\,|\,\alpha))$ is the gradient of $h_x(u)$ at $u = s_R(x\,|\,\alpha)$. Because $f_x(u)$ is pre-specified either numerically or analytically, so that $g'(s_L(x\,|\,\alpha))$ and $h'(s_r(x\,|\,\alpha))$ can be computed, then $s_L(x\,|\,\alpha+T_s)$ and $s_R(x\,|\,\alpha+T_s)$ can be obtained using (6-10) and (6-11), respectively, i.e.:

$$s_L(x\,|\,\alpha+T_s) \approx s_L(x\,|\,\alpha) + \frac{T_s}{g'(s_L(x\,|\,\alpha))} \tag{6-12}$$

$$s_R(x\,|\,\alpha+T_s) \approx s_R(x\,|\,\alpha) + \frac{T_s}{h'(s_R(x\,|\,\alpha))} \tag{6-13}$$

Any α-plane of $\tilde{A}$, $\tilde{A}_\alpha$, can be fully determined by its *LMF* and *UMF*, i.e. $s_L(x|\alpha)$ and $s_R(x|\alpha)$; therefore, (6-12) and (6-13) let $\tilde{A}_{\alpha+T_s}$ be determined [i.e. $s_L(x|\alpha+T_s)$ and $s_R(x|\alpha+T_s)$] by using $\tilde{A}_\alpha$ and the specific forms of the secondary MFs; hence, (6-12) and (6-13) are called *α-plane connection equations*.

6.4 Centroid Flow Algorithm

6.4.1 Alpha-Plane Centroid Connection Equations

The α-plane centroid connection equations that are derived in this section let $c_l(\tilde{A}_{\alpha+T_s})$ and $c_r(\tilde{A}_{\alpha+T_s})$ be computed using $c_l(\tilde{A}_\alpha)$ and $c_r(\tilde{A}_\alpha)$, respectively, and are summarized in Theorem 2. Although its details appear to be quite complicated, we shall demonstrate, in the rest of this chapter, that it is a very computationally efficient way to compute $c_l(\tilde{A}_{\alpha+T_s})$ and $c_r(\tilde{A}_{\alpha+T_s})$.

Theorem 2: When the objective is to compute $c_l(\tilde{A}_{\alpha+T_s})$, set [in (6-16)-(6-22)]

$$\begin{cases} c(\tilde{A}_\alpha) = c_l(\tilde{A}_\alpha) \\ s_1(x \mid \alpha) = s_R(x \mid \alpha) \\ s_2(x \mid \alpha) = s_L(x \mid \alpha) \\ L(x) = h(x) \\ R(x) = g(x) \\ i = 0 \end{cases}; \tag{6-14}$$

and, when the objective is to compute $c_r(\tilde{A}_{\alpha+T_s})$, set [in (6-16)-(6-22)]

$$\begin{cases} c(\tilde{A}_\alpha) = c_r(\tilde{A}_\alpha) \\ s_1(x \mid \alpha) = s_L(x \mid \alpha) \\ s_2(x \mid \alpha) = s_R(x \mid \alpha) \\ L(x) = g(x) \\ R(x) = h(x) \\ i = 1 \end{cases} \tag{6-15}$$

In order to compute $c(\tilde{A}_{\alpha+T_s})$, first compute:

$$B_\alpha = (-1)^i \left[s_1(c(\tilde{A}_\alpha) \mid \alpha) - s_2(c(\tilde{A}_\alpha) \mid \alpha) + \frac{T_s}{L'(s_1(c(\tilde{A}_\alpha) \mid \alpha))} - \frac{T_s}{R'(s_2(c(\tilde{A}_\alpha) \mid \alpha))} \right] \tag{6-16}$$

$$D_{1,\alpha} = T_s \left[\int_{-\infty}^{c(\tilde{A}_\alpha)} \frac{x}{L'(s_1(x \mid \alpha))} dx + \int_{c(\tilde{A}_\alpha)}^{+\infty} \frac{x}{R'(s_2(x \mid \alpha))} dx \right] \tag{6-17}$$

$$D_{2,\alpha} = T_s \left[\int_{-\infty}^{c(\tilde{A}_\alpha)} \frac{1}{L'(s_1(x \mid \alpha))} dx + \int_{c(\tilde{A}_\alpha)}^{+\infty} \frac{1}{R'(s_2(x \mid \alpha))} dx \right] \tag{6-18}$$

$$E_{1,\alpha} = \int_{-\infty}^{c(\tilde{A}_\alpha)} s_1(x \mid \alpha) dx + \int_{c(\tilde{A}_\alpha)}^{+\infty} s_2(x \mid \alpha) dx \tag{6-19}$$

$$E_{2,\alpha} = \int_{-\infty}^{c(\tilde{A}_\alpha)} xs_1(x\mid\alpha)dx + \int_{c(\tilde{A}_\alpha)}^{+\infty} xs_2(x\mid\alpha)dx \,; \tag{6-20}$$

then compute

$$\Delta_\alpha = \frac{\sqrt{(E_{1,\alpha}+D_{2,\alpha})^2 E_{1,\alpha}^2 + (-1)^i 2B_\alpha E_{1,\alpha}(D_{1,\alpha}E_{1,\alpha} - D_{2,\alpha}E_{2,\alpha})} - (E_{1,\alpha}+D_{2,\alpha})E_{1,\alpha}}{B_\alpha E_{1,\alpha}} ; \tag{6-21}$$

and, finally, compute

$$c(\tilde{A}_{\alpha+T_s}) = c(\tilde{A}_\alpha) + (-1)^i \Delta_\alpha \,\blacksquare \tag{6-22}$$

Proof of Theorem 2 when the objective is to compute $c_l(\tilde{A}_{\alpha+T_s})$:

To begin, let

$$\Delta_\alpha \equiv c_l(\tilde{A}_{\alpha+T_s}) - c_l(\tilde{A}_\alpha) \tag{6-23}$$

where, according to (6-6),

$$c_l(\tilde{A}_\alpha) = \frac{\int_{-\infty}^{c_l(\tilde{A}_\alpha)} xs_R(x\mid\alpha)dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} xs_L(x\mid\alpha)dx}{\int_{-\infty}^{c_l(\tilde{A}_\alpha)} s_R(x\mid\alpha)dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} s_L(x\mid\alpha)dx} \tag{6-24}$$

$$c_l(\tilde{A}_{\alpha+T_s}) = \frac{\int_{-\infty}^{c_l(\tilde{A}_{\alpha+T_s})} xs_R(x\mid\alpha+T_s)dx + \int_{c_l(\tilde{A}_{\alpha+T_s})}^{+\infty} xs_L(x\mid\alpha+T_s)dx}{\int_{-\infty}^{c_l(\tilde{A}_{\alpha+T_s})} s_R(x\mid\alpha+T_s)dx + \int_{c_l(\tilde{A}_{\alpha+T_s})}^{+\infty} s_L(x\mid\alpha+T_s)dx} \tag{6-25}$$

Note that when the secondary MFs of $\tilde{A}$ satisfy (5-13), it is true that $c_l(\tilde{A}_\alpha) \le c_l(\tilde{A}_{\alpha+T_s})$ for $\forall \alpha \in [0,1]$; consequently, Δ_α is guaranteed to be non-negative. Substituting (6-12), (6-13) and (6-23) into (6-24) , it follows that:

$$c_l(\tilde{A}_{\alpha+T_s}) \approx \frac{\int_{-\infty}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} x\left(s_R(x\mid\alpha)+\frac{T_s}{h'(s_R(x\mid\alpha))}\right)dx+\int_{c_l(\tilde{A}_\alpha)+\Delta_\alpha}^{+\infty} x\left(s_L(x\mid\alpha)+\frac{T_s}{g'(s_L(x\mid\alpha))}\right)dx}{\int_{-\infty}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} \left(s_R(x\mid\alpha)+\frac{T_s}{h'(s_R(x\mid\alpha))}\right)dx+\int_{c_l(\tilde{A}_\alpha)+\Delta_\alpha}^{+\infty} \left(s_L(x\mid\alpha)+\frac{T_s}{g'(s_L(x\mid\alpha))}\right)dx} \tag{6-26}$$

Next, let:

$$\begin{aligned}\text{Term1} \equiv &\int_{-\infty}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} x\left(s_R(x\mid\alpha)+\frac{T_s}{h'(s_R(x\mid\alpha))}\right)dx\\ &+\int_{c_l(\tilde{A}_\alpha)+\Delta_\alpha}^{+\infty} x\left(s_L(x\mid\alpha)+\frac{T_s}{g'(s_L(x\mid\alpha))}\right)dx\end{aligned} \tag{6-27}$$

$$\begin{aligned}\text{Term2} \equiv &\int_{-\infty}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} \left(s_R(x\mid\alpha)+\frac{T_s}{h'(s_R(x\mid\alpha))}\right)dx\\ &+\int_{c_l(\tilde{A}_\alpha)+\Delta_\alpha}^{+\infty} \left(s_L(x\mid\alpha)+\frac{T_s}{g'(s_L(x\mid\alpha))}\right)dx\end{aligned} \tag{6-28}$$

$$E_{1,\alpha} \equiv \int_{-\infty}^{c_l(\tilde{A}_\alpha)} s_R(x\mid\alpha)dx+\int_{c_l(\tilde{A}_\alpha)}^{+\infty} s_L(x\mid\alpha)dx \tag{6-29}$$

$$E_{2,\alpha} \equiv \int_{-\infty}^{c_l(\tilde{A}_\alpha)} xs_R(x\mid\alpha)dx+\int_{c_l(\tilde{A}_\alpha)}^{+\infty} xs_L(x\mid\alpha)dx \tag{6-30}$$

Using (6-27)-(6-30), (6-23) can be rewritten as:

$$\Delta_\alpha = \frac{\text{Term1}}{\text{Term2}} - \frac{E_{2,\alpha}}{E_{1,\alpha}} = \frac{\text{Term1}\times E_{1,\alpha} - \text{Term2}\times E_{2,\alpha}}{\text{Term2}\times E_{1,\alpha}} \tag{6-31}$$

Assuming that Δ_α is small, Term1 can be approximated as follows:

$$\begin{aligned}
\text{Term1} &= \int_{-\infty}^{c_l(\tilde{A}_\alpha)} x\left(s_R(x\,|\,\alpha) + \frac{T_s}{h'(s_R(x\,|\,\alpha))}\right)dx + \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} x\left(s_R(x\,|\,\alpha) + \frac{T_s}{h'(s_R(x\,|\,\alpha))}\right)dx \\
&\quad + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} x\left(s_L(x\,|\,\alpha) + \frac{T_s}{g'(s_L(x\,|\,\alpha))}\right)dx - \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} x\left(s_L(x\,|\,\alpha) + \frac{T_s}{g'(s_L(x\,|\,\alpha))}\right)dx \\
&\approx \int_{-\infty}^{c_l(\tilde{A}_\alpha)} x s_R(x\,|\,\alpha)dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} x s_L(x\,|\,\alpha)dx \\
&\quad + \int_{-\infty}^{c_l(\tilde{A}_\alpha)} x\frac{T_s}{h'(s_R(x\,|\,\alpha))}dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} x\frac{T_s}{g'(s_L(x\,|\,\alpha))}dx \\
&\quad + \Big[s_R(c_l(\tilde{A}_\alpha)\,|\,\alpha) - s_L(c_l(\tilde{A}_\alpha)\,|\,\alpha) \\
&\qquad + \frac{T_s}{h'(s_R(c_l(\tilde{A}_\alpha)\,|\,\alpha))} - \frac{T_s}{g'(s_L(c_l(\tilde{A}_\alpha)\,|\,\alpha))}\Big] \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} xdx \\
&= E_{2,\alpha} + D_{1,\alpha} + B_\alpha \frac{\Delta_\alpha^2 + 2\Delta_\alpha c_l(\tilde{A}_\alpha)}{2}
\end{aligned} \tag{6-32}$$

where B_α and $D_{1,\alpha}$ are defined, respectively, as:

$$B_\alpha \equiv s_R(c_l(\tilde{A}_\alpha)\,|\,\alpha) - s_L(c_l(\tilde{A}_\alpha)\,|\,\alpha) + \frac{T_s}{h'(s_R(c_l(\tilde{A}_\alpha)\,|\,\alpha))} - \frac{T_s}{g'(s_L(c_l(\tilde{A}_\alpha)\,|\,\alpha))} \tag{6-33}$$

$$D_{1,\alpha} \equiv T_s\left[\int_{-\infty}^{c_l(\tilde{A}_\alpha)} \frac{x}{h'(s_R(x\,|\,\alpha))}dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} \frac{x}{g'(s_L(x\,|\,\alpha))}dx\right] \tag{6-34}$$

Similarly, Term2 can be approximated as follows:

$$
\begin{aligned}
\text{Term2} &= \int_{-\infty}^{c_l(\tilde{A}_\alpha)} \left(s_R(x|\alpha) + \frac{T_s}{h'(s_R(x|\alpha))} \right) dx + \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta} \left(s_R(x|\alpha) + \frac{T_s}{h'(s_R(x|\alpha))} \right) dx \\
&+ \int_{c_l(\tilde{A}_\alpha)}^{+\infty} \left(s_L(x|\alpha) + \frac{T_s}{g'(s_L(x|\alpha))} \right) dx - \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta} \left(s_L(x|\alpha) + \frac{T_s}{g'(s_L(x|\alpha))} \right) dx \\
&\approx \int_{-\infty}^{c_l(\tilde{A}_\alpha)} s_R(x|\alpha) dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} s_L(x|\alpha) dx \\
&+ \int_{-\infty}^{c_l(\tilde{A}_\alpha)} \frac{T_s}{h'(s_R(x|\alpha))} dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} \frac{T_s}{g'(s_L(x|\alpha))} dx + \Delta_\alpha B_\alpha \\
&= E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha
\end{aligned}
\tag{6-35}
$$

where $D_{2,\alpha}$ is defined, as:

$$
D_{2,\alpha} \equiv T_s \left[\int_{-\infty}^{c_l(\tilde{A}_\alpha)} \frac{1}{h'(s_R(x|\alpha))} dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} \frac{1}{g'(s_L(x|\alpha))} dx \right]
\tag{6-36}
$$

Substituting (6-32) and (6-35) into (6-31), it follows that:

$$
\begin{aligned}
\Delta_\alpha &\approx \frac{\left[E_{2,\alpha} + D_{1,\alpha} + B_\alpha \dfrac{\Delta_\alpha^2 + 2\Delta_\alpha c_l(\tilde{A}_\alpha)}{2} \right] \times E_{1,\alpha} - (E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha) \times E_{2,\alpha}}{(E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha) \times E_{1,\alpha}} \\
&= \frac{D_{1,\alpha} \times E_{1,\alpha} - D_{2,\alpha} \times E_{2,\alpha} + \Delta_\alpha B_\alpha \times \left[c_l(\tilde{A}_\alpha) \times E_{1,\alpha} - E_{2,\alpha} \right] + \dfrac{\Delta_\alpha^2}{2} B_\alpha \times E_{1,\alpha}}{(E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha) \times E_{1,\alpha}} \\
&= \frac{D_{1,\alpha} \times E_{1,\alpha} - D_{2,\alpha} \times E_{2,\alpha} + \dfrac{\Delta_\alpha^2}{2} B_\alpha \times E_{1,\alpha}}{(E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha) \times E_{1,\alpha}}
\end{aligned}
\tag{6-36}
$$

The last line follows from (6-23) because

$$
c_l(\tilde{A}_\alpha) = \frac{E_{2,\alpha}}{E_{1,\alpha}}
\tag{6-37}
$$

Note that (6-36) is a quadratic equation in Δ_α, i.e.,

$$\Delta_\alpha\left[\Delta_\alpha B_\alpha E_{1,\alpha}+(E_{1,\alpha}+D_{2,\alpha})E_{1,\alpha}\right]=D_{1,\alpha}E_{1,\alpha}-D_{2,\alpha}E_{2,\alpha}+\frac{\Delta_\alpha^2}{2}B_\alpha E_{1,\alpha} \tag{6-38}$$

$$\frac{B_\alpha E_{1,\alpha}}{2}\Delta_\alpha^2+(E_{1,\alpha}+D_{2,\alpha})E_{1,\alpha}\Delta_\alpha+(D_{2,\alpha}E_{2,\alpha}-D_{1,\alpha}E_{1,\alpha})=0 \tag{6-39}$$

The roots of (6-39) can be obtained using the standard quadratic formula, and, because Δ_α is guaranteed to be non-negative, the following unique solution is obtained for Δ_α:

$$\Delta_\alpha=\frac{\sqrt{(E_{1,\alpha}+D_{2,\alpha})^2E_{1,\alpha}^2+2B_\alpha E_{1,\alpha}(D_{1,\alpha}E_{1,\alpha}-D_{2,\alpha}E_{2,\alpha})}-(E_{1,\alpha}+D_{2,\alpha})E_{1,\alpha}}{B_\alpha E_{1,\alpha}} \tag{6-40}$$

Finally, $c_l(\tilde{A}_{\alpha+T_s})$ can be computed using (6-23) once Δ_α is obtained, i.e.,

$$c_l(\tilde{A}_{\alpha+T_s})=c_l(\tilde{A}_\alpha)+\Delta_\alpha \tag{6-41}$$

Renaming $c_l(\tilde{A}_\alpha)$, $s_R(x\mid\alpha)$, $s_L(x\mid\alpha)$, $h(x)$ and $g(x)$ according to (6-14) and setting i = 0, (6-41) becomes equivalent to (6-22).■

Proof of Theorem 2 when the objective is to compute $c_r(\tilde{A}_{\alpha+T_s})$:

To begin, let

$$\Delta_\alpha\equiv c_r(\tilde{A}_\alpha)-c_r(\tilde{A}_{\alpha+T_s}) \tag{6-42}$$

where, according to (6-9),

$$c_r(\tilde{A}_\alpha)=\frac{\int_{-\infty}^{c_r(\tilde{A}_\alpha)}xs_L(x\mid\alpha)dx+\int_{c_r(\tilde{A}_\alpha)}^{+\infty}xs_R(x\mid\alpha)dx}{\int_{-\infty}^{c_r(\tilde{A}_\alpha)}s_L(x\mid\alpha)dx+\int_{c_r(\tilde{A}_\alpha)}^{+\infty}s_R(x\mid\alpha)dx} \tag{6-43}$$

$$c_r(\tilde{A}_{\alpha+T_s}) = \frac{\int_{-\infty}^{c_r(\tilde{A}_{\alpha+T_s})} x s_L(x \mid \alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha+T_s})}^{+\infty} x s_R(x \mid \alpha+T_s)dx}{\int_{-\infty}^{c_r(\tilde{A}_{\alpha+T_s})} s_L(x \mid \alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha+T_s})}^{+\infty} s_R(x \mid \alpha+T_s)dx} \tag{6-44}$$

Note that when the secondary MFs of $\tilde{A}$ satisfy (5-13), it is true that $c_r(\tilde{A}_\alpha) \geq c_r(\tilde{A}_{\alpha+d\alpha})$ for $\forall \alpha \in [0,1]$, consequently, Δ_α is guaranteed to be non-negative.

Substituting (6-12), (6-13) and (6-42) into (6-44), it follows that:

$$c_r(\tilde{A}_{\alpha+T_s}) \approx$$
$$\frac{\int_{-\infty}^{c_r(\tilde{A}_\alpha)-\Delta} x\left(s_L(x \mid \alpha) + \frac{T_s}{g'(s_L(x \mid \alpha))} \right)dx + \int_{c_r(\tilde{A}_\alpha)-\Delta}^{+\infty} x\left(s_R(x \mid \alpha) + \frac{T_s}{h'(s_R(x \mid \alpha))} \right)dx}{\int_{-\infty}^{c_r(\tilde{A}_\alpha)-\Delta} \left(s_L(x \mid \alpha) + \frac{T_s}{g'(s_L(x \mid \alpha))} \right)dx + \int_{c_r(\tilde{A}_\alpha)-\Delta}^{+\infty} \left(s_R(x \mid \alpha) + \frac{T_s}{h'(s_R(x \mid \alpha))} \right)dx} \tag{6-45}$$

Next, let:

$$E_{1,\alpha} \equiv \int_{-\infty}^{c_r(\tilde{A}_\alpha)} s_L(x \mid \alpha)dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} s_R(x \mid \alpha)dx \tag{6-46}$$

$$E_{2,\alpha} \equiv \int_{-\infty}^{c_r(\tilde{A}_\alpha)} x s_L(x \mid \alpha)dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} x s_R(x \mid \alpha)dx \tag{6-47}$$

$$\begin{aligned} \text{Term3} \equiv & \int_{-\infty}^{c_r(\tilde{A}_\alpha)-\Delta_\alpha} x\left(s_L(x \mid \alpha) + \frac{T_s}{g'(s_L(x \mid \alpha))} \right)dx \\ & + \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{+\infty} x\left(s_R(x \mid \alpha) + \frac{T_s}{h'(s_R(x \mid \alpha))} \right)dx \end{aligned} \tag{6-48}$$

$$\begin{aligned} \text{Term4} \equiv & \int_{-\infty}^{c_r(\tilde{A}_\alpha)-\Delta_\alpha} \left(s_L(x \mid \alpha) + \frac{T_s}{g'(s_L(x \mid \alpha))} \right)dx \\ & + \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{+\infty} \left(s_R(x \mid \alpha) + \frac{T_s}{h'(s_R(x \mid \alpha))} \right)dx \end{aligned} \tag{6-49}$$

Using (6-45)-(6-49), (6-42) can be rewritten as:

$$\Delta_\alpha = \frac{E_{2,\alpha}}{E_{1,\alpha}} - \frac{\text{Term3}}{\text{Term4}} = \frac{E_{2,\alpha}\times\text{Term4} - E_{1,\alpha}\times\text{Term3}}{E_{1,\alpha}\times\text{Term4}} \tag{6-50}$$

Assuming that Δ_α is small, Term3 can be approximated as follows:

$$\begin{aligned}
\text{Term3} =& \int_{-\infty}^{c_r(\tilde{A}_\alpha)} x\left(s_L(x\mid\alpha) + \frac{T_s}{g'(s_L(x\mid\alpha))}\right)dx - \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} x\left(s_L(x\mid\alpha) + \frac{T_s}{g'(s_L(x\mid\alpha))}\right)dx \\
&+ \int_{c_r(\tilde{A}_\alpha)}^{+\infty} x\left(s_R(x\mid\alpha) + \frac{T_s}{h'(s_R(x\mid\alpha))}\right)dx + \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} x\left(s_R(x\mid\alpha) + \frac{T_s}{h'(s_R(x\mid\alpha))}\right)dx \\
\approx& \int_{-\infty}^{c_r(\tilde{A}_\alpha)} xs_L(x\mid\alpha)dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} xs_R(x\mid\alpha)dx \\
&+ \int_{-\infty}^{c_r(\tilde{A}_\alpha)} \frac{xT_s}{g'(s_L(x\mid\alpha))}dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} \frac{xT_s}{h'(s_R(x\mid\alpha))}dx \\
&+\Big[s_R(c_r(\tilde{A}_\alpha)\mid\alpha) - s_L(c_r(\tilde{A}_\alpha)\mid\alpha) \\
&\qquad + \frac{T_s}{h'(s_R(c_r(\tilde{A}_\alpha)\mid\alpha))} - \frac{T_s}{g'(s_L(c_r(\tilde{A}_\alpha)\mid\alpha))}\Big]\int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} xdx \\
=& E_{2,\alpha} + D_{1,\alpha} + B_\alpha \frac{2\Delta_\alpha c_r(\tilde{A}_\alpha) - \Delta_\alpha^2}{2}
\end{aligned} \tag{6-51}$$

where B_α and $D_{1,\alpha}$ are defined, respectively, as:

$$B_\alpha = s_R(c_r(\tilde{A}_\alpha)\mid\alpha) - s_L(c_r(\tilde{A}_\alpha)\mid\alpha) + \frac{T_s}{h'(s_R(c_r(\tilde{A}_\alpha)\mid\alpha))} - \frac{T_s}{g'(s_L(c_r(\tilde{A}_\alpha)\mid\alpha))} \tag{6-52}$$

$$D_{1,\alpha} = T_s\left[\int_{-\infty}^{c_r(\tilde{A}_\alpha)} \frac{x}{g'(s_L(x\mid\alpha))}dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} \frac{x}{h'(s_R(x\mid\alpha))}dx\right] \tag{6-53}$$

Similarly, Term4 can be approximated as follows:

$$\begin{aligned}
\text{Term4} &= \int_{-\infty}^{c_r(\tilde{A}_\alpha)} \left(s_L(x \mid \alpha) + \frac{T_s}{g'(s_L(x \mid \alpha))} \right) dx - \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} \left(s_L(x \mid \alpha) + \frac{T_s}{g'(s_L(x \mid \alpha))} \right) dx \\
&\quad + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} \left(s_R(x \mid \alpha) + \frac{T_s}{h'(s_R(x \mid \alpha))} \right) dx + \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} \left(s_R(x \mid \alpha) + \frac{T_s}{h'(s_R(x \mid \alpha))} \right) dx \\
&= \int_{-\infty}^{c_r(\tilde{A}_\alpha)} s_L(x \mid \alpha) dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} s_R(x \mid \alpha) dx \\
&\quad + \int_{-\infty}^{c_r(\tilde{A}_\alpha)} \frac{T_s}{g'(s_L(x \mid \alpha))} dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} \frac{T_s}{h'(s_R(x \mid \alpha))} dx + \Delta_\alpha B_\alpha \\
&= E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha
\end{aligned} \tag{6-54}$$

where $D_{2,\alpha}$ is defined as follows:

$$D_{2,\alpha} = T_s \left[\int_{-\infty}^{c_r(\tilde{A}_\alpha)} \frac{1}{g'(s_L(x \mid \alpha))} dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} \frac{1}{h'(s_R(x \mid \alpha))} dx \right] \tag{6-55}$$

Substituting (6-51) and (6-54) into (6-50), it follows that:

$$\begin{aligned}
\Delta_\alpha &\approx \frac{E_{2,\alpha} \times (E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha) - E_{1,\alpha} \times \left[\mathrm{E}_{2,\alpha} + D_{1,\alpha} + B_\alpha \dfrac{2\Delta_\alpha c_r(\tilde{A}_\alpha) - \Delta_\alpha^2}{2} \right]}{E_{1,\alpha} \times (E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha)} \\
&= \frac{E_{2,\alpha} D_{2,\alpha} - E_{1,\alpha} D_{1,\alpha} + \Delta_\alpha B_\alpha \left[E_{2,\alpha} - E_{1,\alpha} \times c_r(\tilde{A}_\alpha) \right] + \dfrac{\Delta_\alpha^2}{2} B_\alpha \times E_{1,\alpha}}{E_{1,\alpha} \times (E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha)} \\
&= \frac{E_{2,\alpha} D_{2,\alpha} - E_{1,\alpha} D_{1,\alpha} + \dfrac{\Delta_\alpha^2}{2} B_\alpha \times E_{1,\alpha}}{E_{1,\alpha} \times (E_{1,\alpha} + D_{2,\alpha} + \Delta_\alpha B_\alpha)}
\end{aligned} \tag{6-56}$$

The last line follows from (6-43) because

$$c_r(\tilde{A}_\alpha) = \frac{E_{2,\alpha}}{E_{1,\alpha}} \tag{6-57}$$

(6-56) is a quadratic equation in Δ_α, i.e.,

$$\Delta_\alpha\left[\Delta_\alpha B_\alpha E_{1,\alpha} + (E_{1,\alpha} + D_{2,\alpha})E_{1,\alpha}\right] = E_{2,\alpha}D_{2,\alpha} - E_{1,\alpha}D_{1,\alpha} + \frac{\Delta^2}{2}B_\alpha E_{1,\alpha} \tag{6-58}$$

$$\frac{B_\alpha E_{1,\alpha}}{2}\Delta_\alpha^2 + (E_{1,\alpha} + D_{2,\alpha})E_{1,\alpha}\Delta_\alpha + (E_{1,\alpha}D_{1,\alpha} - E_{2,\alpha}D_{2,\alpha}) = 0 \tag{6-59}$$

The roots of (6-59) can be obtained using the standard quadratic formula, and, because Δ_α can only be non-negative, the following unique solution is obtained for Δ_α :

$$\Delta_\alpha = \frac{\sqrt{(E_{1,\alpha} + D_{2,\alpha})^2 E_{1,\alpha}^2 - 2B_\alpha E_{1,\alpha}(E_{1,\alpha}D_{1,\alpha} - E_{2,\alpha}D_{2,\alpha})} - (E_{1,\alpha} + D_{2,\alpha})E_{1,\alpha}}{B_\alpha E_{1,\alpha}} \tag{6-60}$$

Finally, $c_r(\tilde{A}_{\alpha+T_s})$ can be computed using (6-42) once Δ_α is obtained, i.e.,

$$c_r(\tilde{A}_{\alpha+T_s}) = c_r(\tilde{A}_\alpha) - \Delta_\alpha \tag{6-61}$$

Renaming $c_l(\tilde{A}_\alpha)$, $s_R(x \mid \alpha)$, $s_L(x \mid \alpha)$, $h(x)$ and $g(x)$ according to (6-15) and setting i = 1, (6-61) becomes equivalent to (6-22).■

Note that B_α, $D_{1,\alpha}$, $D_{2,\alpha}$, $E_{1,\alpha}$ and $E_{2,\alpha}$ are computable once $c_l(\tilde{A}_\alpha)$ or $c_r(\tilde{A}_\alpha)$ have been obtained.

Corollary 1: Assume $c(\tilde{A}_\alpha)$ has been computed, and the α-plane centroid connection equations have been used to compute B_α, $D_{1,\alpha}$, $D_{2,\alpha}$, $E_{1,\alpha}$, $E_{2,\alpha}$, Δ_α and $c(\tilde{A}_{\alpha+T_s})$, after which $B_{\alpha+T_s}$, $D_{1,\alpha+T_s}$, $D_{2,\alpha+T_s}$, $E_{1,\alpha+T_s}$, $E_{2,\alpha+T_s}$ and $\Delta_{\alpha+T_s}$ have to be computed in order to compute $c(\tilde{A}_{\alpha+2T_s})$. Instead of computing $E_{1,\alpha+T_s}$ and $E_{2,\alpha+T_s}$ directly by (6-19) and (6-20), they can be computed recursively, as ($\alpha = 0, T_s, 2T_s, ..., 1-T_s$):

$$E_{1,\alpha+T_s} = E_{1,\alpha} + D_{2,\alpha} + \int_{c(\tilde{A}_\alpha)}^{c(\tilde{A}_{\alpha+T_s})} \left[s_1(x \mid \alpha+T_s) - s_2(x \mid \alpha+T_s) \right] dx \tag{6-62}$$

$$E_{2,\alpha+T_s} = E_{2,\alpha} + D_{1,\alpha} + \int_{c(\tilde{A}_\alpha)}^{c(\tilde{A}_{\alpha+T_s})} x\left[s_1(x \mid \alpha+T_s) - s_2(x \mid \alpha+T_s) \right] dx \;\blacksquare \tag{6-63}$$

Derivations of (6-62) and (6-63) when the objective is to compute $c_l(\tilde{A}_{\alpha+T_s})$:

It follows from (6-23) and (6-29) that:

$$\begin{aligned} E_{1,\alpha+T_s} &= \int_{-\infty}^{c_l(\tilde{A}_{\alpha+T_s})} s_R(x \mid \alpha+T_s)dx + \int_{c_l(\tilde{A}_{\alpha+T_s})}^{+\infty} s_L(x \mid \alpha+T_s)dx \\ &= \int_{-\infty}^{c_l(\tilde{A}_\alpha)} s_R(x \mid \alpha+T_s)dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} s_L(x \mid \alpha+T_s)dx \\ &\quad + \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} s_R(x \mid \alpha+T_s)dx - \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} s_L(x \mid \alpha+T_s)dx \end{aligned} \tag{6-64}$$

Substituting (6-12) and (6-13) into the first two terms of (6-64), it follows, that:

$$\begin{aligned} E_{1,\alpha+T_s} &= \int_{-\infty}^{c_l(\tilde{A}_\alpha)} s_R(x \mid \alpha)dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} s_L(x \mid \alpha)dx \\ &\quad + \int_{-\infty}^{c_l(\tilde{A}_\alpha)} \frac{T_s}{h'(s_R(x \mid \alpha))}dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} \frac{T_s}{g'(s_L(x \mid \alpha))}dx \\ &\quad + \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} s_R(x \mid \alpha+T_s)dx - \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} s_L(x \mid \alpha+T_s)dx \\ &= E_{1,\alpha} + D_{2,\alpha} + \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_{\alpha+T_s})} \left[s_R(x \mid \alpha+T_s) - s_L(x \mid \alpha+T_s) \right] dx \end{aligned} \tag{6-65}$$

Similarly, it follows from (6-23) and (6-30) that:

$$\begin{aligned} E_{2,\alpha+T_s} &= \int_{-\infty}^{c_l(\tilde{A}_{\alpha+T_s})} x s_R(x \mid \alpha+T_s)dx + \int_{c_l(\tilde{A}_{\alpha+T_s})}^{+\infty} x s_L(x \mid \alpha+T_s)dx \\ &= \int_{-\infty}^{c_l(\tilde{A}_{\alpha})} x s_R(x \mid \alpha+T_s)dx + \int_{c_l(\tilde{A}_{\alpha})}^{+\infty} x s_L(x \mid \alpha+T_s)dx \\ &\quad + \int_{c_l(\tilde{A}_{\alpha})}^{c_l(\tilde{A}_{\alpha})+\Delta_\alpha} x s_R(x \mid \alpha+T_s)dx - \int_{c_l(\tilde{A}_{\alpha})}^{c_l(\tilde{A}_{\alpha})+\Delta_\alpha} x s_L(x \mid \alpha+T_s)dx \\ &= E_{2,\alpha} + D_{1,\alpha} + \int_{c_l(\tilde{A}_{\alpha})}^{c_l(\tilde{A}_{\alpha+T_s})} x\left[s_R(x \mid \alpha+T_s) - s_L(x \mid \alpha+T_s)\right]dx \end{aligned} \tag{6-66}$$

Renaming $c_l(\tilde{A}_\alpha)$, $s_R(x \mid \alpha)$, $s_L(x \mid \alpha)$, $h(x)$ and $g(x)$ according to (6-14), (6-65) and (6-66) become equivalent to (6-62) and (6-63), respectively.■

Derivations of (6-62) and (6-63) when the objective is to compute $c_r(\tilde{A}_{\alpha+T_s})$:

It follows from (6-42) and (6-46) that:

$$\begin{aligned} E_{1,\alpha+T_s} &= \int_{-\infty}^{c_r(\tilde{A}_{\alpha+T_s})} s_L(x \mid \alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha+T_s})}^{+\infty} s_R(x \mid \alpha+T_s)dx \\ &= \int_{-\infty}^{c_r(\tilde{A}_{\alpha})} s_L(x \mid \alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha})}^{+\infty} s_R(x \mid \alpha+T_s)dx \\ &\quad - \int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})} s_L(x \mid \alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})} s_R(x \mid \alpha+T_s)dx \end{aligned} \tag{6-67}$$

Substituting (6-12) and (6-13) into the first two terms of (6-67), it follows, that:

$$\begin{aligned} E_{1,\alpha+T_s} &= \int_{-\infty}^{c_r(\tilde{A}_{\alpha})} s_L(x \mid \alpha)dx + \int_{c_r(\tilde{A}_{\alpha})}^{+\infty} s_R(x \mid \alpha)dx \\ &\quad + \int_{-\infty}^{c_r(\tilde{A}_{\alpha})} \frac{T_s}{g'(s_L(x \mid \alpha))}dx + \int_{c_r(\tilde{A}_{\alpha})}^{+\infty} \frac{T_s}{h'(s_R(x \mid \alpha))}dx \\ &\quad - \int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})} s_L(x \mid \alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})} s_R(x \mid \alpha+T_s)dx \\ &= E_{1,\alpha} + D_{2,\alpha} + \int_{c_r(\tilde{A}_{\alpha+T_s})}^{c_r(\tilde{A}_{\alpha})} \left[s_R(x \mid \alpha+d\alpha) - s_L(x \mid \alpha+d\alpha)\right]dx \\ &= E_{1,\alpha} + D_{2,\alpha} + \int_{c_r(\tilde{A}_{\alpha})}^{c_r(\tilde{A}_{\alpha+T_s})} \left[s_L(x \mid \alpha+d\alpha) - s_R(x \mid \alpha+d\alpha)\right]dx \end{aligned} \tag{6-68}$$

Similarly, it follows from (6-42) and (6-47) that:

$$\begin{aligned}
E_{2,\alpha+T_s} &= \int_{-\infty}^{c_r(\tilde{A}_{\alpha+T_s})} xs_L(x\,|\,\alpha+T_s)dx + \int_{c_r(\tilde{A}_{\alpha+T_s})}^{+\infty} xs_R(x\,|\,\alpha+T_s)dx \\
&= \int_{-\infty}^{c_r(\tilde{A}_\alpha)} xs_L(x\,|\,\alpha)dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} xs_R(x\,|\,\alpha)dx \\
&\quad + \int_{-\infty}^{c_r(\tilde{A}_\alpha)} \frac{xT_s}{g'(s_L(x\,|\,\alpha))}dx + \int_{c_r(\tilde{A}_\alpha)}^{+\infty} \frac{xT_s}{h'(s_R(x\,|\,\alpha))}dx \\
&\quad - \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} xs_L(x\,|\,\alpha+T_s)dx + \int_{c_r(\tilde{A}_\alpha)-\Delta_\alpha}^{c_r(\tilde{A}_\alpha)} xs_R(x\,|\,\alpha+T_s)dx \\
&= E_{2,\alpha} + D_{1,\alpha} + \int_{c_r(\tilde{A}_{\alpha+T_s})}^{c_r(\tilde{A}_\alpha)} x\left[s_R(x\,|\,\alpha+d\alpha) - s_L(x\,|\,\alpha+d\alpha)\right]dx \\
&= E_{2,\alpha} + D_{1,\alpha} + \int_{c_r(\tilde{A}_\alpha)}^{c_r(\tilde{A}_{\alpha+T_s})} x\left[s_L(x\,|\,\alpha+d\alpha) - s_R(x\,|\,\alpha+d\alpha)\right]dx
\end{aligned} \qquad (6\text{-}69)$$

Finally, renaming $c_l(\tilde{A}_\alpha)$, $s_R(x\,|\,\alpha)$, $s_L(x\,|\,\alpha)$, $h(x)$ and $g(x)$ according to (6-15), (6-68) and (6-69) become equivalent to (6-62) and (6-63), respectively.■

6.4.2 Procedure of the CF Algorithm

The basic idea of the CF algorithm is to use the α-plane centroid connection equations in Theorem 2 and Corollary 1 to compute the centroids of all the α-planes instead of using KM algorithms, because, as is demonstrated later in this section, the α-plane centroid connection equations have much lower computational costs than do KM algorithms.

The complete *CF algorithm* for computing the centroid of a general T2 FS $\tilde{A}$ is:

1. Decide on how many α-planes will be used, where $\alpha \in [0,1]$. Call that number k; its choice will depend on the accuracy that is required. Regardless of k, $\alpha = 0$ and $\alpha = 1$ are always used. Note that the increment between adjacent α values is $T_s = 1/(k-1)$.

2. Use the KM (EKM) algorithms to compute the centroid of $\tilde{A}_0$, i.e., $C_{\tilde{A}_0}(\xi) = [c_l(\tilde{A}_0), c_r(\tilde{A}_0)]$.
3. Apply α-plane centroid connection equations, (6-14)-(6-22)(29)-(37), to compute B_0, $D_{1,0}$, $D_{2,0}$, $E_{1,0}$, $E_{2,0}$ and Δ_0 in order to compute the centroid of $\tilde{A}_{T_s}$, $C_{\tilde{A}_{T_s}}(\xi) = [c_l(\tilde{A}_{T_s}), c_r(\tilde{A}_{T_s})]$.
4. For all remaining α-planes ($\alpha = T_s$, $2T_s$,..., 1), apply the modified α-plane centroid connection equations, (6-14)-(6-18), (6-62), (6-63) and (6-22), to compute B_α, $D_{1,\alpha}$, $D_{2,\alpha}$, $E_{1,\alpha}$, $E_{2,\alpha}$ and Δ_α in order to compute the centroid of $\tilde{A}_{\alpha+T_s}$, $C_{\tilde{A}_{\alpha+T_s}}(\xi) = [c_l(\tilde{A}_{\alpha+T_s}), c_r(\tilde{A}_{\alpha+T_s})]$.
5. Collect all of the k $C_{\tilde{A}_\alpha}(\xi)$ by using (5-18), to obtain $C_{\tilde{A}}(\xi)$.

The computational efficiencies of the CF algorithm, as compared to those of the KM (EKM) algorithms are examined next.

In Step 2 of the CF algorithm, when $c_l(\tilde{A}_0)$ or $c_r(\tilde{A}_0)$ is computed, the CF algorithm uses KM (EKM) algorithms, so the amount of computation is the same for both kinds of algorithms.

In Step 3, when $c_l(\tilde{A}_{T_s})$ or $c_r(\tilde{A}_{T_s})$ is computed, the most time-consuming computations are $D_{1,0}$, $D_{2,0}$, $E_{1,0}$ and $E_{2,0}$, each of which contains two integrals, resulting in a total of eight integrals. On the other hand, one iteration of a continuous KM (EKM) algorithm involves four integrals [e.g., see (6-3)]; so, if the KM (EKM) algorithm converges in N iterations, $4N$ integrals must be computed. Consequently, unless the KM (EKM) algorithms converge in less than two iterations, they will have more integrals to compute than will the CF algorithm. Although the CF algorithms have more addition and multiplication operations than do the KM algorithms, the computational costs of these operations are much smaller than those of integrals, and have therefore been ignored.

In Step 4, when $c_l(\tilde{A}_{\alpha+T_s})$ [or $c_r(\tilde{A}_{\alpha+T_s})$] is computed for all remaining α-planes, similar analyses as just done for Step 3 can be made. Furthermore, as shown in (6-62) and (6-63), $E_{1,\alpha}$ and $E_{2,\alpha}$ now contain only one integral each; and the integration range $[c(\tilde{A}_\alpha), c(\tilde{A}_{\alpha+T_s})]$ is usually much smaller than the integration ranges $[-\infty, c(\tilde{A}_{\alpha+T_s})]$ or $[c(\tilde{A}_{\alpha+T_s}), +\infty]$ used in (6-3); hence, computation time in Step 4 of the CF algorithm will be much less than in Step 3.

Consequently, in theory, the CF algorithm should be computationally much more efficient than the KM algorithms. This is verified by the examples in Section VII.

6.4.3 Further Computation Reduction when using Triangle or Trapezoidal Secondary MFs

Further computation reductions can be achieved for the α-plane centroid connection equations when the secondary MFs are either triangles or trapezoids. In both cases, $g_x(u)$ and $h_x(u)$ in (5-13) will the be linear; thus, $g'_x(u)$ and $h'_x(u)$ will be constants that can be computed using geometry [use a figure like Fig. 6.3(a) in which $g_x(u)$ and $h_x(u)$are straight lines], as:

$$g'_x(u) = \frac{1}{s_L(x|1) - s_L(x|0)} \tag{6-70}$$

$$h_x'(u) = \frac{1}{s_R(x\,|\,1) - s_R(x\,|\,0)} \tag{6-71}$$

Assume $c(\tilde{A}_\alpha)$ has been computed, and, the α-plane centroid connection equations have been used to compute B_α, $D_{1,\alpha}$, $D_{2,\alpha}$, $E_{1,\alpha}$, $E_{2,\alpha}$, Δ_α and $c(\tilde{A}_{\alpha+T_s})$, after which $B_{\alpha+T_s}$, $D_{1,\alpha+T_s}$, $D_{2,\alpha+T_s}$, $E_{1,\alpha+T_s}$, $E_{2,\alpha+T_s}$ and $\Delta_{\alpha+T_s}$ have been computed in order to compute $c(\tilde{A}_{\alpha+2T_s})$. Based on (6-70) and (6-71), it is straightforward to show that $D_{1,\alpha+T_s}$ and $D_{2,\alpha+T_s}$ can also be computed, recursively, as ($\alpha = 0$, T_s, $2T_s$,…, $1 - T_s$):

$$D_{1,\alpha+T_s} = D_{1,\alpha} + T_s\left[\int_{c(\tilde{A}_\alpha)}^{c(\tilde{A}_{\alpha+T_s})} x(s_1(x\,|\,1) - s_1(x\,|\,0) - s_2(x\,|\,1) + s_2(x\,|\,0))dx\right] \tag{6-72}$$

$$D_{2,\alpha+T_s} = D_{2,\alpha} + T_s\left[\int_{c(\tilde{A}_\alpha)}^{c(\tilde{A}_{\alpha+T_s})} (s_1(x\,|\,1) - s_1(x\,|\,0) - s_2(x\,|\,1) + s_2(x\,|\,0))dx\right] \tag{6-73}$$

Derivations of (6-72) and (6-73) when the objective is to compute $c_l(\tilde{A}_{\alpha+T_s})$:

It follows from (6-70), (6-71) and (6-34) that:

$$\begin{aligned}
D_{1,\alpha+T_s} &= T_s\left[\int_{-\infty}^{c_l(\tilde{A}_{\alpha+T_s})} x(s_R(x\,|\,1) - s_R(x\,|\,0))dx + \int_{c_l(\tilde{A}_{\alpha+T_s})}^{+\infty} x(s_L(x\,|\,1) - s_L(x\,|\,0))dx\right] \\
&= T_s\left[\int_{-\infty}^{c_l(\tilde{A}_\alpha)} x(s_R(x\,|\,1) - s_R(x\,|\,0))dx + \int_{c_l(\tilde{A}_\alpha)}^{+\infty} x(s_L(x\,|\,1) - s_L(x\,|\,0))dx\right. \\
&\qquad \left. + \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} x(s_R(x\,|\,1) - s_R(x\,|\,0))dx - \int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_\alpha)+\Delta_\alpha} x(s_L(x\,|\,1) - s_L(x\,|\,0))dx\right] \\
&= D_{1,\alpha} + T_s\left[\int_{c_l(\tilde{A}_\alpha)}^{c_l(\tilde{A}_{\alpha+T_s})} x(s_R(x\,|\,1) - s_R(x\,|\,0) - s_L(x\,|\,1) + s_L(x\,|\,0))dx\right]
\end{aligned} \tag{6-74}$$

Similarly, it follows from (6-70), (6-71) and (6-36) that:

$$\begin{aligned}
D_{2,\alpha+T_s} &= T_s\left[\int_{-\infty}^{c_l(\tilde{A}_{\alpha+T_s})}(s_R(x\,|\,1)-s_R(x\,|\,0))dx+\int_{c_l(\tilde{A}_{\alpha+T_s})}^{+\infty}(s_L(x\,|\,1)-s_L(x\,|\,0))dx\right] \\
&= T_s\left[\int_{-\infty}^{c_l(\tilde{A}_{\alpha})}(s_R(x\,|\,1)-s_R(x\,|\,0))dx+\int_{c_l(\tilde{A}_{\alpha})}^{+\infty}(s_L(x\,|\,1)-s_L(x\,|\,0))dx\right. \\
&\qquad \left.+\int_{c_l(\tilde{A}_{\alpha})}^{c_l(\tilde{A}_{\alpha})+\Delta_\alpha}(s_R(x\,|\,1)-s_R(x\,|\,0))dx-\int_{c_l(\tilde{A}_{\alpha})}^{c_l(\tilde{A}_{\alpha})+\Delta_\alpha}(s_L(x\,|\,1)-s_L(x\,|\,0))dx\right] \\
&= D_{2,\alpha}+T_s\left[\int_{c_l(\tilde{A}_{\alpha})}^{c_l(\tilde{A}_{\alpha+T_s})}(s_R(x\,|\,1)-s_R(x\,|\,0)-s_L(x\,|\,1)+s_L(x\,|\,0))dx\right]
\end{aligned} \tag{6-75}$$

Finally, renaming $c_l(\tilde{A}_\alpha)$, $s_R(x\,|\,\alpha)$, $s_L(x\,|\,\alpha)$, $h(x)$ and $g(x)$ according to (6-14), (6-74) and (6-75) become equivalent to (6-72) and (6-73), respectively.■

Derivations of (6-72) and (6-73) when the objective is to compute $c_r(\tilde{A}_{\alpha+T_s})$:

It follows from (6-70), (6-71) and (A-32) that:

$$\begin{aligned}
D_{1,\alpha+T_s} &= T_s\left[\int_{-\infty}^{c_r(\tilde{A}_{\alpha+T_s})}x(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha+T_s})}^{+\infty}x(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right] \\
&= T_s\left[\int_{-\infty}^{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}x(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{+\infty}x(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right] \\
&= T_s\left[\int_{-\infty}^{c_r(\tilde{A}_{\alpha})}x(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha})}^{+\infty}x(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right. \\
&\qquad \left.-\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})}x(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})}x(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right] \\
&= D_{1,\alpha}+T_s\left[\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})}x(s_R(x\,|\,1)-s_R(x\,|\,0)-s_L(x\,|\,1)+s_L(x\,|\,0))dx\right] \\
&= D_{1,\alpha}+T_s\left[\int_{c_r(\tilde{A}_{\alpha})}^{c_r(\tilde{A}_{\alpha+T_s})}x(s_L(x\,|\,1)-s_L(x\,|\,0)-s_R(x\,|\,1)+s_R(x\,|\,0))dx\right]
\end{aligned} \tag{6-76}$$

Similarly, it follows from (6-70), (6-71) and (6-36) that:

$$
\begin{aligned}
D_{2,\alpha+T_s} &= T_s\left[\int_{-\infty}^{c_r(\tilde{A}_{\alpha+T_s})}(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha+T_s})}^{+\infty}(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right] \\
&= T_s\left[\int_{-\infty}^{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{+\infty}(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right] \\
&= T_s\left[\int_{-\infty}^{c_r(\tilde{A}_{\alpha})}(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha})}^{+\infty}(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right. \\
&\qquad \left.-\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})}(s_L(x\,|\,1)-s_L(x\,|\,0))dx+\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})}(s_R(x\,|\,1)-s_R(x\,|\,0))dx\right] \\
&= D_{2,\alpha}+T_s\left[\int_{c_r(\tilde{A}_{\alpha})-\Delta_\alpha}^{c_r(\tilde{A}_{\alpha})}(s_R(x\,|\,1)-s_R(x\,|\,0)-s_L(x\,|\,1)+s_L(x\,|\,0))dx\right] \\
&= D_{2,\alpha}+T_s\left[\int_{c_r(\tilde{A}_{\alpha})}^{c_r(\tilde{A}_{\alpha+T_s})}(s_L(x\,|\,1)-s_L(x\,|\,0)-s_R(x\,|\,1)+s_R(x\,|\,0))dx\right]
\end{aligned}
\tag{6-77}
$$

Finally, renaming $c_l(\tilde{A}_\alpha)$, $s_R(x\,|\,\alpha)$, $s_L(x\,|\,\alpha)$, $h(x)$ and $g(x)$ according to (6-15), (6-76) and (6-77) become equivalent to (6-72) and (6-73), respectively.■

Therefore, when the CF algorithm is applied for a T2 FS $\tilde{A}$, that has either triangle or trapezoidal secondary MFs, (6-72) and (6-73) are used in Step 4 instead of (6-17) and (6-18), respectively.

Consequently, in theory, the CF algorithm for T2 FSs with triangle or trapezoidal secondary MFs should be even more computationally efficient than for other kinds of secondary MFs, and even more computationally efficient than the KM and EKM algorithms.

6.5 Comparative Studies

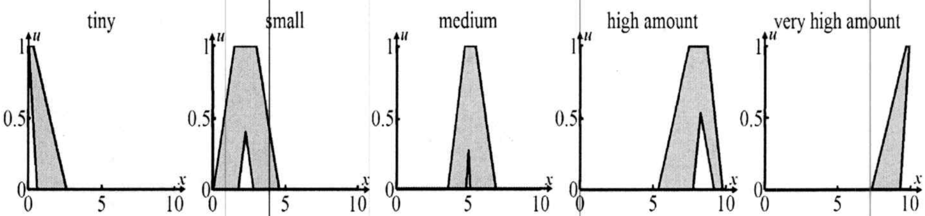


Figure 6.2 Five *FOU*s from a sub-vocabulary selected from the 32-word codebook in [111].

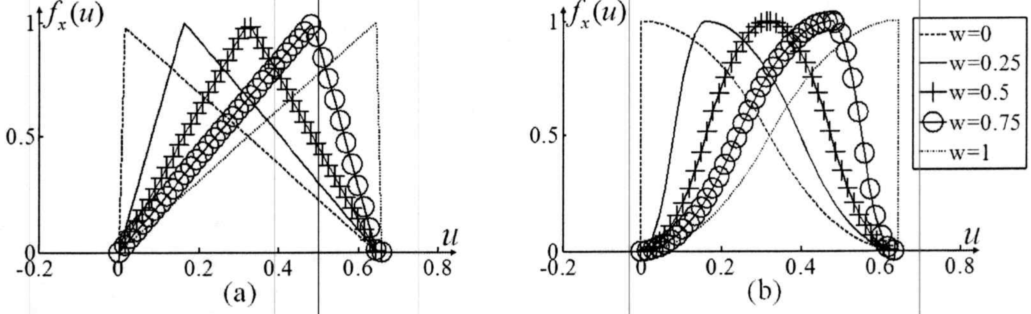


Figure 6.3 (a) Five triangle secondary MFs, and (b) five bell-shaped secondary MFs, when $x = 6$, all parameterized by w [see (6-78)].

Table 6.1 *FOU* data for five words. Each LMF and UMF is represented as a trapezoid (a, b, c, d), where a, b, c and d are the locations of the left end of its base, the left end of its top, the right end of its top and the right end of its base, respectively. The fifth parameter for the LMF is its height.

Word	*LMF*	*UMF*
Tiny	(0, 0, 0.05, 0.63, 1)	(0, 0, 0.36, 2.63)
Small	(1.79, 2.28, 2.28, 2.81, 0.40)	(0.09, 1.50, 3.00, 4.62)
Medium	(4.86, 5.03, 5.03, 5.14, 0.27)	(3.59, 4.75, 5.50, 6.91)
High amount	(7.79, 8.30, 8.30, 9.21, 0.53)	(5.38, 7.50, 8.75, 9.81)
Very high amount	(9.34, 9.95, 10, 10, 1)	(7.37, 9.73, 10, 10)

In this section, comparative studies are performed for KM and EKM algorithms [190] and the CF Algorithm, for both accuracy and computation time. Note that the EKM algorithms are more efficient than the original KM algorithms in the following ways: 1) An optimal initial value is used; 2) A better termination condition is used to remove an unnecessary iteration; and, 3) Each iteration utilizes available values from the previous iteration to reduce redundant computations. The EKM algorithms are included in the simulations to test whether CF algorithms have the ability to out-perform the best kind of KM algorithms. The simulation environment is a HP Compaq 6730s running Windows Vista Business and Matlab 2009a with Intel(R) Core(TM) 2 Duo CPU P8600@2.40GHz and 4.00 GB RAM.

The T2 FSs used in our studies were selected as follows:

1) Five representative *FOUs* were chosen from the 32-word codebook in [111] for the words *Tiny*, *Small*, *Medium*, *High Amount* and *Very High Amount*. Their *FOU*s, $FOU(\tilde{A})$, are depicted in Fig. 6.2 and the parameters of their *UMF*s and *LMF*s are given in Table 6.1. These five *FOU*s were chosen because they have quite different shapes, overlap with neighboring *FOU*s, successively cover different spans of the primary variable's domain, [0,10], and together cover this domain. Each word can be considered as a separate example.

2) For each word, two classes of secondary MFs were used, namely triangles and bell-shaped functions. (a) The triangle secondary MFs have base equal to $s_R(x|0) - s_L(x|0)$ and apex location, *Apex*(*x*), parameterized as [123] (*w*=0, 0.25, 0.5, 0.75, 1) :

$$Apex(x) = s_L(x|0) + w[s_R(x|0) - s_L(x|0)] \tag{6-78}$$

An example of the triangle secondary *MF*s for the word *Medium* is shown in Fig. 3(a) at *x* = 6 for all five *w* values. (b) The bell-shaped functions also have base equal to $s_R(x|0) - s_L(x|0)$ and apex location, *Apex*(*x*), also parameterized as (6-78) (*w*=0, 0.25, 0.5, 0.75, 1). The secondary MF on the left of the apex location, $g_x(u)$, is an S-shaped MF; and, the secondary MF on the right, $h_x(u)$, is a reverse S-shaped MF. An example of the bell-shaped secondary MFs for the word *Medium* is shown in Fig. 3(b) at *x* = 6 for all five *w* values.

So, for each of the five words, there are two kinds of secondary MFs, each with five cases; hence, there are $5 \times 2 \times 5 = 50$ general T2 FSs in our comparative studies.

Table 6.2 The *FOU*s, types of secondary MFs [Triangles (T) or Bell-shaped (B)] and w values of the 10 selected T2 FSs.

	$\tilde{A}_1$	$\tilde{A}_2$	$\tilde{A}_3$	$\tilde{A}_4$	$\tilde{A}_5$	$\tilde{A}_6$	$\tilde{A}_7$	$\tilde{A}_8$	$\tilde{A}_9$	$\tilde{A}_{10}$
FOU[a]	*Tiny*	*Small*	*Medium*	*High amount*	*Very high amount*	*Tiny*	*Small*	*Medium*	*High amount*	*Very high amount*
Secondary MF[b]	T	T	T	T	T	B	B	B	B	B
w	0	0.5	0.25	1	0.75	0	0.5	0.25	1	0.75

a. *FOU*s are in Fig. 6.2.

b. Secondary MFs are in Figs. 6.3(a) and 6.3(b).

6.5.1 Discretization of the Primary Variable *x*

In this section, the number of α-planes, k, is fixed at 25, i.e., T_s = 0.0417 (in Section 6.5.2, the discretization of α-planes is examined and k = 25 is shown to produce accurate results). As mentioned above, the domain of the primary variable x is [0, 10]. The KM and CF algorithms (The EKM algorithms produce exactly the same numerical results as KM algorithms, so only the KM algorithms are studied for accuracy tests.) were applied to all 50 T2 FSs using primary variable sampling intervals, T_x, of 0.5, 0.1, 0.01 and 0.001. It was observed that both the KM and CF algorithms converged in all 50 cases for $T_x \leq 0.01$. Due to space limitations, results are shown here only for 10 of the 50 T2 FSs, labeled $\tilde{A}_1$, $\tilde{A}_2$,..., $\tilde{A}_9$ and $\tilde{A}_{10}$, whose names and parameters are given in Table II. Observe that triangle (bell-shaped) secondary MFs were used for $\tilde{A}_1$-$\tilde{A}_5$ ($\tilde{}_6$-$\tilde{A}_{10}$).

The centroids of $\tilde{A}_1$-$\tilde{A}_5$ and $\tilde{A}_6$-$\tilde{A}_{10}$, computed using KM (solid lines) and CF (circles) algorithms are depicted in Figs. 4 and 5, respectively. Note, here and in several subsequent figures, that although the solid lines appear to be continuous, they were obtained by connecting only the end points of the centroids of the α-Plane. Scanning each row of both figures from left to right, one observes the convergence of the centroid as T_x varies from 0.5 to 0.1 to 0.01 and to 0.001. It is evident that when $T_x \leq 0.01$, the centroids computed by the KM and CF algorithms are approximately the same. Consequently, x is sampled at 0.01 in all of our remaining simulations.

Note that it may be possible to achieve convergence when $0.01 \leq T_x < 0.1$; but, this wasn't examined since our main focus is on convergence with respect to the sampling of α.

6.5.2 Accuracy Tests (Discretization of alpha)

It is important to study how the centroid computed by the CF algorithm compares to the centroid computed by the KM algorithms as a function of k. Let $C_{\tilde{A},KM}(\xi|k)$ and $C_{\tilde{A},CF}(\xi|k)$ denote the centroids computed by KM and CF algorithms, respectively, when k α-planes are used. For all 50 T2 FSs, $C_{\tilde{A},KM}(\xi|k)$ and $C_{\tilde{A},CF}(\xi|k)$ were computed and we observed that they gradually converged as k was increased from 2 to 35 α-planes. To illustrate this, $C_{\tilde{A},KM}(\xi|k)$ and $C_{\tilde{A},CF}(\xi|k)$ (for k = 5, 10, 15 and 20) are depicted in Figs. 6.6 and 6.7 for $\tilde{A}_1$-$\tilde{A}_5$ and $\tilde{A}_6$-$\tilde{A}_{10}$, respectively. And corresponding results for k = 25 have already been given in Figs. 6.4 and 6.5. As can be seen, results from the CF algorithm match those from the KM algorithms better for larger k (i.e., k = 15, 20 and 25).

In order to quantify the comparisons between $C_{\tilde{A},KM}(\xi|k)$ and $C_{\tilde{A},CF}(\xi|k)$, we used the following absolute difference, $d_{\tilde{A}}(k)$, between the defuzzified values of $C_{\tilde{A},KM}(\xi|k)$ and $C_{\tilde{A},CF}(\xi|k)$, i.e. $c(C_{\tilde{A},KM}(\xi|k))$ and $c(C_{\tilde{A},CF}(\xi|k))$, respectively:

$$d_{\tilde{A}}(k) = \left| c(C_{\tilde{A},KM}(\xi \mid k)) - c(C_{\tilde{A},CF}(\xi \mid k)) \right| \tag{6-79}$$

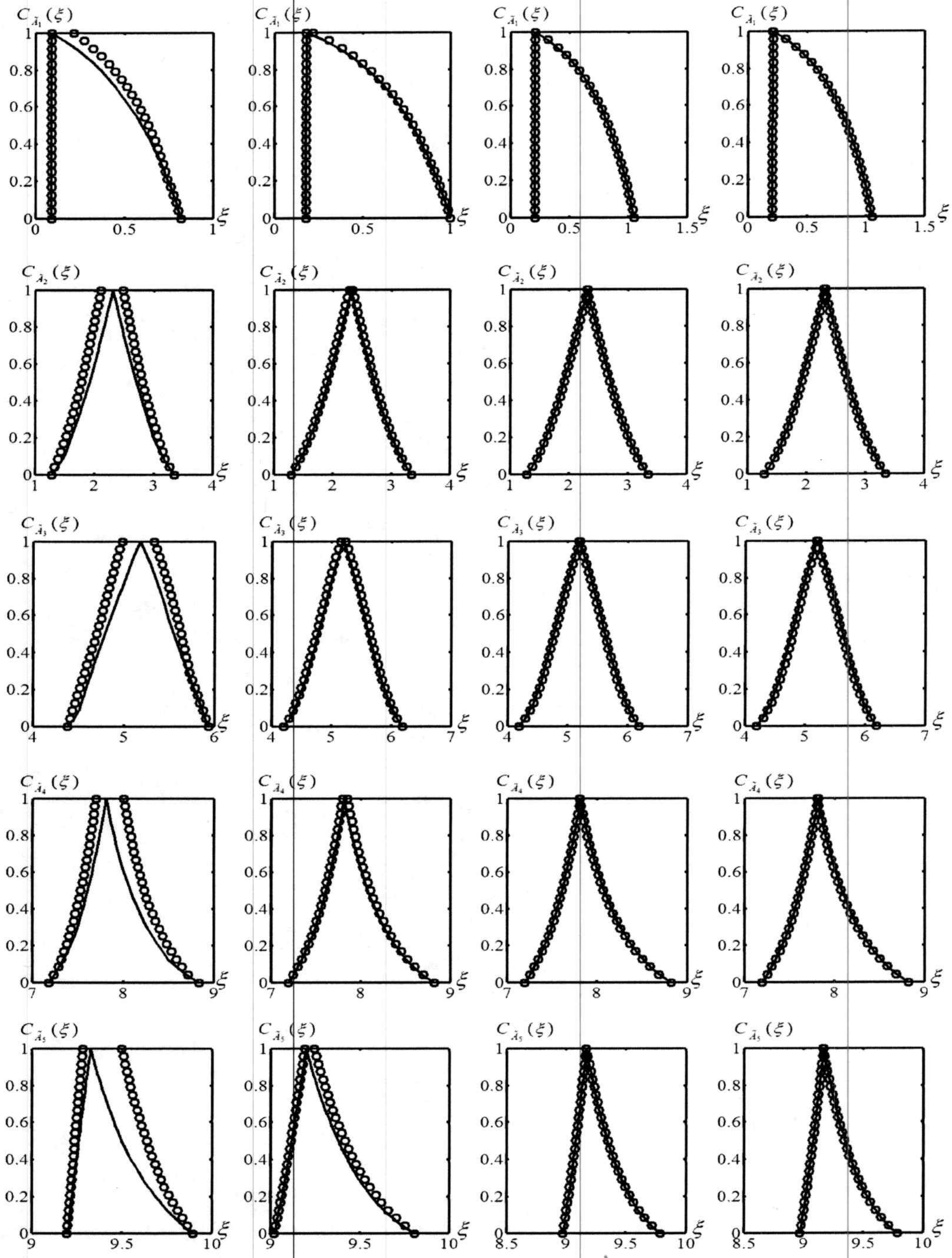


Figure 6.4 Centroids computed by KM (solid lines) and CF (circles) algorithm for $\tilde{A}_1,\ldots,\tilde{A}_5$ (each row is for an $\tilde{A}_i$) and for different primary variable sampling intervals of 0.5, 0.1, 0.01 and 0.001 (from left to right in each row).

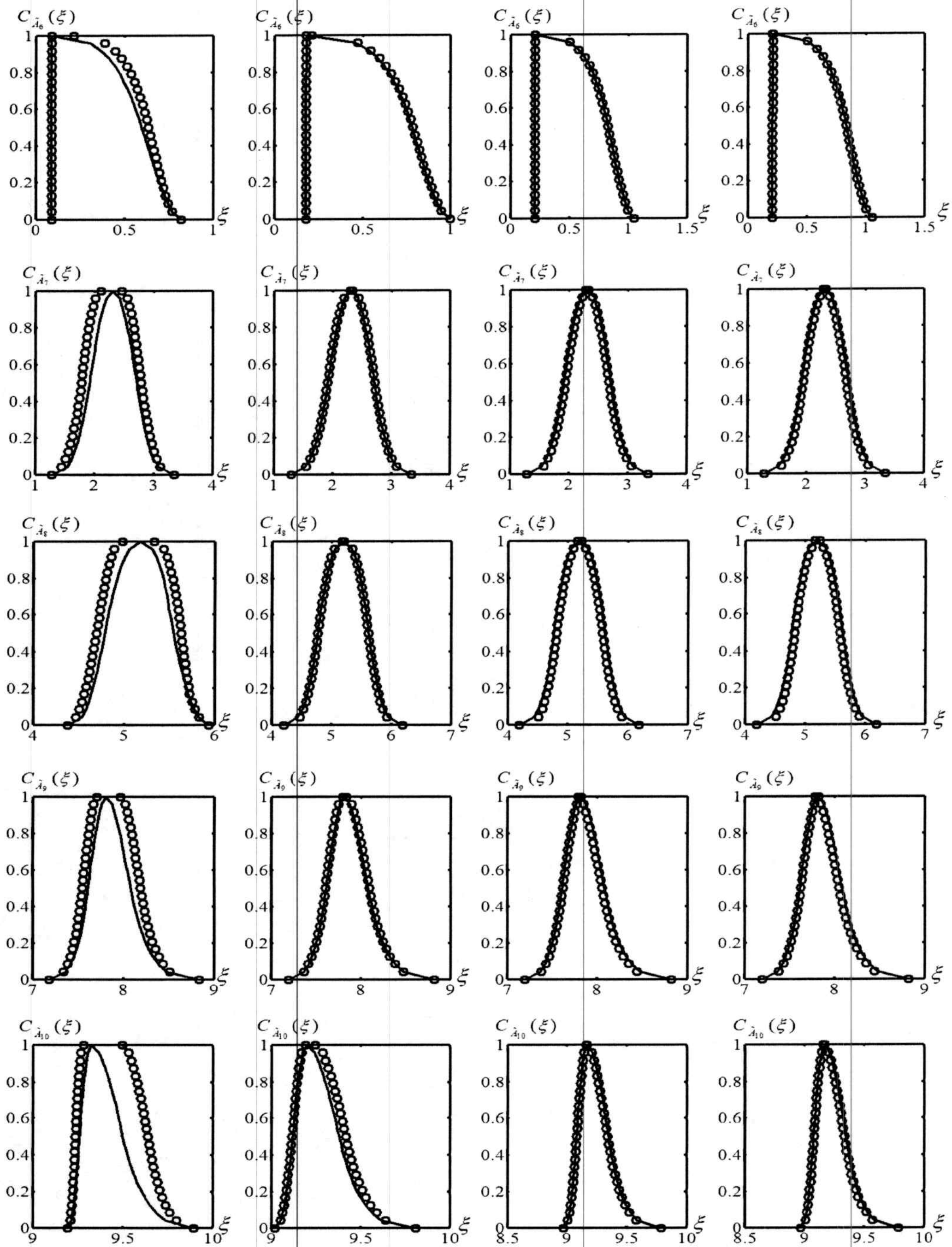


Figure 6.5 Centroids computed by KM (solid lines) and CF (circles) algorithms for $\tilde{A}_6,\ldots,\tilde{A}_{10}$ (each row is for an $\tilde{A}_i$) and for different primary variable sampling intervals of 0.5, 0.1, 0.01 and 0.001 (from left to right in each row).

where we chose $c(C_{\tilde{A},KM}(\xi|k))$ and $c(C_{\tilde{A},CF}(\xi|k))$ to be the center-of-gravity of $C_{\tilde{A},KM}(\xi|k)$ and $C_{\tilde{A},CF}(\xi|k)$, respectively. $d_{\tilde{A}}(k)$ was computed, for all 50 T2 FSs, and we observed that $d_{\tilde{A}}(k)$ generally decreased as k increased. In some cases, however, $d_{\tilde{A}}(k)$ increased for small values of k ($k < 10$), because, when too few α-planes are used, neither the results of the KM nor those of the CF algorithm have started to converge.

To illustrate these results, $d_{\tilde{A}}(k)$ is depicted in Fig. 6.8 for $\tilde{A}_1$-$\tilde{A}_{10}$, where a subjective threshold of 0.01 was applied in order to evaluate the performance of the CF algorithm with respect to k. Observe that $d_{\tilde{A}}(k)$ is always less than 0.01 when k >20 for all $\tilde{A}_1$-$\tilde{A}_{10}$. Observe, also, that $d_{\tilde{A}}(k)$ for T2 FSs with triangle secondary MFs ($\tilde{A}_1$-$\tilde{A}_5$) are always smaller than $d_{\tilde{A}}(k)$ for T2 FSs with bell-shaped secondary MFs ($\tilde{A}_6$-$\tilde{A}_{10}$). This is because triangle secondary MFs exactly satisfy the assumptions made in the derivations of the CF algorithm, namely that $g_x(u)$ and $h_x(u)$ are piecewise linear, whereas bell-shaped secondary MFs are not.

Comment: When the materials in this section was published as a journal paper, one of the anonymous reviewers pointed out that, although the above examples have focused on types of secondary *MF*s most often used by researchers to construct a T2 FS, it would be very interesting to see whether the CF algorithm can still produce accurate results when the secondary *MF*s for the T2 FS contain near-abrupt changes (see Fig. 6.9). In response to this, we constructed a T2 FS $\tilde{A}_{11}$ using such secondary *MF* and the word *High Amount*, and conducted the same kind of accuracy test for $\tilde{A}_{11}$ as above, the results of which are depicted in Fig. 6.10. Observe that these results are similar to that of $\tilde{A}_1$-$\tilde{A}_{10}$ depicted in Fig. 6.6 and 6.7, namely, $C_{\tilde{A}_{11},KM}(\xi)$ and $C_{\tilde{A}_{11},CF}(\xi)$ are very close even when only five α-planes are used (k=5) and they gradually converge as we increase the value of k. Similar results can be obtained if we use any of the other four *FOU*s to construct the T2 FS.

Although such special secondary *MF*s are rarely used in practicality, this test suggested by the reviewer did allow us to confirm the validity of the CF algorithm under such an extreme condition.■

6.5.3 Computation Time Tests

In this section, the computation times for the KM, EKM and CF algorithms are compared. It is difficult to develop general mathematical models that characterize the computation time of each algorithm, because: 1) When applying KM or EKM algorithms to an α-plane, the total number of iterations are unknown ahead of time, 2) The integration regions of the (continuous) KM and EKM algorithms may vary for different *FOU*s at each iteration; and, 3) It is obvious, from (6-17)-(6-20), (6-62), (6-63), (6-72) and (6-73), that the integration switch points, $c(\tilde{A}_\alpha)$ and $c(\tilde{A}_{\alpha+T_s})$, are also different for different *FOU*s, which causes the CF integration regions also to be different.

The actual computation times varied very slightly when the computer ran the KM, EKM and CF programs each time; so, 500 simulations were performed for each T2 FS, from which the average and standard deviation (STD) of the computation times of the KM, EKM and CF algorithms were obtained for all 50 T2 FSs for 34 values of k (k = 2, 3,..., 35). Because the results are similar for T2 FSs with the same kind of secondary MFs, only the average and STD of the computation times for $\tilde{A}_1$-$\tilde{A}_5$ and $\tilde{A}_6$-$\tilde{A}_{10}$ are depicted as functions of k in Figs. 6.11 and 6.12, respectively. Observe that, as expected, all the average

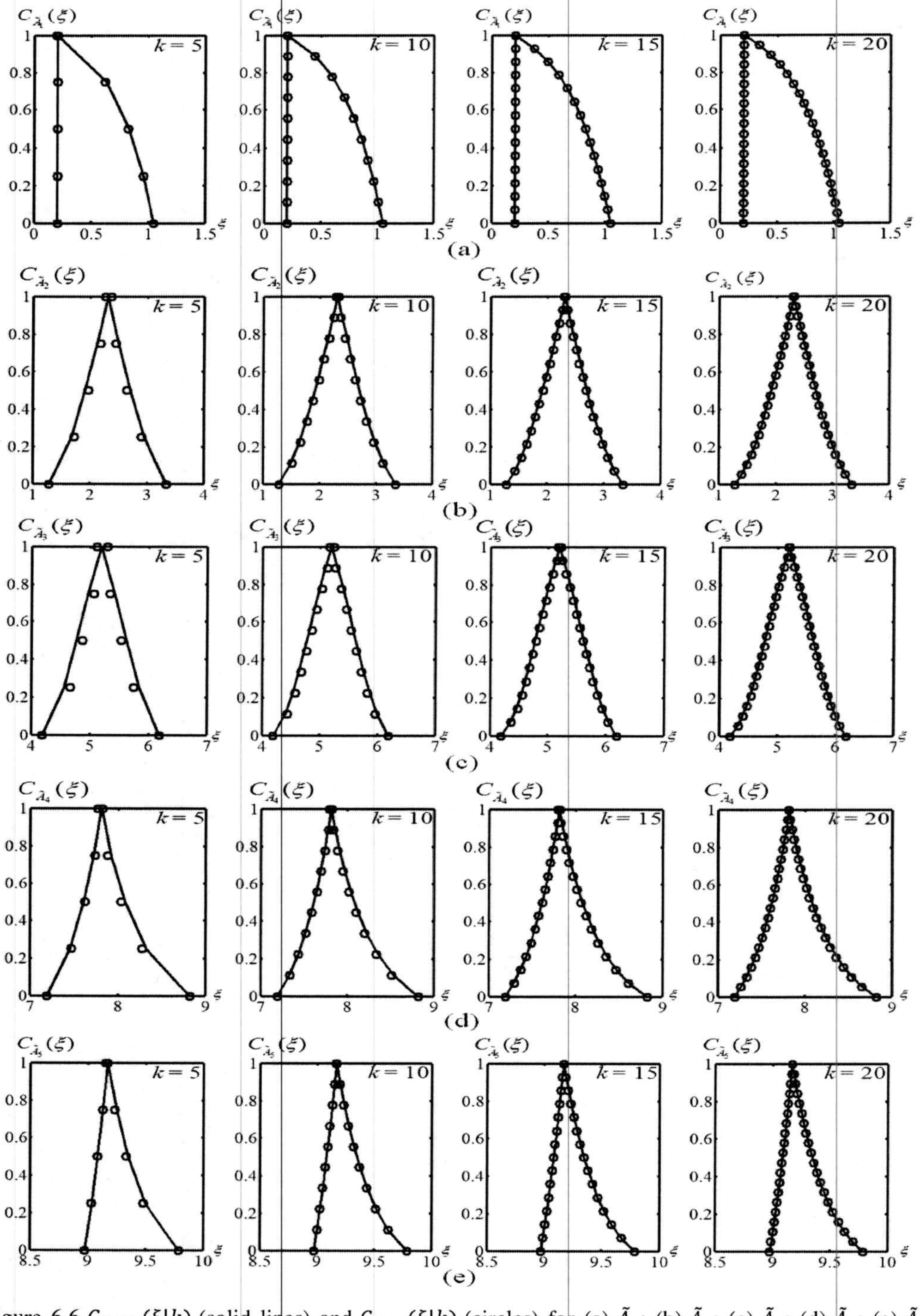


Figure 6.6 $C_{\tilde{A},KM}(\xi|k)$ (solid lines) and $C_{\tilde{A},CF}(\xi|k)$ (circles) for (a) $\tilde{A}_1$; (b) $\tilde{A}_2$; (c) $\tilde{A}_3$; (d) $\tilde{A}_4$; (e) $\tilde{A}_5$ using k = 5, 10, 15 and 20.

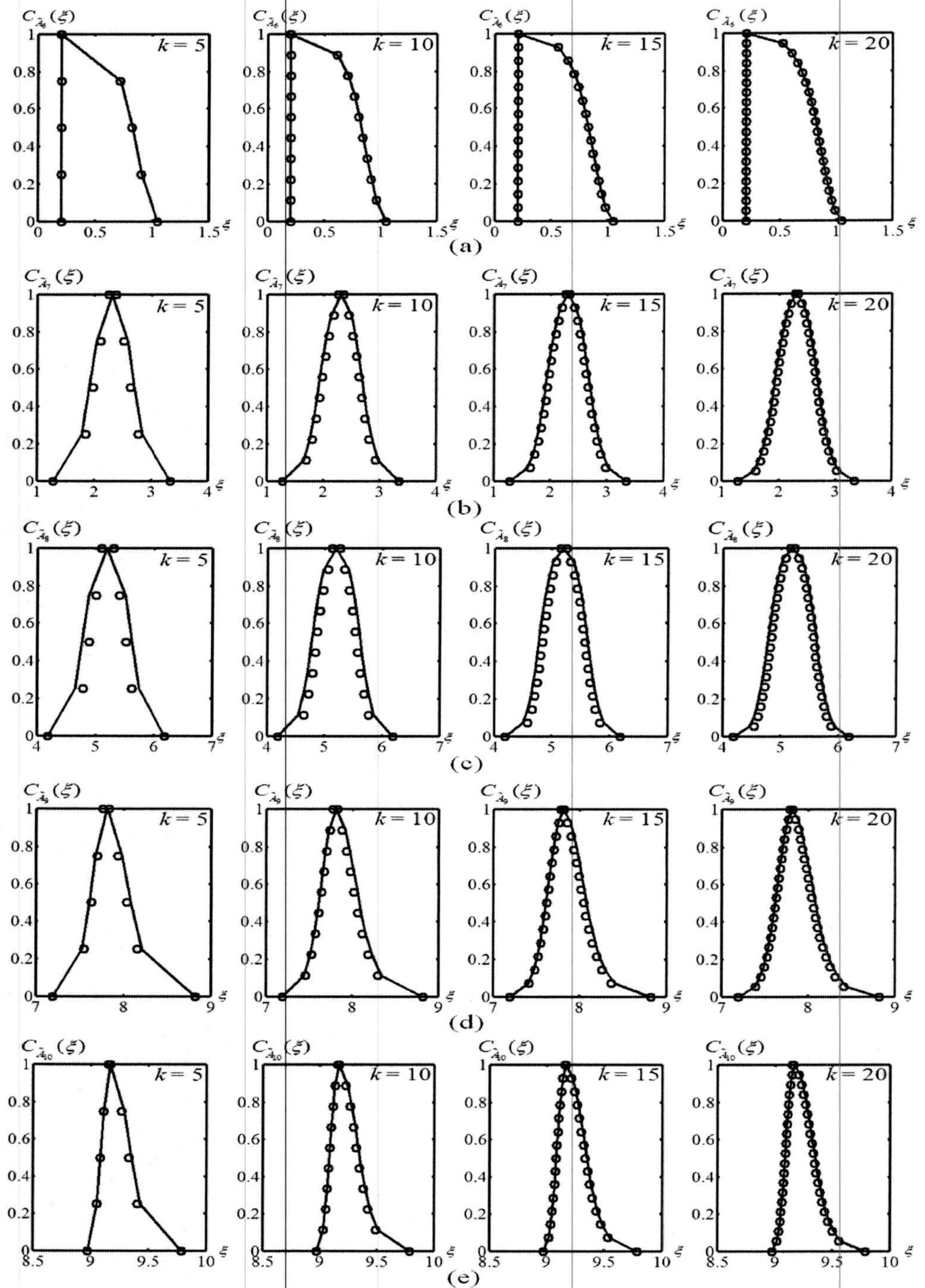


Figure 6.7 $C_{\tilde{A},KM}(\xi|k)$ (solid lines) and $C_{\tilde{A},CF}(\xi|k)$ (circles) for (a) $\tilde{A}_6$; (b) $\tilde{A}_7$; (c) $\tilde{A}_8$; (d) $\tilde{A}_9$; (e) $\tilde{A}_{10}$ using k = 5, 10, 15 and 20.

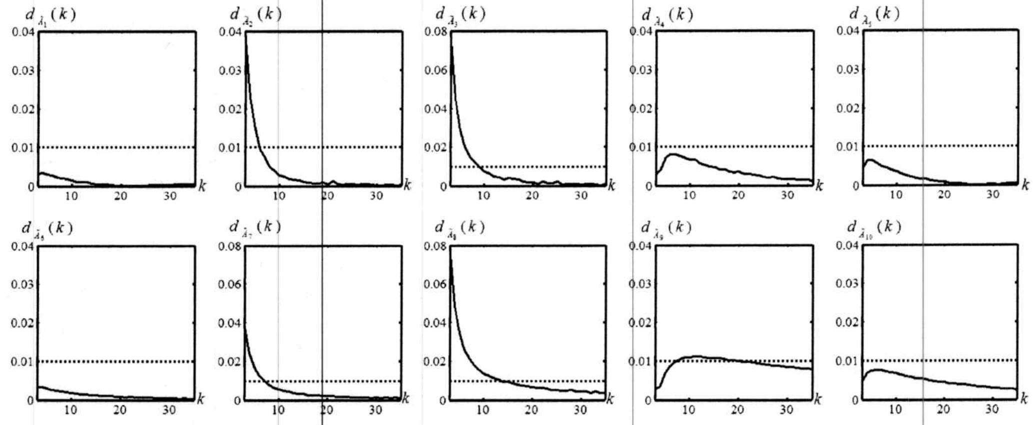


Figure 6.8 $d_{\tilde{A}}(k)$ (solid lines) for $\tilde{A}_1$-$\tilde{A}_{10}$ and the 0.01 threshold (dotted lines).

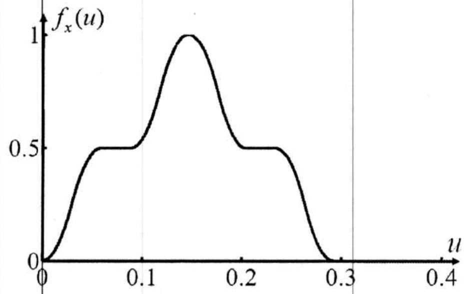


Figure 6.9 Special secondary *MF* that has near-abrupt changes for word *High Amount*, when x = 6.

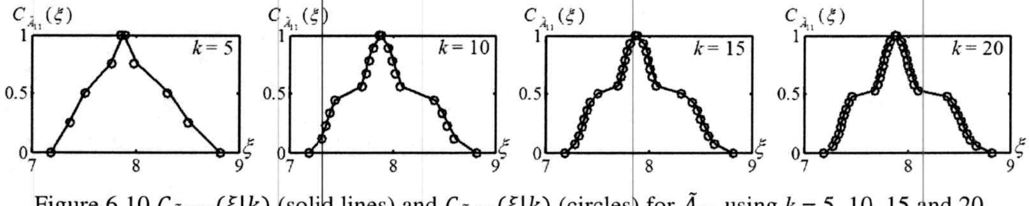


Figure 6.10 $C_{\tilde{A},KM}(\xi|k)$ (solid lines) and $C_{\tilde{A},CF}(\xi|k)$ (circles) for $\tilde{A}_{11}$ using k = 5, 10, 15 and 20.

computation times increase almost linearly as k increases for all three algorithms. Observe, also, that the STD of the computation times are much smaller as compared to the average computation times; hence, we only focus on the average computation times in the rest of the analyses. Finally, by comparing the top rows of Figs. 6.11 and 6.12, observe that the actual average computation times for KM and EKM algorithms are only marginally affected by the shapes of the *FOU*s, whereas the CF algorithm is faster for T2 FSs with triangle secondary MFs than those with bell-shaped secondary MFs.

Because of the observed linearity of $t_{CF}(k)$, a linear model can be used to approximate it for each T2 FS ($k \geq 2$), i.e.:

$$t_{CF}(k) \approx a \times k + b \tag{6-80}$$

where a and b were obtained by performing *Least-Squares* fitting to the actual $t_{CF}(k)$. Then we computed the average values of a and b for $\tilde{A}_1$-$\tilde{A}_5$ and $\tilde{A}_6$-$\tilde{A}_{10}$, respectively, and obtained the following general computation time model for the CF algorithm ($k \geq 2$):

$$\hat{t}_{CF}(k) \approx \begin{cases} 9.0369\times10^{-5}k + 5.28122\times10^{-4}, & \text{with triangle secondary MFs} \\ 20.538\times10^{-5}k + 4.4519\times10^{-4}, & \text{with bell-shaped secondary MFs} \end{cases} \tag{6-81}$$

Observe in Figs. 6.11 and 6.12, that the actual average computation times for the EKM and KM algorithms, $t_{KM}(k)$ and $t_{EKM}(k)$, are also almost linear; however, unlike the CF algorithm, there are no theoretical results for either KM or EKM algorithms that suggests different computation times for triangle and bell-shaped secondary MFs. No such results are observed in Figs. 6.11 and 6.12; hence, only one general linear computation time model is needed for each of the KM and EKM algorithms. The average values of a and b for $\tilde{A}_1$-$\tilde{A}_{10}$ were computed for both algorithms, leading to the following general computation time model, for the KM and EKM algorithms, respectively, ($k \geq 2$):

$$t_{EKM}(k) \approx 4.4586 \times 10^{-4}k - 0.3111 \times 10^{-4} \tag{6-82}$$

$$t_{KM}(k) \approx 8.7018 \times 10^{-4}k - 0.4036 \times 10^{-4} \qquad (6\text{-}83)$$

$t_{KM}(k)$, $t_{EKM}(k)$ and $t_{CF}(k)$ for T2 FSs with triangle and bell-shaped secondary MFs are depicted in Figs. 6.13(a) and 6.13(c), respectively.

In order to quantify computation time improvements, the percentage computation time reductions of the CF algorithm over the KM and EKM algorithms are defined, as:

$$r_{KM}(k) = \frac{t_{KM}(k) - t_{CF}(k)}{t_{KM}(k)} \times 100 \qquad (6\text{-}84)$$

$$r_{EKM}(k) = \frac{t_{EKM}(k) - t_{CF}(k)}{t_{EKM}(k)} \times 100 \qquad (6\text{-}85)$$

$r_{KM}(k)$ and $r_{EKM}(k)$ are depicted in Fig. 14 for $\tilde{A}_1$-$\tilde{A}_{10}$. They can also be approximated by substituting (6-81)-(6-83) into (6-84) and (6-85), and these results are depicted in Figs. 6.13(b) and 6.11(d). Observe that the results in these latter figures closely resemble the actual percentage time reductions for all the T2 FSs shown in Fig. 6.14.

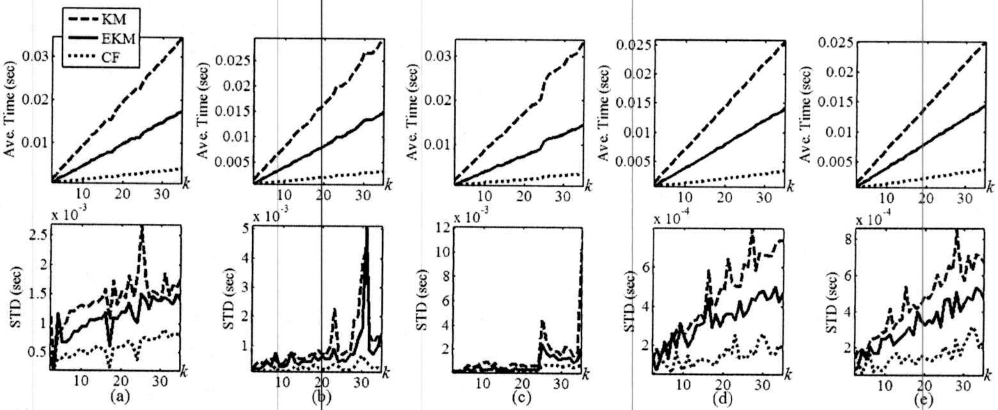


Figure 6.11 Average and STD of the computation times for KM (dashed), EKM (solid) and CF (dotted) algorithms for (a) $\tilde{A}_1$; (b) $\tilde{A}_2$; (c) $\tilde{A}_3$; (d) $\tilde{A}_4$; (e) $\tilde{A}_5$.

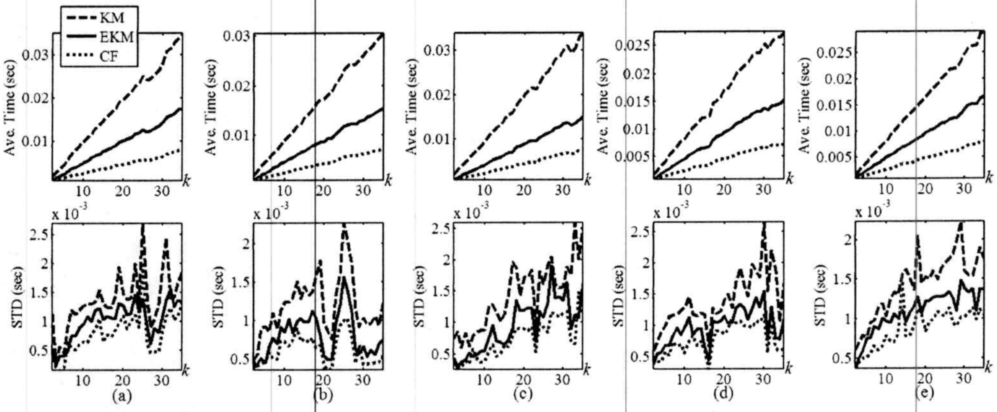


Figure 6.12 Average and STD of the computation time for KM (dashed), EKM (solid) and CF (dotted) algorithms for (a) $\tilde{A}_6$; (b) $\tilde{A}_7$; (c) $\tilde{A}_8$; (d) $\tilde{A}_9$; (e) $\tilde{A}_{10}$.

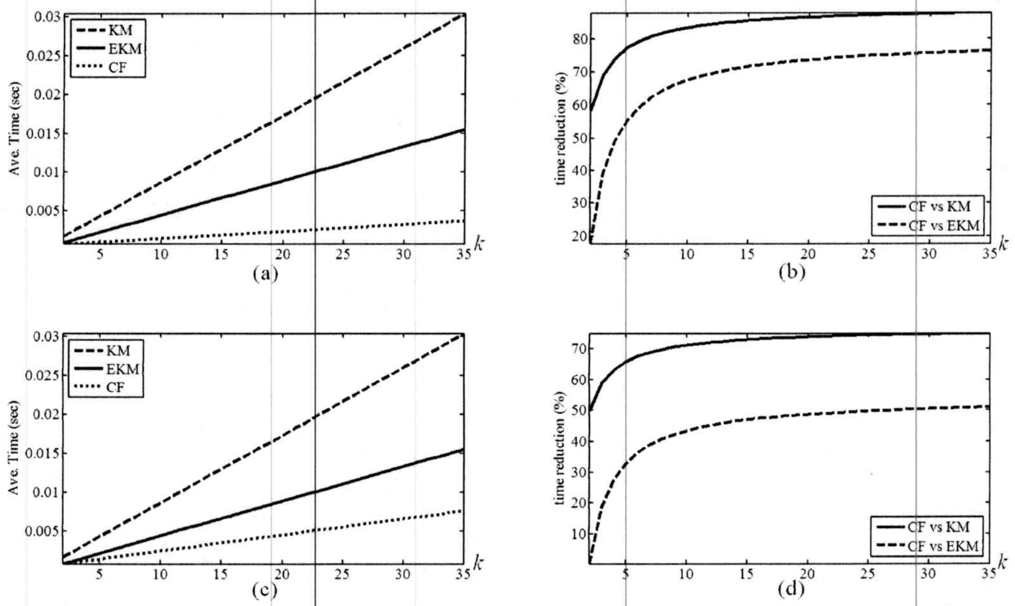


Figure 6.13 $\hat{t}_{KM}(k)$, $\hat{t}_{EKM}(k)$ and $\hat{t}_{CF}(k)$ for T2 FSs with (a) triangle secondary MFs and (c) bell-shaped secondary MFs; and, $\hat{r}_{KM}(k)$ and $\hat{r}_{EKM}(k)$ for T2 FSs with (b) triangle secondary MFs, and (d) bell-shaped secondary MFs

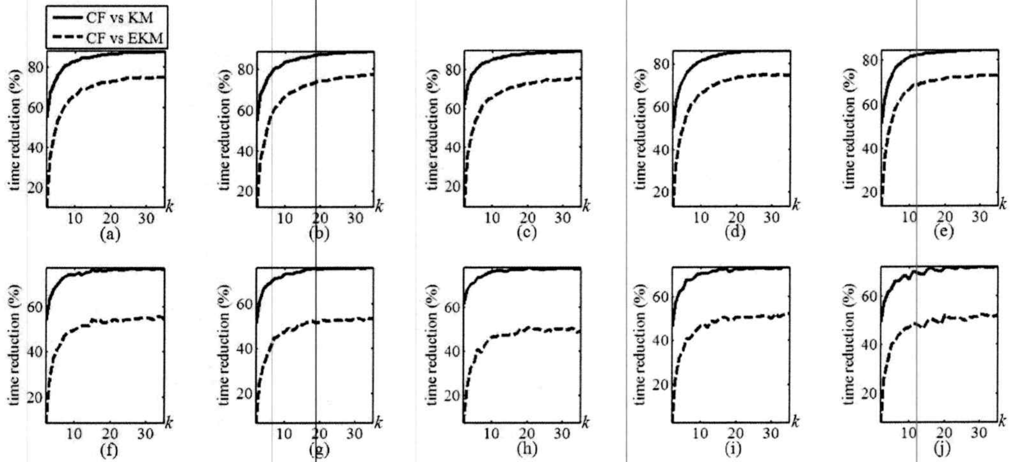


Figure 6.14 The actual percentage computation time reduction of the CF algorithm over the KM algorithms (solid lines) and the EKM algorithms (dashed lines) for (a) $\tilde{A}_1$, (b) $\tilde{A}_2$, (c) $\tilde{A}_3$, (d) $\tilde{A}_4$, (e) $\tilde{A}_5$, (f) $\tilde{A}_6$, (g) $\tilde{A}_7$, (h) $\tilde{A}_8$, (i) $\tilde{A}_9$, (j) $\tilde{A}_{10}$.

In Fig. 6.14, when $k \geq 20$ and for T2 FSs with triangle secondary MFs, the CF algorithm reduces the computation time by over 80% compared to KM, and by about 75% for EKM; and, for T2 FSs with bell-shaped secondary MFs, the CF algorithm reduces the computation time by over 75% compared to KM, and by about 50% for EKM algorithms.

6.6 Summary

This section used two important properties of continuous KM algorithms to develop a new way to compute the centroid of a general T2 FS leading to a new Centroid Flow (CF) algorithm. The CF algorithm avoids the iterative features of KM (EKM) algorithms by utilizing the structural knowledge of the secondary MFs. This section has shown theoretically that, in terms of computation time, the CF algorithm is guaranteed to outperform KM and EKM algorithms for T2 FSs with triangle and trapezoidal secondary

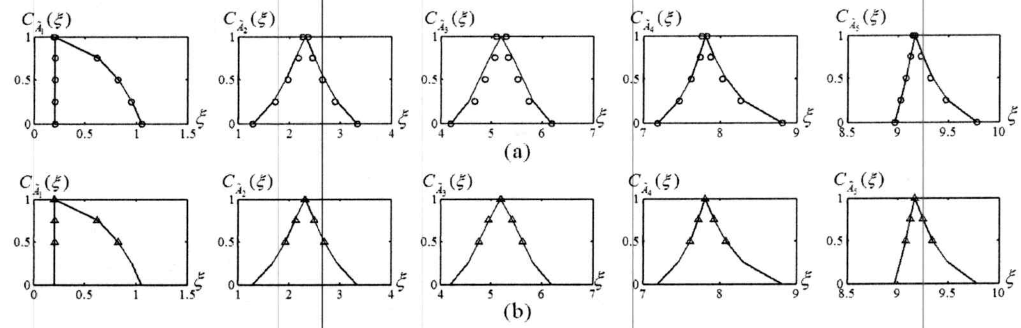


Figure 6.15 $C_{\tilde{A}}(\xi|k=5)$ of $\tilde{A}_1$ - $\tilde{A}_5$ computed by (a) the KM algorithm (solid lines) and the CF algorithm initialized at $\alpha = 0$ α-plane (circles), and (b) the KM algorithms (solid lines) and the CF algorithm initialized at $\alpha = 0.5$ α-plane (triangles).

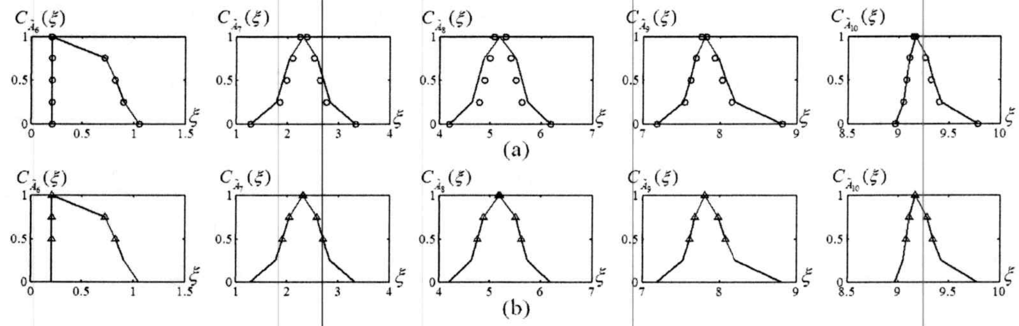


Figure 6.16 $C_{\tilde{A}}(\xi|k=5)$ of $\tilde{A}_6$- $\tilde{A}_{10}$ computed by (a) the KM algorithm (solid lines) and the CF algorithm initialized at $\alpha = 0$ α-plane (circles), and by (b) the KM algorithm (solid lines) and the CF algorithm initialized at $\alpha = 0.5$ α-plane (triangles).

MFs, and will outperform KM and EKM algorithms if they don't converge in less than two iterations for T2 FSs with other kinds of secondary MFs.

Exhaustive experiments were performed for 50 T2 FSs that have different *FOU*s and secondary MFs. The results of the accuracy tests showed that, when α is discretized finely enough, $C_{\tilde{A},CF}(\xi)$ will converge when the primary-variable sampling-interval is less than 0.01. The results also showed that, for primary-variable sampling-interval of 0.01, $c(C_{\tilde{A},CF}(\xi))$ and $c(C_{\tilde{A},KM}(\xi))$ have an absolute difference less than 0.01, when more than 20 α-planes are used.

Computation time tests showed that, when more than 20 α-planes are used: 1) for T2 FSs with triangle secondary MFs, the CF algorithm reduces the computation time by over 80% compared to KM algorithms, and by about 75% for EKM algorithms; and, 2) for T2 FSs with bell-shaped secondary MFs, the CF algorithm reduces the computation time by over 75% compared to KM algorithms, and by about 50% for EKM algorithms. These results represent significant improvements in computation time for the new CF algorithm over the existing KM and EKM algorithms.

In actual applications, the accuracy of the CF algorithms can possibly be enhanced by initializing the algorithm at the $\alpha = 0.5$ α-plane, and have it "flow" both upwards and downwards. Doing so would, in theory, reduce the accumulative error by half as compared to starting from the $\alpha = 0$ α-plane and letting the error accumulate all the way up to the $\alpha = 1$ α-plane (see Figs. 6.15 and 6.16).

In addition to computing the centroid of a general T2 FS, the CF algorithms can be applied in many places where the KM (EKM) algorithms have been used, e.g., computing the variance and skewness of a general T2 FS, performing Center of Gravity type-reduction in a general T2 FLS.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

Chapter 2 of this dissertation proposed a novel data interpolation algorithm called IEKFS that allows a dynamic Extended Kalman Filter (EKF) estimation system, which was originally designed to accurately estimate interwell connectivity only when measurements can be made every day, to obtain equivalently good performances even when a large portion of data is missing, e.g., only one measurement available every eight days. To be self-contained, this chapter provides a comprehensive background review of this particular problem, including the parametric models used to characterize the subsurface reservoir, how different SVM can be established based on this model, practical strategy for applying the EKF technique on a large-scale oilfield, and how to dynamically impose physical constraints on the estimates as they are obtained. The IEKFS algorithm can be viewed as an technique that enhances the robustness of the system in a highly uncertain environment from the data perspective.

In Chapters 3 and 4, two different types of advanced FLSs were developed, namely, a Non-Singleton IT2 FLS and a Mixture FLS, the former for a Universal Image Noise Removal Filter and the latter for a Forest-Fire-Size Predictor. These two types of FLSs are both developed based on the same QPSO designing paradigm. This kind of design framework allows us to easily implement complex FLSs that are previously understudied due to technical difficulties. Based on this framework, readers can easily modify and extend the above FLSs to fit a variety of research problems for data sets that have different features.

In Chapter 5, we focused on general T2 FSs, and provided a unified theorem for computing different uncertainty measures for T2 FSs based on a recently developed α-plane RT, extensive simulation results that demonstrate the applicability of this theorem, and how to interpret the resulting uncertainty measures, which themselves are T1 FSs. Then, based on this unified theorem, Chapter 6 developed a much more efficient CF algorithm for computing the centroid of a T2 FS.

7.2 Future Work

This dissertation has provided two separate paths for improving the performance of learning systems, one from a data perspective and one from a system design perspective. One can easily image the connection between the two paths as follows: given an application that is subject to frequently missing measurements and a great deal of uncertainties, one can first use the IEKFS algorithm to create high quality virtual measurements, and then use the robust FLSs for learning purposes. However, to use the IEKFS algorithm, one has to have a parametric model that describes the system. On the other hand, FLSs are usually model-free. So a question that naturally rises is: If there exists a parametric model for a system, does a FLS still have the value demonstrated in this dissertation? To draw a direct connection between the two paths, one future topic of research is to look for an interesting problem that allows the unified employment of the IEKFS algorithm and FLSs, and demonstrate the superiority of such an integrated system compared to other techniques.

Regarding the details of the interwell connectivity estimation problem, the CEKF and IEKFS haven't been extensively tested on real field data yet; they have been separately tested on hypothetical and reservoir simulator data. To apply the CEKF to real data, a strategy that enables us to eliminate non-contributing injectors and select an optimal initial injector set is needed. This can be quite complex, because the CEKF operates from an injector-centric perspective; therefore, once an injector is eliminated from one producer-centric model, it affects all other producer-centric models that include this injector. We also need a strategy to select optimal contributing injector set for a producer-centric model during the IEKFS processing, before putting this technique into practical use. The RMSEs of the interpolated production rates can be used as our selection criterion. Additionally, to jointly apply IEKFS and CEKF in a real field, we will need a unified framework, under which both interpolation and estimation can be correctly carried out. Because, the CEKF groups multiple producer-centric models together from an injector-centric perspective, the IEKFS is still based on a single producer-centric model. It is possible that the contributing injector set for a producer during IEKFS processing differs from the contributing injector set selected by the same producer during CEKF processing. Whether this will be problematic or not also awaits study.

On the FLS side, an increasingly popular topic is to extend current IT2 FLSs to general T2 FLSs, which are supposedly more robust, as when IT2 FLSs are compared to T1 FLSs. Chapters 5 and 6 can serve as foundations for such studies. The practical values of these materials in the actual design or implementations of general T2 FLSs remain to be shown.

Bibliography

[1] M. Abbaszadeh-Dehghani and W. E. Brigham, "Analysis of Well-to-Well Tracer Flow to Determine Reservoir Heterogeneity," in *Annual California Regional Meeting*, San Francisco, 1982, pp. 24-26.

[2] M. Abbaszadeh-Dehghani and W. E. Brigham, "Analysis of Well-to-Well Tracer Flow to Determine Reservoir Layering," *Journal of Petroleum Technology*, vol. 36, no. 10, pp. 1753-1762, October 1984.

[3] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 1012-1025, June 1996.

[4] I. Aizenberg, C. Butakoff, and D. Paily, "Impulsive noise removal using threshold Boolean filtering based on the impulse detecting functions," *IEEE Signal Processing Letter*, vol. 17, no. 4, pp. 513-520, 2005.

[5] S. Akin, "Analysis of tracer tests with simple spreadsheet models," *Computers & Geosciences*, vol. 27, no. 2, pp. 171-178, 2001.

[6] A. Albertoni and L. W. Lake, "Inferring Interwell Connectivity only from Well-Rate Fluctuations in Waterfloods," *SPE Reservoir Evaluation and Engineering*, vol. 6, no. 1, pp. 6-16, February 2003.

[7] A. N. Araqye-Martinez, "Estimation of autocorrelation and its use in sweep efficiency calculation," University of Texas at Austin, Austin, Master Thesis 1993.

[8] L. Astudillo, O. Castillo, and L. T. Aguilar, "Intelligent control for a perturbed autonomous wheeled mobile robot: a type-2 fuzzy logic approach," *Journal of Nonlinear Studies*, vol. 14, no. 3, pp. 37-48, 2007.

[9] S. Auephanwiriyakul, A. Adrian, and J. M. Keller, "Type-2 fuzzy set analysis in management surveys," in *Proceedings of the FUZZ-IEEE*, Honolulu, HI, 2002, pp. 1321-1325.

[10] P. Baguley, T. Page, V. Koliza, and P. Maropoulos, "Time to market prediction using type-2 fuzzy sets," *Journal of Manufacturing Technology Management*, vol. 17, no. 4, pp. 513-520, 2006.

[11] F. Van den Bergh, "An analysis of particle swarm optimizer," University of Pretoria, Pretoria, South Africa, PhD thesis 2001.

[12] R. A. Berk, *Statistical Learning from a Regression Perspective*, 1st ed. New York City, New York, USA: Springer, 2008.

[13] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, Massachusetts, USA: Athena Scientific, 1999.

[14] E. Besdok, P. Çivicio glu, and M. Alçi, "Impulsive noise suppression from highly corrupted images by using resilient neural networks," *Lecture Notes in Computer Science*, vol. 3070, pp. 670-675, 2004.

[15] E. Besdok and M. E. Yüksel, "Impulsive noise rejection from images with jarque-berra test based median filter," *International Journal of Electronics and Communications*, vol. 59, no. 2, pp. 105-109, 2005.

[16] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. Cary, North Carolina, USA: Oxford University Press, 1996.

[17] P. P. Bonissone, "A fuzzy sets based linguistic approach: theory and applications," in *Proceedings of the 12th Winter Simulation Conference*, Orlando, FL, 1980, pp. 99-111.

[18] P. P. Bonissone, "A pattern recognition approach to the problem of linguistic approximation," in *Proceedings of the IEEE International Conference on Cybernetics and Society*, Denver, CO, 1979, pp. 793-798.

[19] A. Bouchachia and R. Mittermeir, "A neural cascade architecture for document retrieval," in *Proceedings of the International Joint Conference Neural Networks, vol. 3*, Poland, OR, 2003, pp. 1915-1920.

[20] W. E. Brigham and M. Abbaszadeh-Dehghani, "Tracer Testing for Reservoir Description," *Journal of Petroleum Technology*, vol. 39, no. 5, pp. 519-527, 1987.

[21] H. Bustince, "Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets," *International Journal of Approximate Reasoning*, vol. 23, no. 3, pp. 137-209, Mar. 2000.

[22] H. Bustince, E. Barrenechea, and M. Pagola, "Generation of interval-valued fuzzy and Atanassov's intuitionistic fuzzy connectives from fuzzy connectives and from K-alpha operators: Laws for conjunctions and disjunctions, amplitude," *International Journal of Intelligent Systems*, vol. 23, no. 6, pp. 680-714, Jun. 2008.

[23] H. Bustince, V. Mohedano, E. Barrenechea, and M. Pagola, "Definition and construction of fuzzy DI-subsethood measures," *Information Sciences*, vol. 176, no. 21, pp. 3190-3231, Nov. 2006.

[24] M. Carlson, *Practical reservoir simulation: using, assessing, and developing results.* Tulsa, Oklahoma, USA: PennWell Corporation, 2003.

[25] O. Castillo, N. Cazarez, and P. Melin, "Design of stable type-2 fuzzy logic controllers based on a fuzzy Lyapunov approach," in *Proceedings of the FUZZ-IEEE*, Vancouver, Canada, 2006, pp. 2331-2336.

[26] O. Castillo and P. Melin, "Adaptive noise cancellation using type-2 fuzzy logic and neural networks," in *Proceedings of the FUZZ-IEEE, vol. 2*, 2004, pp. 1093-1098.

[27] O. Castillo and P. Melin, "Evolutionary computing for optimizing type-2 fuzzy logic systems in intelligent control of non-linear dynamic plants," in *Proceedings of the North American Fuzzy Information Processing Society (NAFIPS)*, Ann Arbor, MI, 2005, pp. 247-251.

[28] O Castillo and P Melin, *Type-2 Fuzzy Logic: Theory and Applications*, J Kacprzyk, Ed. Berlin Heidelberg: Springer-Verlag, 2008.

[29] J. Y. Chang and J. L. Chen, "Classifier augmented median filter for image restoration," *IEEE Trans. on Instrumentation and Measurement,* vol. 53, no. 2, pp. 351-356, 2004.

[30] R. H. Chan, C. Hu, and M. Nikolova, "An iterative procedure for removing random-valued impulse noise," *IEEE Signal Processing Letter*, vol. 11, no. 12, pp. 921-924, 2004.

[31] S. Chatterjee and A. S. Hadi, *Regression Analysis by Example*, 4th ed. Hoboken, New Jersey, USA: Wiley-Interscience, 2006.

[32] Q. Chen and S. Kawase, "On fuzzy-valued fuzzy reasoning," *Fuzzy Sets and Systems*, vol. 113, pp. 237-251, 2000.

[33] T. Chen, K. K. Ma, and L. H. Chen, "Tri-state median filter for image denoising," *IEEE Trans. on Image Processing*, vol. 8, no. 12, pp. 1834-1838, 1999.

[34] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letter*, vol. 8, no. 1, pp. 1-3, 2001.

[35] T. Chen and H. R. Wu, "Space variant median filters for the restoration of impulse noise corrupted images," *IEEE Trans. on Circuits and Systems*, vol. 48, no. 8, pp. 784-789, 2001.

[36] Y. S. Choi and R. Krishnapuram, "A robust approach to image enhancement based on fuzzy logic," *IEEE Transactions on Image Processing*, vol. 6, no. 6, pp. 808-825, June 1997.

[37] R. Christensen, "Entropy minimax multivariate statistical modeling I: theory," *International Journal of General Systems*, vol. 11, pp. 231-277, 1985.

[38] R. Christensen, "Entropy minimax multivariate statistical modeling II: applications," *International Journal of General Systems*, vol. 12, no. 3, pp. 227-305, 1985.

[39] L. dos Santos Coelho and B. M. Herrera, "Fuzzy identification based on a chaotic particle swarm optimization approach applied to a nonlinear yo-yo motion system," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 6, pp. 3234-3245, 2007.

[40] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Sets and Systems*, vol. 14, no. 1, pp. 5-31, 2003.

[41] P. Cortez and A. Morais, "A data mining approach to predict forest fires using meteorological data," in *Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence*, Guimaraes, Portugal, 2007, pp. 512–523.

[42] S. Coupland, "Type-2 fuzzy sets: geometric defuzzification and type-reduction," in *Proc. of IEEE Symposium on Foundations of Computational Intelligence*, Honolulu, 2007, pp. 622-629.

[43] S. Coupland and R. I. John, "A new and efficient method for the type-2 meet operation," in *Proc. of FUZZ-IEEE*, Budapest, 2004, pp. 959-964.

[44] S. Coupland and R. I. John, "Geometric type-1 and type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol. 15, no. 1, pp. 3-15, 2007.

[45] S. Coupland and R. I. John, "Towards more efficient type-2 fuzzy logic systems," in *Proc. FUZZ-IEEE*, Reno, 2005, pp. 236-241.

[46] T. M. Cover, "The best two independent measurements are not the two best," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 4, pp. 116-117, 1974.

[47] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, New Jersey, USA: Wiley-Interscience, 2006.

[48] V. Crnojevic, V. Senk, and Z. Trpovski, "Advanced impulse detection based on pixel-wise MAD," *IEEE Signal Processing Letter*, vol. 11, no. 7, pp. 589-592, 2004.

[49] V. V. Cross and T. A. Sudkamp, *Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications*. Heidelberge, NY: Physica-Verlag, 2002.

[50] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. on Image Processing*, vol. 16, no. 8, pp. 2080-2095, August 2007.

[51] C. Ding and H. C. Peng, "Minimum redundancy feature selection from microarray gene expression data," in *Proc. of Second IEEE Computational Systems Bioinformatics Conf.*, 2003, pp. 523-528.

[52] D. Dubois and H. Prade, "Fuzzy cardinality and the modeling of imprecise quantification," *Fuzzy Sets and Systems*, vol. 16, pp. 199-230, 1985.

[53] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," *IEEE Trans. on Image Processing*, vol. 10, no. 2, pp. 242-251, February 2001.

[54] J. R. Fanchi, *Principles of applied reservoir simulation*. Burlington, Massachusetts, USA: Elsevier, 2005.

[55] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo, and S. Rojas, "A type-2 fuzzy controller for tracking mobile objects in the context of robotic soccor games," in *Proceedings of the FUZZ-IEEE*, Reno, NV, 2005, pp. 359-364.

[56] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Hoboken, New Jersey, USA: Wiley, 2000.

[57] L. J. Fogel, "Evolutionary programming in perspective: the top-down view," in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II, and C. J. Robinson, Eds. Piscadaway, New Jersey, USA: IEEE Press, 1994.

[58] A. Fowler, A. M. Teredesai, and M. De Cock, "An evolved fuzzy logic system for fire size prediction," in *Proc. of The 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS 2009)*, Cincinnati, OH, USA, 2009, pp. 1-6.

[59] M. Gabbouj, E. J. Coyle, and N. C. Gallager, "An overview of median and stack filtering," *Circuits, Systems, and Signal Processing*, vol. 11, no. 1, pp. 7-45, 1992.

[60] R. G. Gallager, *Information Theory and Reliable Communication*, 1st ed. Hoboken, New Jersey, USA: John Wiley & Sons, 1968.

[61] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. on Image Processing*, vol. 14, no. 11, pp. 105-109, 2005.

[62] J. F. Geer and G. J. Klir, "A mathematical analysis of information preserving transformations between probabilitic and possibilistic formulations of uncertainty," *International Journal of General Systems*, vol. 20, no. 2, pp. 143-176, 1992.

[63] D. E. Goldberg, *Genetic Algorithms in search Optimization and Machine Learning*. Reading, Massachusetts, USA: Addison-Wesley, 1989.

[64] R. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2007.

[65] S. Greenfield and R. I. John, "Optimized generalized type-2 join and meet," in *Proceedings of the FUZZ-IEEE*, London, UK, 2007, pp. 141-146.

[66] S. Greenfield, R. I. John, and S. Coupland, "A novel sampling method for type-2 defuzzification," in *Proc. of UKCI 2005*, London, 2005, pp. 120-127.

[67] L. Gu and Y. Q. Zhang, "Web shopping expert using new interval type-2 fuzzy reasoning," *Soft Computing*, vol. 11, no. 8, pp. 741-751, 2007.

[68] H. Hagras, "Type-2 FLCs: a new generation of fuzzy controllers," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 30-43, 2007.

[69] L. H. Hamel, *Knowledge Discovery with Support Vector Machines*. Hoboken, New Jersey, USA: Wiley-Interscience, 2009.

[70] H. Hamrawi and S. Coupland, "Type-2 fuzzy arithmetic using alpha-planes," in *Proc. IFSA/EUSFLAT*, Lisbon, Portugal, 2009, pp. 606-611.

[71] W. Y. Han and J. C. Lin, "Minimum-maximum exclusive mean filter to remove impulse noise from highly corrupted images," *Electronic Letters*, vol. 33, no. 2, pp. 124-125, 1997.

[72] D. Harmanec, "Measures of uncertainty and information," Society for Imprecise Probability Theory and Applications: http://www.sipta.org/documentation/summary_measures/main.html, 1999.

[73] S. Haykin, *Kalman Filtering and Neural Networks*. New Jersey: John Wiley & Sons, 2001.

[74] M. Higashi and G. Klir, "Measures of Uncertainty and Information Based on Possibility Distributions," in *Fuzzy Sets for Intelligent Systems, Morgan Kaufmann Publishers*, San Mateo, CA, 1993, pp. 217-232.

[75] F. Hoffmann, "Evolutionary algorithms for fuzzy control system design," in *Proceedings of the IEEE*, 2001, pp. 1318–1333.

[76] W. Hsu, M. Lee, and J. Zhang, "Image mining: Trends and developments," *Journal of Intelligent Information Systems*, vol. 19, no. 1, pp. 7-23, 2002.

[77] F. J. Iannarilli and P. A. Rubin, "Feature selection for multiclass discrimination via mixed-integer linear programming," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 779-783, 2003.

[78] J. Jaeger, R. Sengupta, and W. L. Ruzzo, "Improved gene selection for classification of microarrays," in *Proc. of Pacific Symp. Biocomputing*, 2003, pp. 53-64.

[79] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, 2000.

[80] R. I. John and C. Czarnecki, "An adaptive type-2 fuzzy system for learning linguistic membership grades," in *Proc. of FUZZ-IEEE*, Seoul, 1999, pp. 1552-1556.

[81] C. F. Juang, "A tsk-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 155-170, 2001.

[82] C. F. Juang, C. M. Hsiao, and C. H. Hsu, "Hierarchical cluster-based multispecies particle-swarm optimization for fuzzy-system optimization," *IEEE Trans. on Fuzzy Systems*, vol. 18, no. 1, pp. 14-26, 2010.

[83] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Upper Saddle River, New Jersey, USA: Pearson Prentice Hall, 2008.

[84] K.J. Heffer, R.J. Fox, C.A. McGill, BP Exploration Ltd. and N.C. Koutsabeloulis, Vector International Processing Systems Ltd., "Novel Techniques Show Links between Reservoir Flow Directionality, Earth Stress, Fault Structure and Geomechanical Changes in Mature Waterfloods," *SPE Journal*, vol. 2, no. 2, pp. 91-98, June 1997, SPE30711.

[85] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, pp. 195-220, 2001.

[86] N. N. Karnik and J. M. Mendel, "Introduction to type-2 fuzzy logic systems," in *Proc. of FUZZ IEEE*, Anchorage, 1998, pp. 915-920.

[87] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 Fuzzy Logic Systems," *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 6, pp. 643-658, 1999.

[88] J. Kennedy and R. Eberhart, "Paticle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks,* Perth, WA, Australia, 1995, pp. 1942-1948.

[89] C. Kenney, Y. Deng, B. S. Manjunath, and G. Hewer, "Peer group image enhancement," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 326-334, February 2001.

[90] V. V. Khryashchev, I. V. Apalkov, A. L. Priorov, and P. S. Zovanarev, "Image denoising using adaptive switching median filter," in *IEEE Internation Conference on Image Processing*, Genoa, Italy, 2005, pp. 117-120.

[91] G. J. Klir, "A principle of uncertainty and information invariance," *International Journal of General Systems*, vol. 17, no. 2-3, pp. 249-275, 1990.

[92] G. J. Klir, "Principles of uncertainty: what are they? why do we need them?," *Fuzzy Sets and Systems*, vol. 74, pp. 15-31, 1995.

[93] G. J. Klir and B. Parviz, "Probability - possibility transformations: a comparison," *International Journal of General Systems*, vol. 21, no. 3, pp. 291-310, 1992.

[94] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Application*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[95] R. Kohavi and G. John, "Wrapper for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.

[96] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhacement," *IEEE Trans. on Circuits and Systems*, vol. 38, no. 9, pp. 984-993, 1991.

[97] R. A. Krohling and L. dos Santos Coelho, "Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, pp. 1407-1416, 2006.

[98] N. Kwak and C. H. Choi, "Input feature selection by mutual information based on parzen window," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1667-1671, 2002.

[99] C.-S. Lee, Y.-H. Kuo, and P.-T. Yu, "Weighted fuzzy mean filter for image processing," *Fuzzy Sets and Systems*, vol. 89, no. 2, pp. 157-180, 1999.

[100] E. Lee and R. Li, "Comparison based on fuzzy numbers based on probability measures of fuzzy events," *Computer & Mathematics Applications*, vol. 15, pp. 887-896, 1988.

[101] C.-H. Lee and Y.-C. Lin, "Control of nonlinear uncertain systems using type-2 fuzzy neural network and adaptice filter," in *Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, vol. 2*, Taipei, Taiwan, 2004, pp. 1177-1182.

[102] K.H. Lee, A. Ortega, A.M. Nejad, N. Jafroodi, and I. Ershaghi, "Mapping Fractures and High Permeability Channels in Waterfloods by Use of Injection and Production Rates," *Journal of Petroleum Technology*, vol. 62, no. 1, pp. 44-45, 2010.

[103] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. on Fuzzy Systems*, vol. 8, pp. 535-550, 2000.

[104] Q. Liang and L. Wang, "Sensed signal strength forecasting for wireless sensors using interval type-2 fuzzy logic systems," in *Proceedings of the FUZZ-IEEE*, Reno, NV, 2005, pp. 25-30.

[105] P.-Z. Lin, C.-F. Hsu, and T.-T. Lee, "Type-2 fuzzy logic controller design for buck DC-DC converters," in *Proceedings of the FUZZ-IEEE*, Reno, NV, 2005, pp. 365-370.

[106] F. Liu, "An efficient centroid type reduction strategy for general type-2 fuzzy logic system," *Information Sciences*, vol. 178, pp. 2224-2236, 2008.

[107] B. D. Liu, C. Y. Chen, and J. Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, no. 1, pp. 32-53, 2001.

[108] F. Liu and J. M. Mendel, "An interval approach to fuzzistics for interval type-2 fuzzy sets," in *Proceedings of the FUZZ-IEEE*, London, UK, 2007, pp. 1-6.

[109] F. Liu and J. M. Mendel, "Encoding words into interval type-2 fuzzy sets using an interval approach," *IEEE Trans. on Fuzzy Systems*, vol. 16, no. 6, pp. 1503-1521, 2008.

[110] F. Liu, J. M. Mendel, and A. M. Nejad, "Forecasting Injector-Producer Relationships From Production and Injection Rates Using an Extended Kalman Filter," *SPE Journal*, vol. 14, no. 4, pp. 653-664, 2009.

[111] L. A. Lucas, T. M. Centeno, and R. M. Delgado, "General type-2 inference systems: analysis, design and computational aspects," in *Proc. of FUZZ-IEEE*, London, 2007, pp. 1107-1112.

[112] A. De Luca and S. Termini, "A definition of non-probabilistic entropy in the setting of fuzzy sets theory," *Information and Computation*, vol. 20, pp. 301-312, 1972.

[113] W. Luo, "An efficient detail-preserving approach for removing impulse noise in images," *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 413-416, July 2006.

[114] C. Lynch, H. Hagras, and V. Callaghan, "Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines," in *Proceedings of the FUZZ-IEEE*, Vancouver, Canada, 2006, pp. 7217-7224.

[115] Z. Ma, H. R. Wu, and B. Qiu, "An robust structure adaptive hybrid vector filter for color image restoration," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 1900-2001, December 2005.

[116] P. Melin and O. Castillo, "An intelligent hybrid approach for industrial quality control combining neural networks, type-2 fuzzy logic and fractual theory," *Information Sciences*, vol. 177, no. 7, pp. 1543-1557, 2007.

[117] P. Melin et al., "Voice recognition with neural networks, type-2 fuzzy logic and genetic theory," *Information Sciences*, vol. 13, no. 2, pp. 108-116, 2006.

[118] J. M. Mendel, "Adavances in type-2 fuzzy sets and systems," *Information Sciences*, vol. 177, no. 1, pp. 84-110, 2007.

[119] J. M. Mendel, "An architecture for making judgements using computing with words," *International Journal of Applied Mathematics and Computer Science*, vol. 12, no. 3, pp. 325-335, 2002.

[120] J. M. Mendel, "Comments on "Alpha-Plane Representation for Type-2 Fuzzy Sets: Theory and Applications"," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 229-230, Feb. 2010.

[121] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Englewood Cliffs, New Jersey, USA: Prentice Hall, 1995.

[122] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[123] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 117-127, 2002.

[124] J. M. Mendel and F. Liu, "Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set," *IEEE Trans. on Fuzzy Systems*, vol. 15, pp. 309-320, 2007.

[125] J. M. Mendel, F. Liu, and D. Zhai, "Alpha-plane representation for type-2 fuzzy sets: theory and applications," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 5, pp. 1189-1207, 2009.

[126] J. M. Mendel and H. Wu, "Centroid uncertainty bounds for interval type-2 fuzzy sets: Forward and inverse problems," in *Proceedings of the FUZZ-IEEE, vol. 2*, Budapest, Hungary, 2004, pp. 947-952.

[127] J. M. Mendel and H. Wu, "New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule," in *Proc. EUSFLAT-LFA*, Barcelona, Spain, 2005, pp. 872-877.

[128] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*. Hoboken, New Jersey: Wiley-IEEE Press, 2010.

[129] J. M. Mendel and H. Wu, "Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems," *IEEE Trans. on Fuzzy Systems*, vol. 14, no. 6, pp. 781-792, 2006.

[130] A. Mendez-Vazquez, P. Gader, J. M. Keller, and K. Chamberlin, "Minimum classification error training for choquet integrals with applications to landmine detection," *IEEE Trans. on Fuzzy Systems*, vol. 16, no. 1, pp. 225-238, 2008.

[131] M. J. Miranda and P. L. Fackler, *Applied Computational Economics and Finance*, 1st ed. Boston, Massachusetts, USA: The MIT Press, 2002.

[132] T. M. Mitchell, *Machine Learning*, 1st ed., E. M. Munson, Ed. Columbus, OH, USA: McGraw-Hill, 1997.

[133] S. Morillas, V. Gregori, G. Peris-Fajarne, and P. Latorre, "A fast impulsive noise color image filter using fuzzy metrics," *Real-Time Imaging*, vol. 12, no. 5, pp. 629-639, May 1990.

[134] A. Niewiadomski and M. Bartyzel, "Elements of type-2 semantics in summarizing databases," *Lecture Notes in Artificial Intelligence*, vol. 4029, pp. 278-287, 2006.

[135] A. Niewiadomski and P. S. Szczepaniak, "News generating based on type-2 linguistic summaries of databases," in *Proceedings of the IPMU*, Paris, France, 2006, pp. 1324-1331.

[136] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Berlin, New York: Springer-Verlag, 2006.

[137] D. S. Oliver, "The Sensitivity of Tracer Concentration to Nonuniform Permeability and Porosity," *Transport in Porous Media*, vol. 30, no. 2, pp. 155-175, 1998.

[138] C.-M. Own, H.-H. Tsai, P.-T. Yu, and Y.-J. Lee, "Adaptive type-2 fuzzy median filter design for removal of impulse noise," *Imaging Science*, vol. 54, no. 1, pp. 3-18, 2006.

[139] T. Ozen and J. M. Garibaldi, "Effect of type-2 fuzzy membership funcion shape on modelling variation in human decision making," in *Proceedings of the FUZZ-IEEE*, Budapest, Hungary, 2004, pp. 971-976.

[140] M. N. Panda and A. K. Chopra, "An integrated approach to estimate well interation," in *SPE India Oil and Gas Conference and Exhibition*, New Delhi, 1998, SPE39563.

[141] J. B. Paris, *The Uncertain Reasoner's Companion - A Mathematical Perspective.*: Cambridge University Press, 1994.

[142] H. C. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max relevance, and min-redundancy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, 2005.

[143] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 629-639, May 1990.

[144] J. Pevsner, *Bioinformatics and Functional Genomics*, 1st ed. Hoboken, New Jersey, USA: John Wiley & Sons, 2009.

[145] J. O. S. A. Pizarro, "Estimating injectivity and lateral autocorrelation in heterogeneous media," University of Texas at Austin, Austin, TX, Ph.D Dissertation 1998.

[146] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*, 1st ed. Heidelberg, Germany: Springer, 2000.

[147] K. N. Plataniotis, S. Vinayagamoorthy, D. Androutsos, and A. N. Venetsanopoulos, "An adaptive nearest neighbor multichannel filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 6, pp. 699-703, December 1996.

[148] G. Pok, Y. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Trans. on Image Processing*, vol. 12, no. 1, pp. 85-92, 2003.

[149] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4th ed. Hoboken, New Jersey, USA: Wiley-Interscience, 2007.

[150] S. M. M. Rahman and M. K. Hasan, "Wavelet-domain iterative center weighted median filter for image denoising," *Signal Processing*, vol. 83, no. 5, pp. 1001-1012, 2003.

[151] B. T. Refunjol, "Reservoir characterization of North Buck Draw field based on tracer response and production/injection analysis," University of Texas at Austin, Austin, Texas, Master Thesis 1996.

[152] F. C.-H. Rhee, "Uncertainty fuzzy clustering: insights and recommendations," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 44-56, 2007.

[153] L. Rokach and O. Z. Maimon, *Data Mining with Decision Trees: Theroy and Applications*, 4th ed. Singapore: World Scientific Publishing Company, 2006.

[154] F. Russo, "Fire operators for image processing," *Fuzzy Sets and Systems*, vol. 103, no. 2, pp. 265-275, 1999.

[155] F. Russo, "Impulse noise cancellation in image data using a two-output nonlinear filter," *Measurement*, vol. 36, pp. 205–213, 2004.

[156] F. Russo, "Noise removal from image data using recursive neurofuzzy filters," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 2, pp. 307-314, April 2000.

[157] F. Russo and G. Ramponi, "A fuzzy filter for image corrupted by impulse noise," *IEEE Signal Processing Letters*, vol. 3, no. 6, pp. 168-170, June 1996.

[158] M. Sayarpour, C. S. Kabir, and L. W. Lake, "Field Applications of Capacitance-Resistance Models in Waterfloods," *SPE Reservoir Evaluation & Engineering*, vol. 12, no. 6, pp. 853-864, 2009.

[159] S. Schulte, M. Nachtegael, V. De Witte, D. Van der Weken, and E. E. Kerre, "A fuzzy impulse noise detection and reduction method," *IEEE Trans. on Image Processing*, vol. 15, no. 5, pp. 1153-1162, May 2006.

[160] R. Sepulveda, O. Castillo, P. Melin, A. Rodriguez-Diaz, and O. Montiel, "Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic," *Information Sciences*, vol. 177, no. 10, pp. 2023-2048, 2007.

[161] P. Sevastjanov and P. Figat, "Aggregation of aggregating modes in MCDM: synthesis of type 2 and level 2 fuzzy sets," *Omega*, vol. 35, no. 5, pp. 505-523, 2007.

[162] H. Shu and Q. Liang, "Wireless sensor network lifetime analysis using interval type-2 fuzzy logic systems," in *Proceedings of the FUZZ-IEEE*, Reno, NV, 2005, pp. 19-24.

[163] D. Simon, *Optimal State Estimation*. New Jersey: John Wiley & Sons, 2006.

[164] D. Simon and T. L. Chia, "Kalman Filtering with State Equality Constraints," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128-136, 2002.

[165] D. Simon and D. L. Simon, "Kalman Filtering with Inequality Constraints for Turbofan Engine Health Estimation," *IEE Proceedings-Control Theory and Applications*, vol. 153, no. 3, p. 9, May 2006.

[166] B. Smolka and A. Chydzinski, "Fast detection and impulsive noise removal in color images," *Journal of Real-Time Image Processing*, vol. 15, no. 5, pp. 389-402, 2005.

[167] B. Smolka et al., "Self-adaptive algorithm of impulsive noise reduction in color images," *Pattern Recognition*, vol. 35, no. 8, pp. 1771-1784, 2002.

[168] T. Soeriawinata and M. Kelkar, "Reservoir management using production data," in *SPE Mid-Continent Operation Symposium*, Okalahoma City, 1999, SPE52224.

[169] J. T. Starczewski, "A triangular type-2 fuzzy logic system," in *Proc. of FUZZ IEEE*, British Columbia, 2006, pp. 7231-7238.

[170] J. T. Starczewski, "Extended triangular norms on Gaussian fuzzy sets," in *Proc. of EUSFLAT-LFA*, Barcelona, 2005, pp. 872-877.

[171] D. Stojanova, P. Panov, A. Kobler, S. Dzeroski, and K. Taskova, "Learning to predict forest fires with different data mining techniques," in *Proc. of 9th International Multiconference on Information Society (IS 2006)*, Ljubljana, Slovenia, 2006, pp. 622–629.

[172] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognition Letter*, vol. 15, no. 4, pp. 341-347, 1994.

[173] J. Sun, W. B. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proc. of 2004 IEEE Conference on Cybernetics and Intelligent Systems*, 2004, pp. 111-116.

[174] H. Tahayori, A. G. B. Tettamanzi, and G. D. Antoni, "Approximated type-2 fuzzy set operations," in *Proc. of FUZZ IEEE*, Vancouver, 2006, pp. 9042-9049.

[175] G. C. Thakur and A. Satter, *Integrated Waterflood Asset Management*. Tulsa, Oklahoma, USA: PennWell Corporation, 1998.

[176] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. of IEEE International Conference on Computer Vision*, Bombay, India, 1998.

[177] S. E. Umbaugh, *Computer Vision and Image Processing: A Practical Approach using CVIPTools*. Upper Saddle River, New Jersey, USA: Prentice Hall, 1998.

[178] C. Vega-Garcia, B. Lee, P. Woodard, and S. Titus, "Applying neural network technology to human-caused wildfire occurence prediction," *AI Applications*, vol. 10, no. 3, pp. 9-18, 1996.

[179] D. Van De Ville et al., "Noise reduction by fuzzy image filtering," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 429-436, August 2003.

[180] C. Wagner and H. Hagras, "Towards general Type-2 Fuzzy Logic Systems based on zSlices," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 637 - 660 , August 2010.

[181] C. Wagner and H. Hagras, "z slices - towards bridging the gap between interval and general type-2 fuzzy logic," in *Proc. of FUZZ IEEE*, Hong Kong, 2008, pp. 489-497.

[182] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. on Circuits and Systems*, vol. 46, no. 1, pp. 78-80, 1999.

[183] F. Wenstop, "Quantitative analysis with linguistic values," *Fuzzy Sets and Systems*, vol. 4, pp. 99-115, 1980.

[184] P. S. Windyga, "Fast impulsive noise removal," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 173-179, 2001.

[185] S. Wonneberge, "Generalization of an invertible mapping between probability and possibility," *Fuzzy Sets and Systems*, vol. 64, no. 2, pp. 229-240, 1994.

[186] D. Wu and J. M. Mendel, "A comparative study of ranking methods, similarity measures and uncertainty measure for interval type-2 fuzzy sets," *Information Sciences*, vol. 179, pp. 1169-1192, 2009.

[187] D. Wu and J. M. Mendel, "A vector similarity measure for linguistic approximation: Interval type-2 and type-1 fuzzy sets," *Information Sciences*, vol. 178, pp. 381-402, 2008.

[188] D. Wu and J. M. Mendel, "Aggregation using the linguistic weighted average and interval type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1145-1666, 2007.

[189] H. Wu and J. M. Mendel, "Antecedent connector word models for interval type-2 fuzzy logic systems," in *Proceedings of the FUZZ-IEEE, vol. 2*, Budapest, Hungary, 2004, pp. 1099-1104.

[190] D. Wu and J. M. Mendel, "Enhanced Karnik- Mendel algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 4, pp. 923-934, 2009.

[191] D. Wu and J. M. Mendel, "Uncertainty measures for interval type-2 fuzzy sets," *Information Sciences*, vol. 177, pp. 5378-5393, 2007.

[192] D. Wu and W. W. Tan, "A simplified type-2 fuzzy controller for real-time control," *ISA Transactions*, vol. 15, no. 4, pp. 503-516, 2006.

[193] D. Wu and W. W. Tan, "Genetic learning and performance evaluation of type-2 fuzzy logic controllers," *International Journal of Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 829-841, 2006.

[194] M. Wygralak, *Cardinalities of Fuzzy Sets*. Heidelberg: Springer, 2003.

[195] M. Xi, J. Sun, and W. Xu, "An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position," *Applied Mathematics and Computation*, vol. 205, pp. 751-759, 2008.

[196] H. Xu, G. Zhu, H. Peng, and D. Wang, "Adaptive fuzzy switching filter for images corrupted by impulse noise," *Pattern Recognization Letters*, vol. 25, no. 15, pp. 1657-1663, November 2004.

[197] R. R. Yager, "A measurement-informational discussion of fuzzy union and fuzzy intersection," *International J.of Man-Machine Studies*, vol. 11, pp. 189-200, 1979.

[198] R. R. Yager, "Fuzzy subsets of type-2 in decisions," *Journal of Cybernetics*, vol. 10, pp. 137-159, 1980.

[199] R. R. Yager, "Level sets and the extension principle for interval valued fuzzy sets and its application to uncertainty measures," *Information Sciences*, vol. 178, pp. 3565-3576, 2008.

[200] R. R. Yager, "Quantified propositions in a linguistic logic," in *Proceedings of the 2nd International Seminar on Fuzzy Set Theory*, Linz, Australia, 1980, pp. 69-124.

[201] N. Yamashita, M. Ogura, J. Lu, H. Sekiya, and T. Yahagia, "Random valued impulse noise detector using level detection," in *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, 2005, pp. 6292-6595.

[202] C. Yang and E. Blasch, "Kalman Filtering with Nonlinear State Constraints," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 70-84, Jan 2009.

[203] J.-X. Yang and H.-R. Wu, "Mixed Gaussian and uniform impulse noise analysis using robust estimation for digital images," in *16th International Conference on Digital Signal Processing*, Santorini-Hellas, Greece, 2009, pp. 1-5.

[204] M. T. Yildirim, A. Baştürk, and M. E. Yüksel, "Impulse noise removal from digital images by a detail-preserving filter based on type-2 fuzzy logic," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 920-928, 2008.

[205] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. on Circuits and Systems II*, vol. 43, no. 3, pp. 157-192, 1996.

[206] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," *IEEE Trans. on Signal Processing*, vol. 39, no. 2, pp. 395-410, 1991.

[207] A. A. Yousef, P. Gentil, J. L. Jensen, and L. W. Lake, "A Capacitance Model to Infer Connectivity From Production and Injection Rate Fluctuations," *SPE Reservoir Evaluation and Engineering*, vol. 9, no. 6, pp. 630-646, December 2006.

[208] S. Q. Yuan and Y. H. Tan, "Impulse noise removal by a global-local noise detector and adaptive median filter," *Signal Processing*, vol. 86, no. 8, pp. 2123-2128, 2006.

[209] M. E. Yüksel, "A hybrid neuro-fuzzy filter for edge preserving restoration of images corrupted by impulse noise," *IEEE Trans. on Image Processing*, vol. 15, no. 4, pp. 928-936, April 2006.

[210] M. E. Yüksel and A. Bastürk, "Efficient removal of impulse noise from highly corrupted digital images by a simple neuro-fuzzy operator," *International Journal of Electronics and Communications*, vol. 57, no. 3, pp. 214-219, 2003.

[211] M. E. Yüksel and E. Besdok, "A simple neuro-fuzzy impulse detector for efficient blur reduction of impulse noise removal operators for digital images," *IEEE Trans. on Fuzzy Systems*, vol. 12, no. 6, pp. 854-865, December 2004.

[212] Z. Chen, *Reservoir simulation: mathematical techniques in oil recovery*. Philadephia, Pennsylvania, USA: Society for Industrial and Applied Mathematics, 2007.

[213] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," *Computers and Mathematicas with Applications*, vol. 9, pp. 149-184, 1983.

[214] L. A. Zadeh, "A theory of approximate reasoning," in *Machine Intelligence*, J. E. Hayes, D. Michie, and L. I. Mikuhch, Eds. NY: Wiley, 1979, pp. 149-194.

[215] L. A. Zadeh, "From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions," *IEEE Transactions on Circiuts and Systems - I: Fundamental Theory and Applications*, vol. 4, pp. 105-119, 1999.

[216] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 103-111, 1996.

[217] L. A. Zadeh, "Fuzzy probabilities and their role in decision analysis," in *Proceedings of the IFAC Symposium on Theory and Application of Digital Control*, New Delhi, 1982, pp. 15-23.

[218] L. A. Zadeh, "Fuzzy Sets and Information Granularity," in *Advances in Fuzzy Set Theory and Applications*, M. Gupta, R. Ragade, and R. Yager, Eds. Amsterdam: North-Holland Publishing Co., 1979, pp. 3-18.

[219] L. A. Zadeh, "Possibility theory and soft data analysis," in *Mathematical Frontiers of the Social and Policy Sciences*, L. Cobb and R. M. Thrall, Eds. Boulder, CO: Westview Press, 1981, pp. 69-129.

[220] L. A. Zadeh, "PRUF, a meaning representation language for natural languages," *International Journal of Man-Machine Studies*, vol. 10, pp. 395-460, 1978.

[221] L. A. Zadeh, "Test-score semantics for natural languages and meaning representation via PRUF," in *Empirical Semantics*, B. B. Rieger, Ed. Boulder, CO: Westview Press, 1981, ch. 1, pp. 281-349.

[222] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning - 1," *Information Sciences*, vol. 8, pp. 199-249, 1975.

[223] L. A. Zadeh, "Toward a generalized theory of uncertainty (GTU) - an outline," *Information Sciences*, vol. 172, pp. 1-40, 2005.

[224] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems*, vol. 19, pp. 111-127, 1997.

[225] A. Zenebea, L. Zhoub, and A. F. Norcio, "User preferences discovery using fuzzy models," *Fuzzy Sets and Systems*, vol. 161, no. 23, pp. 3044–3063, December 2010.

[226] J. Zeng and Z.-O. Liu, "Type-2 fuzzy hidden Markov models and their applications to speech recognition," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 3, pp. 454-467, 2006.

[227] D. Zhai, M. Hao, and J. M. Mendel, "A non-singleton interval type-2 fuzzy logic system for universal image noise removal using quantum-behaved particle swarm optimization," in *Proc. of FUZZ-IEEE*, Taipei, ROC, 2011.

[228] D. Zhai and J. M. Mendel, "An Inequality-Constrained Extended Kalman Filter for Continual Forecasting of Interwell Connectivities in Waterfloods," in *SPE Western Regional Meeting*, Anaheim, CA, USA, 2010.

[229] D. Zhai and J. M. Mendel, "Uncertainty Measures for General Type-2 Fuzzy Sets," *Information Sciences*, Accepted for publication.

[230] D. Zhai, J. M. Mendel, and F. Liu, "A New Method for Continual Forecasting of Interwell Connectivity in Waterfloods Using an Extended Kalman Filter," in *SPE Western Regional Meeting*, San Jose, California, USA, 2009.

[231] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters," *IEEE Signal Processing Letter*, vol. 9, no. 11, pp. 360-363, 2002.

[232] Z. H. Zhan, J Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362-1381, 2009.