2015-04-24

State Estimation and Optimization of Large-Scale Dynamic Systems with Improved Particle Filters

Xiaoran Shi
University of Miami, xiaoran_eileen@163.com

UNIVERSITY OF MIAMI

STATE ESTIMATION AND OPTIMIZATION OF LARGE-SCALE DYNAMIC SYSTEMS WITH IMPROVED PARTICLE FILTERS

By

Xiaoran Shi

A DISSERTATION

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

Coral Gables, Florida

May 2015

UNIVERSITY OF MIAMI

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

STATE ESTIMATION AND OPTIMIZATION OF LARGE-SCALE DYNAMIC SYSTEMS WITH IMPROVED PARTICLE FILTERS

Xiaoran Shi

Approved:

_____ Nurcin Celik, Ph.D. Assistant Professor of Industrial Engineering	_____ Shihab S. Asfour, Ph.D. Professor of Industrial Engineering
_____ Kamal Premaratne, Ph.D. Professor of Electrical and Computer Engineering	_____ Murat Erkoc, Ph.D. Associate Professor of Industrial Engineering
_____ Nazrul Shaikh, Ph.D. Assistant Professor of Industrial Engineering	_____ M. Brian Blake, Ph.D. Dean of the Graduate School

SHI, XIAORAN (Ph.D., Industrial Engineering)
State Estimation and Optimization of Large-Scale Dynamic Systems with Improved Particle Filters. (May 2015)

Abstract of a dissertation at the University of Miami

Dissertation supervised by Professor Nurcin Celik.
No. of pages in text. (131)

This thesis explores novel methodologies for improving the particle filtering algorithm, and tackles state estimation and optimization problems of large-scale dynamic systems through the use of the improved particle filters. First of all, an importance density selection scheme for the particle filtering algorithm is first proposed based on the minimum relative entropy and the theorem of Taylor series expansion. By considering both the transition prior (previous states) and the likelihood (measurements), the proposed density selection scheme improves the performance of the particle filters especially when the measurements appear in the tail of the prior or the prior differs significantly from the posterior. Secondly, a particle filtering-based optimization algorithm for the multi-objective optimization problem is developed to establish a connection between the population-based optimization methods and the particle filtering algorithm. Here, the deterministic multi-objective optimization problem is represented using a state-space model. Then, samples (i.e., candidate solutions) are drawn from a distribution function, which can be computed recursively based on the performance of the prior particle set and the newly arrived observations. As the iteration progresses, the distribution function becomes more and more concentrated on the promising region of the

solution space, indicating the convergence capability of the proposed algorithm. When it comes to the practical contribution, two popular state estimation problems are studied. Specifically, a daily electricity demand forecasting problem is addressed by the way of incorporating the particle filters that embed the proposed density selection scheme into a developed state-space model. In a similar vein, a problem of low elevation target tracking over the sea surface in the presence of multipath effects is considered, and a corresponding tracking mechanism is proposed based on the state-space modeling methodologies and the improved particle filters. In addition to the state estimation problems, one of the most famous problems in the area of multi-objective optimization, which is the economic and environmental load dispatch (EELD) problem on an IEEE-30 bus system, is also included in this doctoral study. Experimental results are benchmarked against several algorithms studied in the literature. Through these practical state estimation and optimization problems, the validity and effectiveness of the proposed methodologies is successfully demonstrated. Finally, recommendations for further study are enclosed.

ACKNOWLEDGEMENTS

First of all, I would like to especially express my sincerest gratitude to my advisor Dr. Nurcin Celik, for her constant support and encouragement throughout the completion of my doctoral degree. She has guided me in research and in career, and has provided me with tremendous support in every aspect of my graduate life, for which I will always be grateful. Because of her dedication to the quality and integrity of research, I regard her as one of my role models for my career life in the future.

I would also like to thank my committee members: Dr. Shihab Asfour, Dr. Murat Erkoc, Dr. Nazrul Shaikh, and Dr. Kamal Premaratne, for their invaluable input into my doctoral research. Their rigorous attitudes towards science, words of encouragement and fair-minded assessment of this work, have inspired me to strive to do my best.

I would also like to state my appreciation to Dr. Daniel Berg, Dr. Vincent Omachonu, Dr. Nina Miville, Dr. Mei-Ling Shyu, Dr. Francesco Travascio, Ms. Praxie Alegria, Ms. Daritza Berio Blanco, Mr. Augusto Roca, and the rest of the faculty and staff at the University of Miami, who have helped me a lot along this path.

Moreover, thanks to all members in our Simulation and Optimization Research Lab (SimLab): Aristotelis Thanos, Mehrad Bastani, Talal Mohammadghazali Alyamani, Haluk Damgacioglu, Duygu Yasar, Greg Collins, Danielle Coogan, Tomas Cacidedo, Delante E. Moore, Juan Pablo Saenz, Eric D. Antmann, Breanna Hayton. I have been greatly influenced by their great passion in research and pursuit of new knowledge. I will always remember my experience cooperating with them and will always cherish our friendship.

Finally, I cannot express enough thanks to my father Mingshan Shi, my mother Shuying Du, my sister Xiaohong Shi, and my husband Zhijiang Wang, who always cheer me up, give me unconditional love, and stand by me through the hard times.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1. Overview of Particle Filtering

Particle filtering (also known as sequential Monte Carlo (SMC) methods) is a simulation-based estimation technique that is capable of handling massive datasets and diverse external factors. It is a feasible solution to the Bayesian filtering problem, in which the latent variables are related to the Hidden Markov Model (HMM), especially when nonlinearities and non-Gaussian distributions exhibited in the resulting models preclude the analytical solutions (van der Merwe et al., 2001). Particle filtering originates from the idea of representing the probability densities by a set of randomly chosen samples and estimating the posterior distribution as new data arrives (Gordon et al., 1993; Doucet et al., 2000). The particle filtering algorithm has displayed good performances in providing estimation for large scale and dynamic systems. As a result it has been heavily studied and applied to various fields, such as signal processing (Liu and Chen, 1995; Kotecha and Djuric, 2003), target tracking (Guivant and Nebot, 2003; Schiff and Goldberg, 2006; Martinez et al., 2008; Niknejad et al., 2012), power systems (Chen et al., 2011; Thanos et al., 2014), financial econometrics (Lopes and Tsay, 2011), supply chain management (Pardoe and Stone, 2007; Celik and Son, 2011), robotics (Renfrew et al., 2013; Kehoe et al., 2014), and manufacturing (Schirru et al., 2010), amongst many others.

A Bayesian filtering problem can be formulized as the computation of the posterior distribution $p(x_n|y_{1:n})$, where the state of the system x_n evolves with time and its estimation necessitates the usage of a sequence of noisy measurements $y_{1:n}$ (Arulampalam et al., 2002). This can be achieved via two steps, a prediction step (shown in (1.1)) where a sequential update of the filtered posterior distribution (i.e., $p(x_n|y_{1:n})$) can be computed, and an update step (shown in (1.2)) where the likelihood of

measurement $y_{1:n}$ is combined with the predicted state via Bayes rule. Here, the prediction step uses the system model to predict the probability distribution of the state forward from one measurement time to the next, and the update operation uses the latest measurement to trigger the update of the prediction of the probability density function,

$$p(x_n|y_{1:n-1}) = \int f(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1} \quad \text{(prediction step)} \quad (1.1)$$

$$p(x_n|y_{1:n}) = \frac{g(y_n|x_n)p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})} \quad \text{(update step)} \quad (1.2)$$

where $p(y_n|y_{1:n-1}) = \int p(x_n|y_{1:n-1})g(y_n|x_n)dx_n$.

In several linear Gaussian cases, the computations of prediction and update steps (i.e., equations in (1) and (2)) are analytically tractable. In this context, the well-known Kalman filter algorithm and its variations, such as extended Kalman filter (EKF) and unscented Kalman filter (UKF), do have the capabilities of providing estimations of the posterior with great accuracies. However, in most of the real-world applications, the state and measurement functions developed in the state-space model are not linear, thereby the prediction and update steps are difficult to execute or consume tremendous amounts of computation resources. In this context, one has to resort to the approximate methods, thereby triggering out the development of the particle filtering algorithm.

The generic particle filter consists of two steps in each iteration, which is importance sampling and resampling, respectively. In the importance sampling step, particles are sampled from a proposal distribution, which is also called the importance density. These particles are then weighted according to the ratio of the conditional density to the importance density and the Bayes' rule. In the next step, a resampling procedure is

triggered if most of the particles having insignificant weights. In the resampling step, a new set of particles are regenerated, and their importance weights are recalculated. These new particles go through the same steps at the next time/iteration. Finally, these weighted particles represent a discrete distribution, where support points equal to the locations of the particles and the associated probabilities are represented via the importance weights of the particles. The obtained discrete distribution is considered as an approximation of the true conditional distribution, and it will converge to the true conditional distribution when the number of particles drawn in each iteration increases to infinity.

From this perspective, it is noticed that the performance of the particle filtering algorithm relies heavily on importance sampling and, as a result, requires a good selection of the proposal distribution (importance density). The most popular sampling strategy is to draw samples from the transition prior, which is presented as the distribution of the current state given the previous state (i.e., $p(x_n|x_{n-1})$. The benefit of this choice of importance density is straightforward, as embedding the transition prior as the importance density is easy to implement and drawing sampling from the transition prior is uncomplicated. However, since it ignores the most recent measurement y_n, it results in a new problem – the degeneracy problem where all the mass is concentrated on a few samples with all other samples having negligible weights. This degeneracy problem may lead to a significant degradation of the performance of particle filters. In order to resolve this issue, many improved particle filters have been proposed with a better choice of the importance density, such as the Gaussian sum particle filter (Kotecha and Djuric, 2003), the auxiliary particle filter (Pitt and Shephard, 1999; Johansen and Doucet, 2008), and the unscented particle filter (van der Merwe et al., 2001; Cheng et al., 2008). To this

end, in the first part of this doctoral research, we develop a novel density selection scheme which takes both the transition prior and likelihood into consideration to improve the performance of the particle filters in terms of the estimation accuracy.

1.2. Particle Filtering for Radar Tracking

As mentioned in the above section, particle filtering has been heavily studied for solving state estimation problems of large-scale dynamic systems. Therefore, a target tracking problem, which is one of the most popular state estimation problems, is studied as part of this doctoral study.

Radar is a pulse technique-based object-detection system, used to identify the location, direction, or speed of the objects (i.e., ships, submarines, and aircrafts) by measuring the time delay between transmission of a pulse and its subsequent echo (Translation Bureau, 2013). While originally developed to satisfy the needs of the military for surveillance and weapon control, radar systems have been applied to various fields such as air traffic control for safe travelling (Li and Bar-Shalom, 1993; Collet et al., 2009), remote sensing for meteorological precipitation monitoring (Germann et al., 2006; Kollias et al., 2007), and geological observations (Radebaugh et al., 2007), etc. Yet, all of these applications rely significantly on high precision, and thus necessitate a radar that is capable of extracting objects from high noise levels and complex multipath interference, both of which frequently occur over the ocean surface.

Operation wise, radar emits radio waves that are called radar signals in predetermined directions and frequencies through its transmitter. Once these signals hit the target, they are reflected, amplified, and processed through the radar's receiver. While

only the signals that are directly reflected back towards the receiver are essential for the radar's estimates, there always exists more than one path for signals' propagation between the transmitter and receiver in the presence of sea surface multipath, as illustrated in Figure 1. The signals received from indirect paths interfere with the required signals in amplitude and phase. Such interferences result in the appearance of fake targets that behave like the normal targets, misleading the signal receiver when isolating the correct target echo and seriously degrading the tracking performance of the radar system. Therefore, the accuracy of radar systems is highly dependent on the development of a tracking mechanism that is capable of handling the effects of multipath propagation.

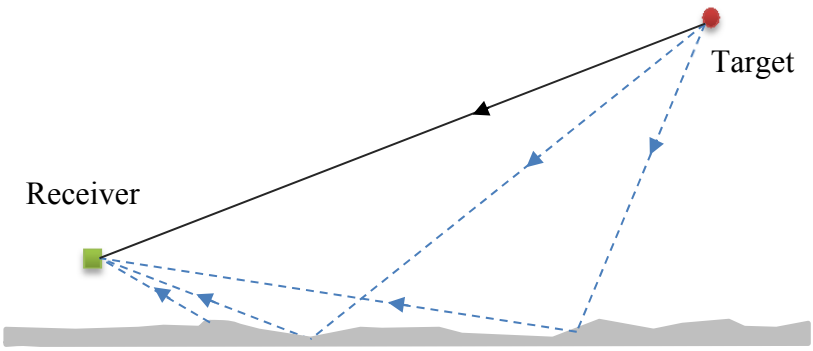


Figure 1: Multipath propagation from the ocean surface

Generally, in a target tracking problem, we aim to find the posterior state of a target (i.e., range, elevation, and velocity) based on sequentially received measurements (i.e., returned signal) via a radar system. However, this is not a trivial task. First, complex real-world situations (unstable sea-surface state, various noises, etc.) generate great difficulties in investigation and implementation of signal processing techniques, which are essential to any radar system. Second, transmitted waveform may vary momentously in terms of its characteristics (i.e., amplitude, frequency, signal-to-noise ratio, etc.), hindering the determination of the most suitable waveform type to be used in the radar

system. Moreover, achieving accurate estimation of a target's state in low elevation is especially challenging over the sea surface due to the multipath propagation. The impact of multipath propagation may depend on various factors such as the frequency of the radar waveform, height of the radar antenna, roughness of the sea-surface, the target's elevation and range, and meteorological conditions (Bruder and Saffold, 1991). It should be also noted that most of these factors are multifaceted and fuzzy, making the problem even more complicated.

In order to improve the estimation accuracy of a radar system when dealing with low elevation target tracking over the sea surface, a tracking mechanism is developed, which consists of a state-space model incorporating the multipath interference and a particle filtering algorithm estimating the target's state utilizing this newly developed state-space model. Here, the state-space model is established to ensure the robustness of the estimation of a desired elevation angle, which is very sensitive to phase differences between signals that return directly and indirectly. In order to derive a measurement function addressing such phase differences generated by all received signal echoes (considering that the phase differences are highly related to the target's location and signals' propagation), the following variables are incorporated into the proposed state-space model: (1) the range between a radar transmitter and a target R_k, (2) the elevation angle of a target θ_k, and (3) the current sea surface state S_k (i.e., roughness factor, surface reflection coefficient, etc.), where k indicates the time. Thenceforth, a particle filtering algorithm is incorporated to obtain the estimation of the target's state considering the specular and diffuse reflections in its measurement functions. Per se, the proposed algorithm relaxes the assumptions that the reflecting surface is perfectly smooth with no

other sources of existing interference and multipath noises are uncorrelated. It is the very same assumption that is often adopted by the literature focusing on the sea-surface target tracking problems (e.g., Ahn et al., 2010; Sen et al., 2011). Additionally, separate reflection types are considered for signals that return from different paths to make the investigation of the multipath interference more reliable against varying magnitudes of the surface reflection coefficient, the differences of received signal power, and the phase shifts.

1.3. Particle Filtering for Multi-objective Optimization

The particle filtering algorithms have been widely used to solve state estimation problems. However, to the best of our knowledge, it has been rarely developed for solving optimization problems, especially in the area of multi-objective optimization. As the third part of this doctoral study, a novel particle filtering-based multi-objective optimization algorithm is proposed.

Multi-objective optimization is an area of optimization where two or more objectives need to be optimized simultaneously while satisfying various constraints (Fonseca and Fleming, 1993; Ngatchou et al., 2005; Celik et al., 2012). Compared to the single-objective optimization problems, multi-objective optimization problems are more difficult to solve as the objectives always conflict with each other such that the potential optimal solutions contain a set of "best compromises" among different objectives instead of a unique solution. As such, most of the real-world multi-objective optimization problems are proven to be NP-hard (Zitzler and Thiele, 1999) where the Pareto front cannot be computed efficiently. Over the past two decades, these kind of problems have

been studied in a wide range of areas, including economics (Toffolo and Lazzaretto, 2002), system design and control (Eusuff and Lansey, 2003), sustainable energy systems (Abido, 2003), solid waste management (Antmann et al., 2012), and wireless sensors networks (Le Berre et al., 2011), amongst many others. To tackle these multi-objective optimization problems, a novel algorithm is proposed in this thesis, where a particle filtering algorithm with well-designed state evolution, measurement functions and an efficient sampling procedure is incorporated.

The proposed algorithm establishes a connection between the population-based optimization methods and the particle filtering algorithm. This idea comes from the similarities that the particle filtering algorithm shares with several population-based optimization methods, as shown in Figure 2. First, particles drawn from sampling distributions behave as if they are the candidate solutions generated from the solution space. Second, the importance weights assigned to these particles are then considered as the evaluation of the performances of the generated solutions. Last but not least, the sampling and resampling procedures in the particle filtering algorithm are similar to the searching and updating procedures in the population-based optimization methods. From this perspective, the deterministic multi-objective optimization problem is represented using a state-space model, in which the optimal solution is considered as the unobserved state that is yet to be "estimated" and the objective values observed are specified as an n-dimension measurements. The state is dynamically evolved based on the performance of the prior particle set and the newly arrived measurement. As the iteration progresses, the sampling distribution finally becomes concentrated on a particular region of the solution space, where promising solutions can be obtained with higher probabilities.

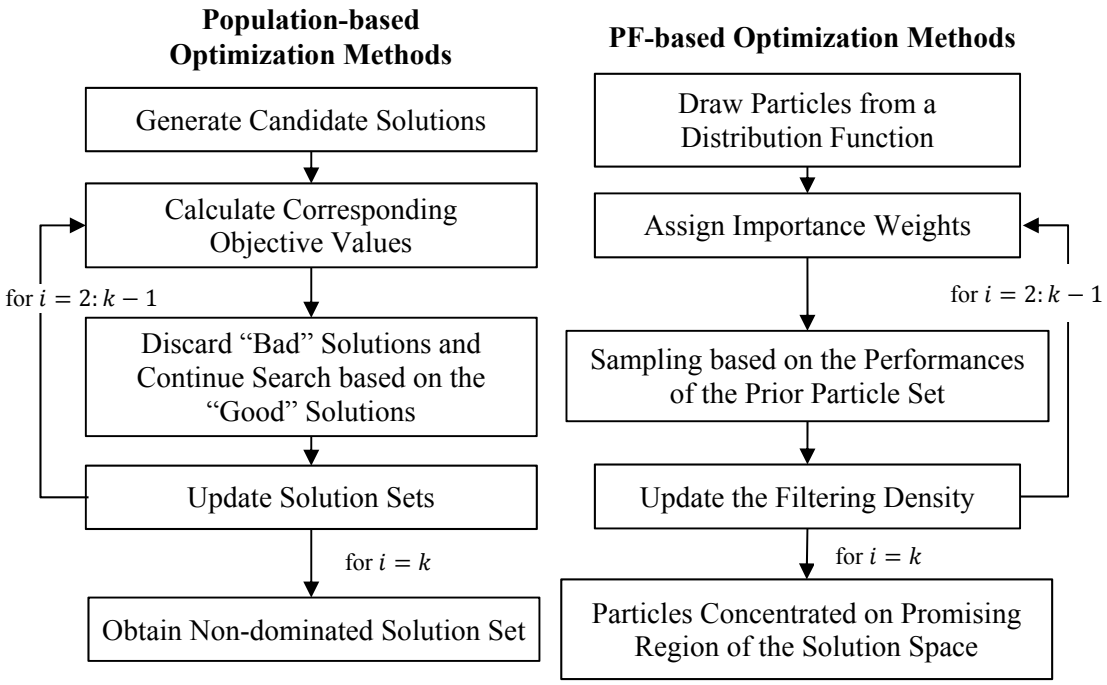


Figure 2: Comparison between population-based and the PF-based optimization methods

Although the core idea of the proposed algorithm is identified, there are two technical tasks to overcome in order to successfully address the multi-objective optimization problems. On one hand, for multi-objective optimization, no unique solution can accomplish all objectives due to the conflicts among these objectives. In this context, how to evaluate the performances of the obtained particles in terms of multiple objectives has to be resolved. On the other hand, convergence is a critical issue for any optimization algorithm. Therefore, a strategy that effectively governs the evolution of the distribution functions to realize fast convergence without destroying the diversity of the solutions is a necessity. In order to remedy the aforementioned challenges, the proposed algorithm is composed of two major procedures, namely, a dynamic weighted allocation procedure (DWA) and a performance-based sampling and resampling procedure (PSR). The DWA procedure is developed to quantify the significance of each objective. In contrast to the

traditional weighted aggregation procedure, the periodically aggregated objective weights obtained via the proposed DWA procedure are added to calculate particles' importance weights, instead of being bundled with their corresponding objective functions. The proposed PSR procedure is equipped with two iterative steps: 1) generate candidate solutions from a sampling distribution; 2) select the candidates with "good" performances and update the sampling distribution using the selected solutions for the next iteration. Moreover, an effective resampling rule is also embedded to mutate the sampling distributions for breaking the local optima. In order to draw samples efficiently and achieve the optima without taking considerable computational efforts, an optimal computing budget allocation (OCBA) procedure is incorporated in the proposed algorithm. A detailed explanation of how these procedures are implemented is provided for a bi-objective optimization problem in this thesis. However, the proposed algorithm also presents the potential to be extended for solving multi-objective optimization problems.

1.4. Summary of Proposed Contributions

The main contributions of this research may be categorized as theoretical contributions to importance density selection scheme of the particle filtering algorithm and particle filtering-based optimization algorithm, and practical contributions to state estimation and optimization of large-scale dynamic systems using the improved particle filters. These contributions are summarized below:

- *Theoretical Contribution 1: A minimum relative entropy-based importance density selection scheme*

A novel importance density selection scheme for the particle filtering algorithm is proposed based on the minimum relative entropy and the theorem of Taylor series expansion. By considering both the transition prior (previous states) and the likelihood (measurements), the proposed density selection scheme improves the performance of the particle filters especially when the measurements appear in the tail of the prior or the prior differs significantly from the posterior.

- *Theoretical Contribution 2: A particle filtering-based optimization algorithm for multi-objective optimization problem*

A novel particle filtering-based algorithm for solving multi-objective optimization problems is proposed in this study. The proposed algorithm is equipped with a dynamic weighted allocation (DWA) procedure for the distribution of objective weights and a performance-based sampling and resampling (PSR) procedure for the generation of the candidate solutions and the update of the sampling distribution. As the iteration progresses, the distribution function, from which samples (candidate solutions) are generated, becomes more and more concentrated on the promising region of the solution space, indicating higher probabilities of obtaining solutions with good performances while lower chances to get "bad" ones.

- *Practical Contribution 1: Short-term load forecasting using improved particle filters*

Utility costs in terms of energy consumption accounts for a significant portion of total monthly expenditures in large-scale enterprises. Given the organization's production schedules, determination of the lowest cost method of providing the required amount of electricity relies heavily on accurate load forecasts for the forthcoming periods. From this

perspective, a daily electricity demand forecasting problem is addressed, by the way of incorporating the particle filters that embed the proposed density selection scheme into a developed state-space model. Through the case study conducted for a private company located in Florida, the validity and effectiveness of the proposed density selection scheme is successfully demonstrated.

- *Practical Contribution 2: A state-space modeling based tracking mechanism for low elevation target tracking in the presence of sea surface multipath interference*

Target tracking is considered as one of most popular Bayesian state estimation problems. In order to improve the estimation accuracy of a radar system when dealing with low elevation target tracking over the sea surface, a tracking mechanism is developed based on the state-space modeling methodologies and the improved particle filters. Both the specular reflection and the diffuse reflection are considered simultaneously when analyzing the returned radar signals, relaxing the most common assumption adopted in historical studies that the reflecting surface is perfectly smooth with no other sources of existing interference. Estimates of the target's states are achieved via the improved particle filters, which further illustrates the feasibility and validity of the proposed importance density scheme in various state estimation applications.

- *Practical Contribution 3 : A study of environmental and economic load dispatch (EELD) through the use of the proposed PF-based optimization algorithm*

In order to evaluate the performance of the proposed particle filtering-based multi-objective optimization algorithm, a study of environmental and economic load dispatch problem is conducted. The developed model is applied in a standard IEEE-30 bus system,

and the optimization is achieved via the proposed algorithm. Its performance is evaluated, benchmarking with that of other algorithms established in the literature. The obtained results have revealed the competitive performance of the proposed algorithm in terms of the diversity of the obtained Pareto optimal solutions and the convergence speed.

Chapter 2: Literature Review

In this chapter, previous works presented in the literature are summarized depending on their relationship to the study herein presented. A survey of methods focusing on effective sampling and the choice of importance density is summarized in Section 2.1, a survey of low elevation target tracking techniques in the presence of sea surface multipath is presented in Section 2.2, and a survey on methods addressing the multi-objective optimization problems is provided in Section 2.3.

2.1. Literature on Effective Sampling

Particle filtering relies heavily on effective sampling from a sequence of probability distributions. Amongst the various sampling procedures presented in the literature, Monte Carlo sampling (MCS), importance sampling (IS), sequential importance sampling (SIS), and sequential importance sampling with resampling (SISR) denote the most widely used four methods. The comparison of these four sampling methods in terms of their main ideas, as well as their respective advantages and disadvantages, is provided in Table 1.

Table 1: Comparison of sampling methods in SMC

Method	Main Idea	Benefits	Drawbacks
MCS	▪ Estimate $p(x)$ by the discrete distribution $\hat{p}_n(x) = \frac{1}{N_s}\sum_{i=1}^{N_s} \delta(x - x^{(i)})$, where $x^{(i)}$ are the particles with weights $w^{(i)} = \frac{1}{N_s}$, $\{x^{(i)}\}_{i=1}^{N_s}$ are independent samples from $p(x)$	▪ Straightforward framework ▪ emphasize the more weighted variables of $p(x)$ ▪ Rate of convergence does not directly depend on dim(x)	▪ Difficult to sample from the $p(x)$ in most cases ▪ Computational complexity of the algorithm increases with n
IS	▪ Draw samples from the importance distribution q(x), and approximate $p(x) \approx \hat{p}_N(x) = \sum_{i=n}^{N} \tilde{w}^{(i)}\delta(x - x^{(i)})$, where $x^{(i)} \sim q(x), i = 1, \cdots, N_s$, and $w^{(i)} = p(x^{(i)})/q(x^{(i)})$	▪ Easier to sample from q than p ▪ Covariance σ^2 may be small, leading to a faster convergence and a greater degree accuracy	▪ Complicated selection of the importance distribution $q(x)$ ▪ Computational complexity increases at each time step

SIS	▪ At time n, sample particles $x_n^{(i)} \sim f(x_n \mid x_{n-1}^{(i)})$ and compute $w_n^{(i)} \propto w_{n-1}^{(i)} \cdot g(y_n \mid x_n^{(i)})$ ▪ Estimate $p(x_n \mid y_{1:n}) \approx \sum_{i=1}^{N_s} w_n^{(i)} \delta(x_{1:n} - x_{1:n}^{(i)})$	▪ Can be easily parallelized ▪ Computational complexity does not increase over time ▪ Weights only depend on $\{x_{n-1:n}^{(i)}\}$	▪ Only functions effectively in moderately sized problems ▪ As n increases, mass is concentrated in a few samples
SISR	▪ At time 1, sample N_s particles $x_1^{(i)} \sim \mu(x_1)$ and compute $w_1^{(i)} \propto 1/N_s$ ▪ At time n, sample N_s particles $x_n^{(i)} \sim f(x_n \mid x_{n-1}^{(i)})$ and compute $w_n^{(i)} \propto g(y_n \mid x_n^{(i)})$ ▪ Resample $\{x_n^{(i)}, w_n^{(i)}\}$ ▪ Estimate $p(x_n \mid y_{1:n}) \approx \sum_{i=1}^{N_s} w_n^{(i)} \cdot \delta(x_{1:n} - x_{1:n}^{(i)})$	▪ Variance reduction can be developed ▪ SMC methods can be used to sample from virtually any sequence of distributions ▪ May lead to less degeneracies	▪ Resampling rules have a major impact on the estimation accuracy ▪ Does not consider the observation when conducting estimation

From the perspective of the theoretical advancements of particle filtering, many studies have been conducted on effective sampling from a sequence of probability distributions due to its critical role in the performance of the particle filtering and its capacity to resolve the potential issues involving weight degeneration. The idea behind the sampling mechanism can be summarized in two aspects. First, emphasizing the values of the input variables that have a relatively greater impact on the resultant parameters of interest (than others) by sampling more frequently reduces the estimator variance. Second, the simulation outputs are weighted in order to make sure that the estimator is unbiased even when they are used for the estimation of biased distributions. When it comes to the selection of the sampling density, one of the most common strategies, proposed by Gordon et al. (1993), is to draw samples from the transition prior. While this widely used strategy may be computationally and algorithmically easy to implement, it does not take into consideration the impact of the measurements. To avoid this shortcoming, several variations of the generic particle filtering algorithm are proposed, among which the auxiliary particle filter (Pitt and Shephard, 1999) allows to obtain particles from the

optimal importance density by introducing an auxiliary variable, the unscented particle filter (also known as sigma-point particle filter) (van der Merwe et al., 2001) uses the unscented Kalman filter (UKF) to generate its importance distribution, and the Gaussian sum particle filter (Kotecha and Djuric, 2003) approximates the proposal distribution as Gaussian mixtures, etc.

In a similar vein, Moral et al. (2006) propose a path for sampling based on the idea of minimizing the variance of the estimates and the importance weights. They provide the expressions for the asymptotic variances which are associated with central limit theorems. Simulations demonstrate that the method is potentially powerful. However, significant limitations are encountered on the application of the proposed method because of the restrictive conditions, which can be summarized as the specific definitions of the sequence of targets and the selection of the initial sampling density. In order to apply particle filtering into the evolutionary biology and epidemiology related problems, in which the likelihood function is analytically or computationally intractable, Sisson et al. (2007) propose a sampling method without likelihood based on the improved rejection sampling and the approximate Bayesian computation algorithm. Importance weights in their work are specified to the adherences of the particles to the target distribution. This method relies greatly on the selection of the prior distribution, and fails to provide a reasonable estimation if the sampling distribution and the target distribution mismatch. Furthermore, Wen and Qicong (2007) apply artificial neural networks theory for the purpose of obtaining a corresponded importance density. In their work, samples drawn from the prior density are adjusted with general regression neural network (GRNN) proposed by Specht (1991) to approximate the importance density. Then, another set of

samples are drawn from the estimated density function to achieve the posterior distribution. Particle filters that embed their proposed method provides improved estimation accuracies than the prior-sampling based particle filters. However, these improvements are achieved at the expense of a dramatic increase in the computational resource usage needed for the realization of GRNN.

Following the contributions of these previous studies, in the first part of this doctoral research, my focus is to refine the particle filtering algorithm by developing a scheme for the selection of effective importance densities in order to avoid the potential threat of degeneracy and provide better estimation accuracies, especially when the current observation produces significant impacts on the posterior states. Based on the recursive updating step of the importance weights, a minimum relative entropy-based importance density selection rule is proposed taking both the previous state and the current observation into account.

2.2. Literature on Low Elevation Target Tracking

Low elevation target tracking in the presence of sea surface multipath interference has been studied in the literature from different perspectives. Multipath effects consist of specular and diffuse reflections. The specular reflection causes significant signal fading whereas the diffuse reflection causes an approximately constant bias and higher standard deviation to the in-phase component of the monopulse ratio (Sinha, et al., 2003). Sherman (1971) points out that the multipath propagation leads to a degradation in the true measurement of returned signals, and analyzes the effect of the specular reflection in the monopulse tracking. Sherman's theoretical deviation for the effect of specular

reflection has then become the foundation for the studies concerning target tracking with multipath interference over the ocean surface. More recently, Ahn et al. (2010) propose a phase comparison method to compensate the angle estimation error due to the interference caused by the directly reflected signals (signals that received from specular reflection). In addition, they divide the phased array antenna of radar into two identical, partially overlapped sub-apertures to get supplementary degrees of freedom and sharp beams. Consequently, an accurate estimation of the elevation angle of a target can be obtained without sustaining a specific monopulse slope. In a similar vein, Han et al. (2010) propose a hybrid technique for low elevation tracking in the presence of sea surface multipath effects. Based on the idea that it is necessary to remove coherency of the data vector before estimating a target's direction of arrival, a principle is applied to eliminate the correlation so that one can distinguish the real angle of elevation from the grazing angle of specular reflection. This, in turn, will promote the accuracy and performance of the angle measurement by using a multiple signal classification algorithm. Simulation results presented in Ahn's and Han's works demonstrate good performance of their proposed methods. Yet, both of these two works rely heavily on the assumption that the reflecting surface is perfectly smooth with only specular reflection. This largely unrealistic assumption makes their proposed methods inapplicable to many complex real-world situations.

As opposed to the strong assumptions made in the aforementioned literature, the ocean surface typically acts as an imperfect mirror with frequently occurring diffuse reflections. Blair and Brandt-Pearce (2001) demonstrate the nature of the diffuse reflection for Rayleigh target tracking by a complex Gaussian processing model. They

also provide a method of calculation for the diffuse reflection coefficient in the same work. In their work, a probability density function of monopulse measurements for a Rayleigh target is presented and statistics of multipath-corrupted monopulse measurements are illustrated. Finally, they have conducted their tracking scheme through Extended Kalman Filters (EFK) which incorporates these functions. However, their work also suffers from reliance on the assumption that there is only one source of interference (diffuse reflection) when dealing with the low-elevation tracking problems. Additional studies focused on the investigation of multipath reduction techniques are compared in Table 2. These techniques may provide sufficient measurement error reduction under the assumptions that there is no diffuse reflection on the sea surface, and/or the impact of specular reflection is small enough to be ignored over rough sea-surfaces. However, the development of an algorithm that is not bound by these assumptions and takes into account both the specular and diffuse reflections is needed to improve the precision when dealing with the target tracking, especially those related to critical navy operations.

Table 2: Selected works on investigation of multipath reduction techniques

Authors (year)	Method	Main Idea	Advantages	Disadvantages
Jao (1994)	Matched array beamforming	Using a predicted signal vector in the array beamforming process, the true target altitude may be estimated from the maximum of the array response	▪ Process is designed to be fully matched to the target propagation environment	▪ Only specular multipath interference is considered ▪ Does not consider the sensitivity of the array response
Inaba and Araki (2004)	Maximum ratio combining	The delay time difference between returned signals and Doppler frequency shifts can be utilized to extract waveforms from different paths	▪ Size of the transmission and reception system is decreased ▪ Computational overhead is reduced	▪ Array configuration in the azimuth direction is neglected
Okuda et al. (2005)	GPS-based simulation	Effect of ocean surface reflection of multipath signal can be reduced by lowering	▪ Wave height and direction can be accurately obtained	▪ Accuracy of proposed system relies heavily on the

		the antenna height of the radar		frequency of the waveform
Chung et al. (2006)	Complex indicated angle algorithm	If the complex indicated angle is a complex number, the signals received by the monopulse radar indicate that multipath propagation exists or there are unresolved targets	▪ Locations that the measurements are affected by the specular reflection are clearly indicated	▪ Lower estimation accuracies
Yedukondalu et al. (2011)	Adaptive filtering	Multipath error can be mitigated using the prominent adaption algorithms	▪ Estimation error can be reduced to a great extent	▪ Very prone to noise

Moreover, simulation-based algorithms and state-space modeling methods in dealing with low elevation target tracking problems with the multipath interference over the sea surface have attracted many researchers' attention in the past. The extended Kalman filtering (EKF) has been one of the most widely studied algorithms (Ramachandra et al., 1993; Kramer and Stubberud, 2008; Melzi et al., 2010; Tay et al., 2011). However, a highly matched priori density is required for the measurements while using the EKF-based algorithms. Belonging to the same family of filtering yet releasing this matched priori requirement, particle filtering algorithms have been studied extensively in other works for the same target tracking problem (Okuma et al., 2004; Khan et al. 2005; Djurić et al., 2008; Wu and Su, 2010; Jatoth et al., 2013). To this end, Jishy and Lehmann (2009) propose an adaptive filtering procedure to cancel any unwanted signals returned due to the multipath propagation, and consequently separate the direct signals from the totality of received signals. While their method provides a complete and homogeneous solution for various tracking problems, estimation accuracy of their algorithm heavily relies on the predetermined auxiliary functions for the adaptive filtering and its implementation requires significant computational effort as the matched filter has to be specified for each target echo.

Developing on these works, in the second part of this doctoral research, my goal is to investigate the impact of multipath interference on the estimation of a target's location and trajectory (i.e., elevation angle, range, etc.) to address the tracking problem over the sea surface considering both specular and diffuse reflections, simultaneously. Estimation of the target's posterior state is obtained via the particle filters embedding the improved density selection scheme proposed in the first part of this research.

2.3. Literature on Multi-objective Optimization

Many researchers have contributed to the literature of multi-objective problems via three major approaches, namely, aggregation approaches, population-based non-Pareto techniques, and Pareto-based approaches (Ngatchou et al., 2005). The aggregation approach (also known as the weighted-sum or scalarization methods) and its variants address the multi-objective optimization problem by transforming it into a single-objective problem via assigning a weight to each objective based on the priori preferential information provided by decision makers. With this approach, a single best compromise solution is obtained finally. Their heavy dependence on this priori information comprises a major weakness for these approaches in general. Table 3 summarize several major methods belonging to this category and provides a comparison among them in terms of their major idea and formulation, as well as advantages and disadvantages of these respective methods.

Table 3: Comparison of selected aggregation approaches and their variants

Approaches	Related Works	Formulation	Advantages & Disadvantages
Weighted Aggregation	• Tabucanon (1988)	$Min\ Z = \sum_{j=1}^{N} w_j f_j(x)$ with $w_j \geq 0$ and $\sum_{j=1}^{N} w_j = 1$	• Simple to implement • The weights must be specified for each

			objectives as a prior • The solution yields only one "optimal" result if the weights are fixed; the concave portion of the frontier is missed if the weights are dynamically changed.
Dynamic Weighted Aggregation	• Jin et al. (2001)	$Min\ Z = \sum_{j=1}^{N} w_j f_j(x)$ with $w_j \geq 0$ and $\sum_{j=1}^{N} w_j = 1$ (weights are dynamically changed)	• Simple to implement • Does not generally yield the non-dominating front • Misses the concave portion of the frontier • The diversity along the Pareto front is difficult to control
Goal Programming	• Clayton et al. (1982) • Tamiz et al. (1988)	$Min\ Z = \sum_{j=1}^{N} w_j \lvert f_j(x) - T_j \rvert$	• Priori information is needed
ε Constraint	• Becerra and Coello (2006) • Mavrotas, (2009)	$Min\ f_j(x), x \in \Omega$ s.t. $f_k(x) \leq \varepsilon_k,\ g_i(x) \leq 0$ $(k \neq j, k = 1,2, \dots N,$ $i = 1,2, \dots M)$	• Great computational burden • The solutions found are not necessarily globally non-dominated

In contrast to the aggregation-based techniques, several intelligent techniques are designed for direct generation of the Pareto front by optimizing the individual objectives simultaneously (Ngatchou et al. 2005), thereby offering greater flexibility for decision makers especially when the priori information is not available in cases of most real-world applications. The evolutionary algorithm (EA) is one major category of such techniques that has been heavily studied and successfully applied to the area of optimization. Evolutionary algorithms are adaptive search techniques inspired from nature and followed the principle of Darwin's survival-of-the-fittest theory (Tiwari et al., 2008). Optimization methods that belong to the class of EAs all operate on a set of candidate solutions, while the differences among them are the way that the fitness selection, mutation and crossover procedures are performed. Although the superior performances of these algorithms have been demonstrated in several studies, challenges encountered on how to guide the search towards the Pareto-optimal set and how to maintain a diverse population to prevent premature convergence still need to be further addressed.

In terms of the population-based non-Pareto approaches, the Vector Evaluated Genetic Algorithm (VEGA) proposed by Schaffer in 1985 has been extensively studied and referred to. In his work, the total population is split into several subsets according to the number of objectives that needs to be optimized. Each subset of population is used to independently optimize one objective. After a few iterations, these subsets are shuffled together, followed by a crossover and mutation operation to obtain new solutions. This algorithm is easy to implement. However, since the optimal solution for one objective may not be the global optima for the entirety of objectives, it is shown to fail to converge to compromise solutions. Another non-Pareto approach can be found in Hajela and Lin's work (1992), known as the Hajelas and Lins Genetic Algorithm (HLGA). This algorithm uses the weighted method for fitness assignment, and the weights assigned to different objectives are encoded in the genotype for parallel searching of multiple solutions in a single run. Moreover, phenotypic fitness sharing is utilized in their algorithm to maintain the diversity in terms of the allocated weights to different objectives.

The Pareto approaches, which have also been investigated in the literature, are developed based on the idea of obtaining a Pareto front by simultaneously optimizing the individual objectives. Pareto-based approaches explicitly use Pareto-ranking for determining the probability of replication of an individual. In particular, these approaches aim to find the set of non-dominated individuals in the population, and then assigned the highest rank to them for the procedure of eliminating in the future iterations. The process is repeated as the iteration processes, with all remaining individuals in the entire population ranked and assigned with a fitness value. In conjunction with Pareto-based fitness assignment, a niching mechanism is used to prevent the algorithm from

converging to a single region of the Pareto front (Coello, 1999). A popular niching technique called sharing is heavily used in the computation of fitness value in their works, which regulates the density of solutions in the hyperspace spanned by either the objective vector or the decision vector. After that, mutation and crossover operations are performed to get the next generation of individuals. One of the simple and efficient methods is Multi Objective Genetic Algorithm (MOGA), which is proposed by Fonseca and Fleming in 1993. In their algorithm, the fitness value of an individual is proportional to the number of other individuals it dominates. Niching can be performed either in the objective space or the decision space. Another version is the Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1994), in which a layered classification technique is used. Differing from the MOGA, in the NSGA, all non-dominated individuals are assigned the same fitness value and sharing is applied in the decision variable space. The process is repeated for the remainder of the population with a progressively lower fitness value assigned to the non-dominated individuals. Meanwhile, Horn et al. (1994) propose a method called the Niched Pareto Genetic Algorithm (NPGA). In this method, two individuals are compared with respect to a comparison set (usually 10% of the entire population) instead of bilateral direct comparison. When one candidate is dominated by the set while the other is not, the latter is selected. If neither or both the candidates are dominated, fitness sharing is used to decide selection. NPGA introduces a new variable (size of the comparison set), but is computationally faster than the previous techniques, since the selection step is applied only to a subset of the population. Afterwards, Zitzler and Thiele (1999) introduce a Strength Pareto Evolutionary Algorithm (SPEA) that provides the traditional evolutional algorithms with an additional set of population to

archive the non-dominated solutions. With this additional archive, the fitness of other candidate solutions is evaluated with respect to the Pareto dominance relationship. Meanwhile, further progress has been made on several advanced methods, including the Pareto Envelope-based Selection Algorithm (PESA) (Corne et al., 2000) and NSGAII (Deb et al., 2002), amongst many others. In 2001, Zitzler et al. propose an improved version of the SPEA algorithm which is called SPEAII. Compared to the SPEA algorithm, the SPEAII shows its superiority from three aspects. First, the fitness assignment scheme is improved. For each individual, the number of individuals it dominates and the number of individuals it is dominated by are all considered. Second, to provide a more accurate guidance through the searching processes, a nearest neighbor density estimation technique is utilized. Last but not least, a new archive truncation method is incorporated into the SPEAII, which guarantees the preservation of boundary solutions. In a similar vein, Zhang and Li (2007) develop an algorithm based on the evolutionary algorithm, in which the MO problem is decomposed into a series of sub-problems using the Tchebycheff approach and each sub-problem is optimized via the evolutionary algorithm based on the information from its neighboring sub-problems. Further more recent developments on evolutionary algorithms for MO problems are made through the ant colony optimization algorithm and its variants (e.g., Qing et al., 2010; Lopez-Ibanez and Stuetzle, 2012). While successfully addressing the major limitation of the aggregation approaches (i.e., heavy utilization of prior information), these methods also suffer from the performance degradation and lack of computational efficiency as the number of objectives increases.

More recently, the Multi-Objective Particle Swarm (MOPS) algorithm gains a lot of popularity. The Particle Swarm Optimization (PSO) algorithm is first proposed by Kennedy and Eberhart in 1995, inspired by the choreography of a bird flock. The algorithm can be considered as a distributed behavioral algorithm that searches the optimal solution multi-dimensionally. In particular, each particle represents a solution in an n-dimensional space, and searches the global best solution by archiving its previous best experience throughout from the entire swarm. Recently, several algorithms are proposed as the extension of the PSO to handle multi-objective optimization problems (i.e., Coello and Salazar, 2002; Fieldsend and Singh, 2002; Raquel and Naval, 2005). Obviously, there is a tradeoff between convergence speed and the values of final fitness of nonlinear optimization methods. It means that improving one is at the expense of the other (Ji et al., 2008). To break out this dilemma, several researchers have worked on the combination of filtering and swarm moving for optimization problems. Monson and Seppi (2004) propose the Kalman Swarm algorithm, in which a reformulated PSO update equation is treated as the system dynamic in the state space model while the observation function records the best solution each particle has obtained in the past searching. This algorithm is proved to both significantly improve the final fitness value. However, as the major limitation encountered by the Kalman filter, this algorithm cannot be applied to nonlinear optimization problems. In the work presented by Ji et al (2008), a particle filter is incorporated to guide the movement of the particles to move towards location of the global optima sequentially by learning the information from the dynamic state-space model. Although these algorithms improve on many other PSO algorithms empirically, they lack a convergence guarantee. In 2008, Zhou et al. first propose to utilize the particle

filters in the area of optimization. In their work, the optimization problem is reformulated as a filtering problem, which leads to a desired sequence of filtering distributions that has proved convergence to the global optimal solution. Their work revealed the ability of applying the particle filtering algorithm to the optimization problems and the probability of extending it to the multi-objective optimization problems.

Given the new insights of applying particle filtering to solve the optimization problems, as the third part of this doctoral research, a particle filtering algorithm combined with an efficient sampling procedure is developed for multi-objective optimization.

Chapter 3: Improved Particle Filters with a Proposed Relative Entropy-based Density Selection Scheme

3.1. Generic Particle Filtering Algorithm and Degeneracy Problems

Bayesian filtering problems, in which the latent state variables are related to the Hidden Markov Model (HMM), are usually presented in a state-space form with a transition equation and an observation equation. In particular, the transition equation (as given in (3.1)) describes the hidden state process, where x_n and v_n represent the system state vector and the process noise (state noise) at time n, respectively, and $f_n(\cdot)$ is a time-dependent function describing the state evolution process. Here, the state vector is latent and can be obtained only through the distinct noisy measurement function, as shown in (3.2), where y_n is the vector of observed variables, h_n represent the observation noises and $g_n(\cdot)$ denotes the measurement function at time n.

$$x_n = f_n(x_{n-1}, v_{n-1}) \tag{3.1}$$

$$y_n = g_n(x_n, h_n) \tag{3.2}$$

In the Bayesian filtering problem, the goal is to obtain x_n given all the measurements up to (and including) time n (i.e., $y_{1:n}$). This can be considered as the estimation of the probability distribution of the posterior state $p(x_n|y_{1:n})$. In cases that the state and measurement functions are non-linear and the noises are not generated according to the Gaussian distributions, the analytical solutions are not tractable any more, thereby necessitating the use of the particle filtering algorithm.

Particle filtering (PF) originates from the idea of representing the target distribution (i.e., $p(x_n|y_{1:n})$) by samples sequentially drawn from a proposal distribution (also called importance density). As shown in Figure 3, after initialization, a set of samples $x_n =$

$\{x_n^1, x_n^2, \cdots, x_n^i\}$ ($i = 1,2, \cdots, N_s$) are generated from the proposal distribution $q(\tilde{x}_n|x_{n-1}, y_n)$ in the recursive importance sampling and resampling step.

[Step 1]: Initialization

(1) Draw N_s particles $x_0^{(i)}$ ($i = 1,2, \cdots, N_s$) from the initial density $\mu(x_0)$.

(2) Assign them equal weights $w_0^{(i)} = \frac{1}{N_s}$.

(3) Let $x_0 = \left\{x_0^{(i)}, w_0^{(i)}\right\}_{i=1}^{N_s}$.

[Step 2]: Importance Sampling and Resampling

Given $x_{n-1} = \{x_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^{N_s}$, take the following steps:

(1) Draw N_s particles $\tilde{x}_n^{(i)}(i = 1,2, \cdots, N_s)$ from the importance density $q(\tilde{x}_n|x_{n-1}, y_n)$.

(2) Set weights $\widetilde{w}_n^{(i)} = \frac{p\left(y_n\middle|\tilde{x}_n^{(i)}\right)p\left(\tilde{x}_n^{(i)}\middle|x_{n-1}^{(i)}\right)}{q\left(\tilde{x}_n^{(i)}\middle|x_{n-1}^{(i)}, y_n\right)} w_{n-1}^{(i)}$.

(3) Normalize the importance weights via $w_n^{(i)} = \frac{\widetilde{w}_n^{(i)}}{\sum_{j=1}^{N_s} \widetilde{w}_n^{(j)}}$.

(4) Resampling threshold check.

a. If it hits the threshold, resample N_s particles, set $\tilde{x}_n^{(i)} = \tilde{x}_n^{(k)}$ and $\widetilde{w}_n^{(i)} = \frac{1}{N_s}$.

b. Else, set $\tilde{x}_n^{(i)} = \tilde{x}_n^{(i)}$, $\widetilde{w}_n^{(i)} = \widetilde{w}_n^{(i)}$.

(5) Set $x_n = \left\{x_n^{(i)}, w_n^{(i)}\right\}_{i=1}^{N_s}$.

[Step 3]: Estimation

Estimate the distribution of the posterior state via $p(x_n|y_{1:n}) \approx \sum_{i=1}^{N_s} w_n^{(i)} \cdot \delta\left(x_{1:n} - x_{1:n}^{(i)}\right)$.

Figure 3: Operations of a generic particle filter

Performances of these chosen samples are evaluated in terms of their approximations to the posterior distribution. Then, a threshold check for resampling is conducted. Finally, the posterior distribution is estimated via the samples as well as their corresponding normalized importance weights $w_n^{(i)}$ in a delta function δ centered at $x_{1:n}^{(i)}$.

The density function $p(x_n|x_{n-1})$, which is called the transition prior, denotes the probability density associated with the state evolution process (moving from x_{n-1} to x_n), while the density function $p(y_n|x_n)$ represents the likelihood, indicating the distribution of the observed data conditional to the state variables.

More specifically, the particle filtering algorithm is structurally formed with two major steps, which are importance sampling and resampling, respectively. Importance sampling, by definition is estimating the properties of the desired distribution using an alternative distribution to generate samples rather than using the distribution of interest. The performance of the particle filters depends on samples drawn from the importance sampling step, making the selection of this alternative distribution (importance density) crucial. According to Pierre and Doucet's work (2007), the optimal importance density in terms of minimizing the variance of the importance weights is defined as depicted in (3.3), and the corresponding unnormalized importance weights of particles can be calculated via (3.4).

$$q_{opt}(x) = p(\tilde{x}_n|x_{n-1}, y_n) \tag{3.3}$$

$$w_n^{(i)} = w_{n-1}^{(i)} \int p(y_n|\tilde{x}_n^{(i)}) p(\tilde{x}_n^{(i)}|\tilde{x}_{n-1}^{(i)}) d\tilde{x}_n^{(i)} \tag{3.4}$$

However, given these equations, the PF algorithm faces two challenges. First, $q(\cdot)_{opt}$ is not known or very difficult to draw samples from in most of the real-world Bayesian estimation problems. Second, solving the integral $w_n^{(i)}$ is either impossible or takes significant computational resources. To resolve these limitations, the most widely used strategy is to draw samples from the transition prior, as shown in (3.5). Meanwhile,

the recursive update of the particles' importance weights can be simplified as given in (3.6).

$$\tilde{x}_n \sim p(\tilde{x}_n | x_{n-1}) \tag{3.5}$$

$$\widetilde{w}_n^{(i)} \propto w_{n-1}^{(i)} \frac{p\left(y_n \middle| \tilde{x}_n^{(i)}\right) p\left(\tilde{x}_n^{(i)} \middle| x_{n-1}^{(i)}\right)}{q\left(\tilde{x}_n^{(i)} \middle| x_{n-1}^{(i)}, y_n\right)} = w_{n-1}^{(i)} \frac{p\left(y_n \middle| \tilde{x}_n^{(i)}\right) p\left(\tilde{x}_n^{(i)} \middle| x_{n-1}^{(i)}\right)}{p\left(\tilde{x}_n^{(i)} \middle| x_{n-1}^{(i)}\right)} = w_{n-1}^{(i)} p\left(y_n \middle| \tilde{x}_n^{(i)}\right) \tag{3.6}$$

This sampling strategy makes it easier to execute the PF algorithm. However, it results in a new problem as it ignores the most recent measurement y_n. In particular, as iteration processes, all of the mass may be concentrated on a few random samples with most of the particles having negligible weights, causing a potential degeneracy problem. As shown in Figure 4(a), when new measurements appear in the tail of the prior or when the likelihood is too narrow compared to its prior (having a measurement noise with small variance), most of the particles generated from the transition prior are located in the low-likelihood area (far away from the peak of the likelihood). Here, the importance weights of samples vary greatly as n increases and are concentrated in a few particles in a short period of time. However, if the importance density is selected taking both the prior and likelihood into consideration, as shown in Figure 4(b), particles' importance weights vary slightly through iterations. Consequently, the degeneracy problem happens much later in time than that in Figure 4(a).

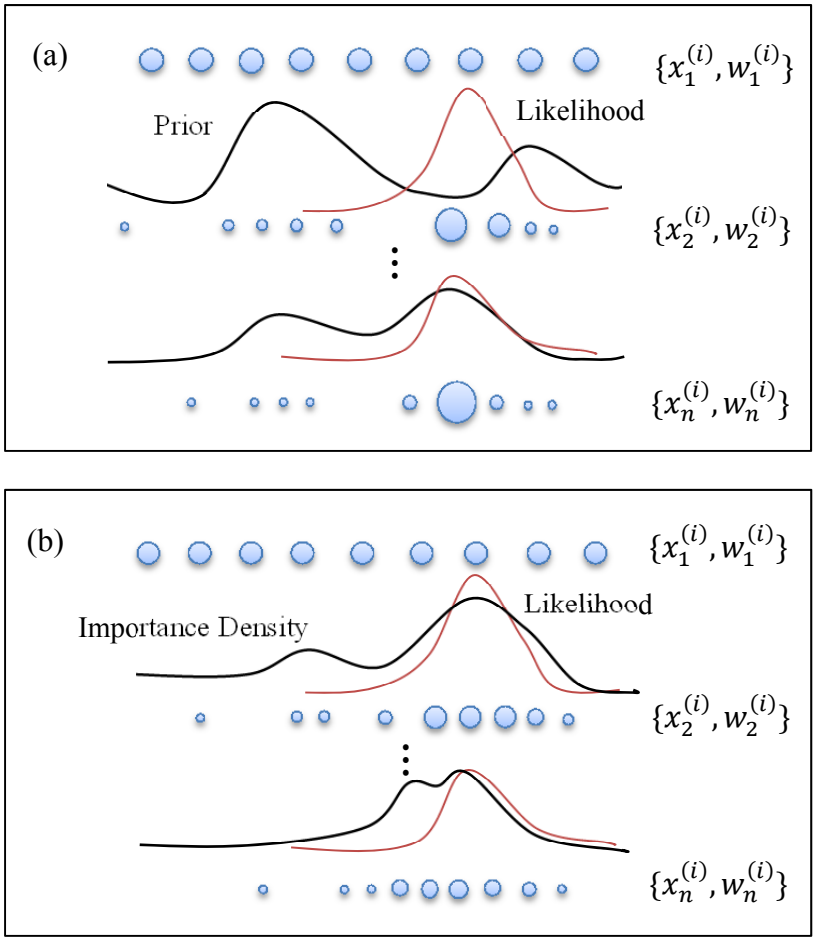


Figure 4: Exemplary degeneracy situation in two different importance densities

When the degeneracy problem is detected, the resampling step is triggered to update the particle trajectories, in which repeated samples are drawn from subsets of available data. Mathematically, suppose at time n, we have a set of samples $x_n = \left\{x_n^{(i)}, w_n^{(i)}\right\}_{i=1}^{N_s}$ properly weighted with respect to the density $q(\cdot)$ where the coefficient of variation of importance weights for $x_n^{(i)}$ is $cv_n^2 = var(w_n^{(i)})/E^2(w_n^{(i)})$ (Gordon et al, 1993). Then, the effective sample size, calculated as $N_e = N_s/(1 + cv_n^2)$, is used to determine the threshold for resampling, representing the equivalent number of random samples at time n. Particularly, once N_e at time n drops below the threshold ($N_e \leq$

$\lambda N_s, 0 < \lambda < 1)$, resampling is triggered. While resampling may be considered as a counter measure for degeneracy problems, it limits the parallelization of the algorithm and leads to the loss of diversity in samples (Doucet and Johansen, 2008; Celik and Son, 2011).

To this end, my goal is to investigate a good choice of importance density considering both the prior and the likelihood that can decelerate the degeneracy while, at the same time, improve the estimation accuracy of particle filters.

3.2. Proposed Density Selection Scheme based on the Minimum Relative Entropy

In this study, the relative entropy is utilized as a measurement standard for the selection of a good importance density. Details of the minimum relative entropy principle and the deviation of the proposed density selection scheme are provided in the following subsections.

3.2.1. Relative Entropy

In probability theory, relative entropy (also known as Kullback-Leibler divergence) is defined as a non-symmetric measure of the difference between two probability distributions. The formula of relative entropy is defined in (3.7) where F and Q represent two different probability distributions of a continuous random variable, $f(x)$ and $q(x)$ denote the densities of F and Q, respectively.

$$D_{KL}(F||Q) = \int_{-\infty}^{\infty} f(x) ln(f(x)/q(x))\, dx \tag{3.7}$$

Given the recursive specifications of target ($p(x_{1:n}|y_{1:n})$) and importance densities ($q(x_{1:n}|y_{1:n})$) as shown in (3.8)-(3.9), particles' importance weights are also updated recursively, as given in (3.10)-(3.11). To this end, the ratio of weights within two iterations can be presented via (3.12) when samples are drawn from the density function $q(x_n|x_{n-1}, y_n)$.

$$p(x_{1:n}|y_{1:n}) \propto g(y_n|x_n)f(x_n|x_{n-1})p(x_{1:n-1}|y_{1:n-1}) \tag{3.8}$$

$$q(x_{1:n}|y_{1:n}) \propto q(x_n|x_{n-1}, y_n)q(x_{1:n-1}|y_{1:n-1}) \tag{3.9}$$

$$w_n^{(i)} = \frac{p\left(x_{1:n}^{(i)}\middle|y_{1:n}\right)}{q\left(x_{1:n}^{(i)}\middle|y_{1:n}\right)} = \frac{p\left(y_n\middle|x_n^{(i)}\right)p\left(x_n^{(i)}\middle|x_{n-1}^{(i)}\right)}{q\left(x_n^{(i)}\middle|x_{n-1}^{(i)},y_n\right)}\frac{p\left(x_{1:n-1}^{(i)}\middle|y_{1:n-1}\right)}{q\left(x_{1:n-1}^{(i)}\middle|y_{1:n-1}\right)} \tag{3.10}$$

$$w_n^{(i)} \propto \frac{p\left(y_n\middle|x_n^{(i)}\right)p\left(x_n^{(i)}\middle|x_{n-1}^{(i)}\right)}{q\left(x_n^{(i)}\middle|x_{n-1}^{(i)},y_n\right)}w_{n-1}^{(i)} \tag{3.11}$$

$$r = \frac{w_n}{w_{n-1}} = \frac{p(y_n|x_n)p(x_n|x_{n-1})}{q(x_n|x_{n-1},y_n)} \tag{3.12}$$

According to Moral and Doucet (2006), the optimal sampling strategy may provide the minimum variance among particles' importance weights. Since particles' importance weights are assigned uniformly (i.e., $w_0^{(i)} = 1/N_s$) upon initialization, the variation in weights would not change dramatically if the ratio r given in (3.12) keeps close to 1 through iterations. From this perspective, if a slight difference is shown between a density function and the product of the transition prior and the likelihood functions (i.e., $p(y_n|x_n)p(x_n|x_{n-1})$), this density can be treated as a good choice of importance density to draw sample from.

To this end, we define $f(x)$ as $f(x) = p(y_n|x_n)p(x_n|x_{n-1})$, assuming the transition prior and the likelihood are independent. Although relative entropy is a non-symmetric measure, meaning that different results may be obtained if the sequences of the probability functions are changed, we seek to minimize the relative entropy between the density function $f(x)$ and the importance density $q(\cdot)$ (instead of minimizing the relative entropy between the importance density $q(\cdot)$ and the density function $f(x)$) in this study for two reasons. On one hand, this definition follows the same structure of how the particles' importance weights are calculated (see equations (3.10)-(3.12)). On the other hand, this definition simplifies the derivation. As $q(\cdot)$ is the target probability density to be approximated, with this definition, there is only one variable contained in the objective function. The proof to obtain a closed form formula of the relative entropy between $q(x)$ and $f(x)$ is provided in Appendix I. It is noticed that the mathematic expression of the obtained $q(x)$ is extremely complex, making it difficult to draw samples from.

In the following subsections, we provide the Taylor series expansion of relative entropy to obtain a closed form that can be easily minimized.

3.2.2. Closed Form Representation of Relative Entropy using Theorem of Taylor Series Expansion

In this subsection, the proof to obtain a closed form formula of the relative entropy between $f(x)$ and $q(x)$ is provided using the theorem of Taylor series expansion. Following the same notation provided in Figure 3, a random vector $x_n = \{x_n^1, x_n^2, \cdots, x_n^{(i)}, \cdots, x_n^{N_s}\}$ with new samples of size N_s is considered at time n.

However, throughout this section, only the random vector $x = \{x^1, x^2, \cdots, x^i \cdots, x^{N_s}\}$ is used to represent the samples obtained in any iteration n to facilitate explanation by hiding the index n. Moreover, to simplify the derivation, the function of relative entropy between $f(x)$ and $q(x)$ is presented as $g(\omega, \eta) = \omega ln(\omega/\eta)$ by treating the density functions $f(x)$ and $q(x)$ as variables ω and η. Then, the first and second derivatives of $g(\omega, \eta)$ can be provided as in (3.13)-(3.14), respectively.

$$g_\omega(\omega, \eta) = \frac{\partial g}{\partial \omega} = 1 + ln\omega - ln\eta,\ g_\eta(\omega, \eta) = \frac{\partial g}{\partial \eta} = -\frac{\omega}{\eta} \tag{3.13}$$

$$g_{\omega\omega}(\omega, \eta) = \frac{1}{\omega},\ g_{\omega\eta}(\omega, \eta) = -\frac{1}{\eta},\ g_{\eta\eta}(\omega, \eta) = \frac{\omega}{\eta^2} \tag{3.14}$$

The Taylor series expansion of $g(\omega, \eta)$ for the neighborhood of point (a, b) can be written as in (3.15),

$$\begin{aligned} g(\omega, \eta) = g(a, b) &+ \frac{1}{1!}\left[d\omega g_\omega(a, b) + d\eta g_\eta(a, b)\right] \\ &+ \frac{1}{2!}\left[(d\omega)^2 g_{\omega\omega}(a, b) + 2g_{\omega\eta}(a, b)d\omega d\eta + (d\eta)^2 g_{\eta\eta}(a, b)\right] \\ &+ \cdots + \frac{1}{n!}\left[\sum_{k=0}^{n}\binom{n}{k}\frac{\partial^n g}{\partial \omega^{n-k}\partial \eta^k}|_{(a,b)}(d\omega)^{n-k}(d\eta)^k\right] \end{aligned} \tag{3.15}$$

where $d\omega = \omega - a$, $d\eta = \eta - b$.

At this point, by substituting (3.13) and (3.14) in (3.15), the closed form for the $g(\omega, \eta)$ around the neighborhood of the point (a, b) can be written as in (3.16), where ε denotes the remainder after evaluating the second order of Taylor Polynomial terms.

$$g(\omega,\eta) = aln(a/b) + (1 + lna - lnb)(\omega - a) + \left(-\frac{a}{b}\right)(\eta - b)$$
$$+\frac{1}{2!}[\frac{1}{a}(\omega - a)^2 + \left(-\frac{2}{b}\right)(\omega - a)(\eta - b) + \frac{a}{b^2}(\eta - b)^2] + \varepsilon$$
$$= \omega + \omega lna - \omega lnb - \frac{a}{b}\eta + \frac{1}{2a}\omega^2 - \frac{1}{b}\omega\eta + \frac{a}{2b^2}\eta^2 + \varepsilon \quad (3.16)$$

Finally, by assuming that the higher order of partial derivatives and the Lagrange remainder are insignificant (i.e. $\varepsilon \approx 0$), the closed form of the relative entropy between $f(x)$ and $q(x)$ can be represented as in (3.17) by replacing the neighborhood of point (a, b) with $(\mu(x_0)\ ,\vartheta(x_0))$ and variables ω and η with the density functions $f(x)$ and $q(x)$. Here, $\mu(x_0)$ represents the arbitrary initial density function and x are the samples drawn from $q(x)$. The determination of the density function $\vartheta(x_0)$ that depends on the initial density $\mu(x_0)$ will be discussed later in the following subsection.

$$D_{KL}(F||Q) \approx \sum_x f(x)[1 + ln\frac{\mu(x_0)}{\vartheta(x_0)} - \frac{q(x)}{\vartheta(x_0)}] - \sum_x \left[\frac{\mu(x_0)q(x)}{\vartheta(x_0)} - \frac{f(x)^2}{2\mu(x_0)} - \frac{\mu(x_0)q(x)^2}{2\vartheta(x_0)^2}\right] \quad (3.17)$$

3.2.3. Minimization of Relative Entropy

Given the fact that the minimum relative entropy between two probability distributions should be zero if and only if the two probability distribution functions are the same, the equation indicating the approximated relative entropy between $f(x)$ and $q(x)$ (i.e., the equation shown in (3.17)) is set to zero, where the insignificant ε is ignored. In this way, the minimization of the relative entropy between $f(x)$ and $q(x)$ is achieved, thereby the approximated form of importance density $q(x)$ can be obtained.

Mathematically, by solving the equation (3.18), the $q(x)$ can be presented as a function of $f(x)$, $\vartheta(x_0)$, and the initial density $\mu(x_0)$ as shown in (3.19).

$$\frac{f(x)^2}{2\mu(x_0)} + f(x)\left(1 + ln\frac{\mu(x_0)}{\vartheta(x_0)} - \frac{q(x)}{\vartheta(x_0)}\right) - \frac{\mu(x_0)q(x)}{\vartheta(x_0)} + \frac{\mu(x_0)q(x)^2}{2\vartheta(x_0)^2} = 0 \tag{3.18}$$

$$q(x) = \frac{\frac{\mu(x_0)+f(x)}{\vartheta(x_0)} \pm \sqrt{\left(\frac{\mu(x_0)+f(x)}{\vartheta(x_0)}\right)^2 - \frac{2\mu(x_0)}{\vartheta(x_0)^2}\left[f(x)(1+ln\frac{\mu(x_0)}{\vartheta(x_0)}+\frac{f(x)}{2\mu(x_0)}\right]}}{\frac{\mu(x_0)}{\vartheta(x_0)^2}} \tag{3.19}$$

In order to guarantee a unique solution for the importance density $q(x)$, the discriminant in the quadratic shown in (3.19) is set to zero as given in (3.20). Then, we can obtain $\vartheta(x_0) = \mu(x_0)e^{-\mu(x_0)/[2f(x)]}$. Finally, the probability density function $q(x)$ that shows slight difference from $f(x)$ is provided in (3.21), and the importance density function $q(x_n|x_{n-1}, y_n)$, which is embedded in the particle filtering algorithm, can be obtained via substituting $f(x) = p(x_n|x_{n-1})p(y_n|x_n)$ in (3.21) as presented in (3.22).

$$\left(\frac{\mu(x_0)}{\vartheta(x_0)} + \frac{f(x)}{\vartheta(x_0)}\right)^2 - \frac{2\mu(x_0)}{\vartheta(x_0)^2}\left(f(x)(1 + ln\frac{\mu(x_0)}{\vartheta(x_0)} + \frac{f(x)}{2\mu(x_0)}\right) = 0 \tag{3.20}$$

$$q(x) = [\mu(x_0) + f(x)]e^{-\frac{\mu(x_0)}{2f(x)}} \tag{3.21}$$

$$q(x) = [\mu(x_0) + p(x_n|x_{n-1})p(y_n|x_n)\,]e^{-\frac{\mu(x_0)}{2p(x_n|x_{n-1})p(y_n|x_n)}} \tag{3.22}$$

In the next section, the validity of the proposed density selection scheme is demonstrated, and the performance of filters embedding the proposed density selection scheme is benchmarked against that of particle filters embedding other density selection schemes in terms of their estimation qualities and computational efficiencies.

3.3. Synthetic Experiments and Results

In this section, the feasibility and validity of the proposed minimum relative entropy-based importance density selection scheme is demonstrated via two different sets of synthetic experiments. In the first set of experiments, the performance of the proposed density selection rule is tested when it is not embedded into a particle filter, in which importance density $q(x)$ is obtained based on the (3.21) with a given $f(x)$. In the second set of experiments, the particle filter embedding with the proposed importance density selection scheme is applied to a synthetic state-space model. Results obtained are presented and discussed in detail.

3.3.1. Validation of the Proposed Rule When Not Embedded into a Particle Filter

The importance density selection rule is proposed based on the fact that smaller variances may be obtained in importance weights when the ratio of weights within two iterations grows closer to one. In other words, based on (3.10)-(3.12), the performance of the proposed selection scheme improves as the difference between $q(x)$ and $f(x)$ decreases. From this perspective, the proposed scheme is expected to perform better when the function $q(x)$ obtained via (3.21) is similar to that of $f(x)$ in their major characteristics (i.e., shape, output ranges, etc.). In order to validate the proposed density selection rule, two synthetic experiments are carried out as the following. Both of the simulations are conducted using Matlab R2010b with ten replications.

Experiment 1: $f(x)$ is a standard normal probability density

In this case, $f(x)$ is selected as a standard normal density. The results are shown in Figure 5, where (a) is the comparison between $f(x)$ and alternative $q(x)$ obtained

through the experiments based on (3.21) for 10 simulations, and (b) is the comparison between $f(x)$ and the average of different $q(x)$'s obtained for each simulation. The resultant $q(x)$ consistently presents a high degree of similarity in shape and output ranges with the given $f(x)$.

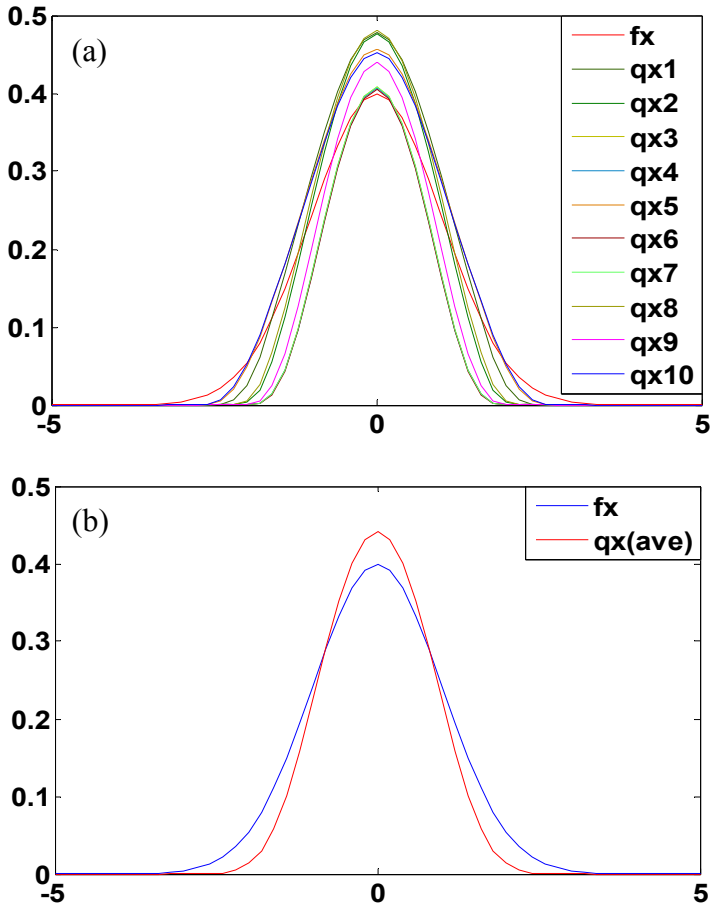


Figure 5: Simulation results of experiment 1

Experiment 2: $f(x)$ is a self-defined complex density

In this case, the distribution density $f(x)$ is defined as a complex function as shown in (3.23), where a, b, c, and d are arbitrarily selected to be -0.02, 2, 5.85e-4, and 0.02, respectively.

$$f(x) = \begin{cases} -a\,sin(bx) + cx^2 + d, & -10 \le x \le 10 \\ 0, & otherwise \end{cases} \tag{3.23}$$

Similar to the previous experiment, simulations were conducted using (3.21) to obtain the density function $q(x)$. All $q(x)$ obtained from the ten replications present high similarity with $f(x)$, and only minor differences are observed between $f(x)$ and the average of various $q(x)$'s obtained for each simulation, as shown in Figure 6 (a) and (b), respectively.

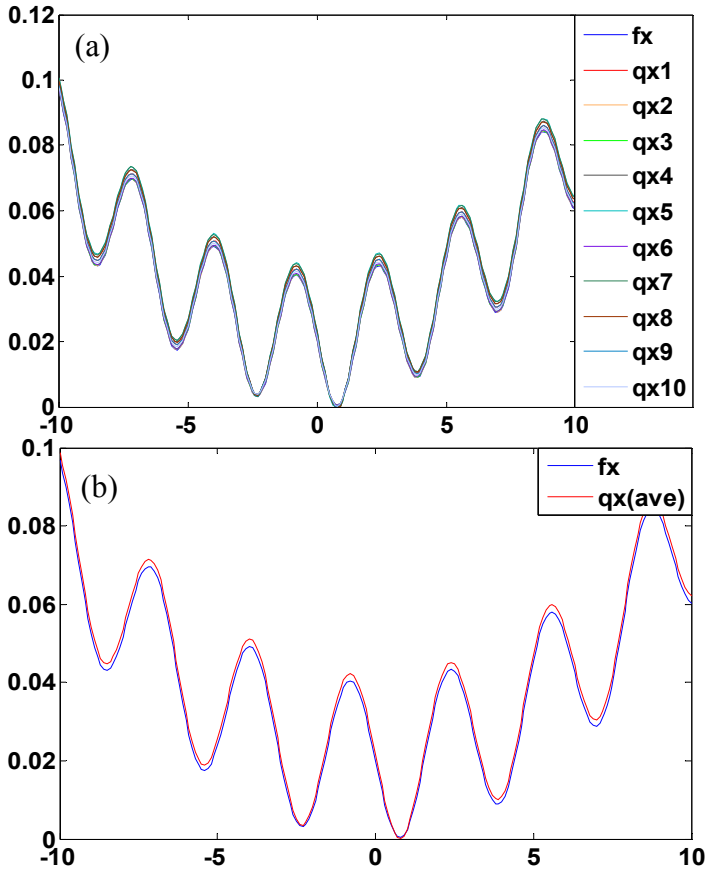


Figure 6: Simulation results of experiment 2

In the next subsection, the proposed density selection scheme is incorporated into a particle filter, to estimate the posterior states of a system, whose performance is given by a synthetic state-space model.

3.3.2. Validation of the Proposed Rule When Embedded into a Particle Filter

In this subsection, the performance of the particle filter embedding the proposed density selection scheme (MREIS-PF) is tested against that of the prior-sampling based particle filter (PRIOR-PF), the Gaussian Sum Particle Filter (GSPF), and the Sigma-Point Particle Filter (SPPF), in terms of their estimation accuracies and computational efforts in a non-linear and non-Gaussian system setting whose posterior distribution (dynamic state status) is known. Here, the estimation accuracies are measured via prediction errors, which are defined as the differences between true values and estimated values. Among the prediction-error metrics, the root mean square error (RMSE) statistic is usually more advantageous than the others (mean absolute error, mean absolute percentage error, etc.) in terms of the sensitivity, as the squaring process gives disproportionate weight to occasional large errors, which might occur in the considered large scale and complex system within a short time interval (Celik and Son, 2011). Therefore, in this study, the RMSEs between the estimation results obtained via these filters and true states are used as an evaluation mechanism where lower prediction error indicates a better estimation.

As stated earlier in Section 3.1, the PF algorithm estimates the posterior distribution via a set of weighted samples drawn from a proposal distribution based on the measurements fed into the algorithm. In order to capture the behavior of the synthetic

system considered in this study, a set of nonlinear state space equations is composed, as given in (3.24) and (3.25), respectively,

$$x_{k+T} = f_x(x_k, v_k) = cos(a^2\pi k) + \frac{8a^3}{1+5a^2} + bx_k + u_k \tag{3.24}$$

$$y_k = g_y(x_k, h_k) = bx_k^2 + v_k \tag{3.25}$$

where a and b are constants that determine the amplitude of the state evolution and observation functions. Process noises u_k and observation noise v_k are generated from Gamma and Gaussian distributions, respectively. Here, the threshold value for the resampling procedure should be determined with care. If a large number is set for this value, the resampling process may repeat in each iteration with regenerating all samples and equally assigning all particles' importance weights. In this case, the recursive importance sampling procedure lost its control of the performance of the particle filtering algorithm and different choices of the importance density function do not affect the estimation accuracy. On the other hand, if a very small value is selected, the particles may prohibit the resampling process and the algorithm may cease to provide accurate results due to the consequent degeneracy problems. In our experiments, the resampling threshold is set to be $0.5N_s$ (N_s represents the number of particles) as an empirical value in order to avoid both extremes.

Figure 7 shows the screenshots of running algorithms for a single replication where blue lines with star markers represent the true states obtained from the original functions and red lines with circle markers represent the estimated posterior states obtained from MREIS-PFs (a), PRIOR-PFs (b), SPPFs (c), and GSPF (d), respectively. Results of the

state estimation obtained from the particle filters embedding different density selection schemes have been depicted via the presented plots.

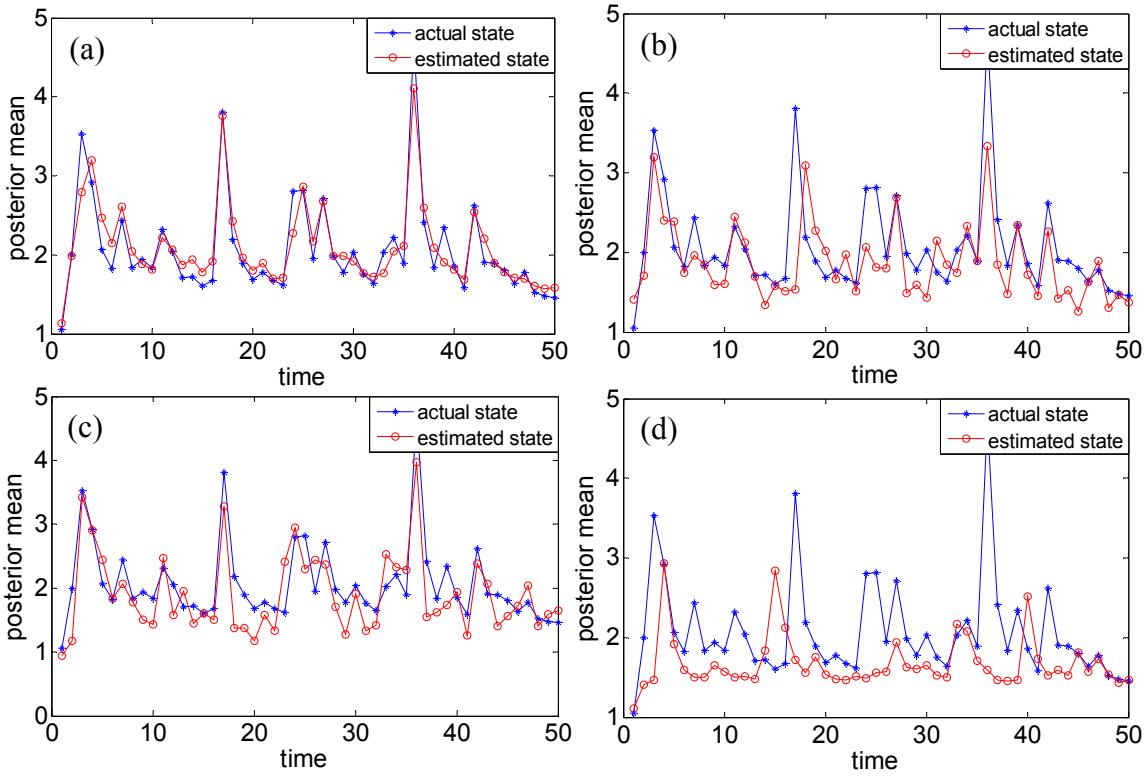


Figure 7: Estimated posterior states (obtained from particle filters embedding different density selection schemes) versus the actual posterior states

In order to validate the consistency of the results, the same sets of experiments have been carried out for 100 replications each lasting 50 time steps, and the initial sample size is set to 200 particles. Overall, the results shown in Table 4 demonstrate that the MREIS-PF provides better estimation accuracy than that of other PFs under parametric variations of the system's dynamic model. It is noted that for all the cases where process noise are generated from Gamma (0.3, 0.5), generating the observation noise from a Gaussian distribution with a larger variance produces a negative impact on the estimation accuracies. Similarly, for the cases where observation noise are generated from Gaussian

(0.2, 0.1), the performances of the particle filters are also degraded when the incorporated process noise is generated from a Gamma distribution with a larger rate parameter β. However, while they both have a considerable impact on the performance of the particle filters, neither the observation (likelihood) nor the previous state (prior) imposes a restriction on the application of the filter with the proposed minimum relative entropy-based importance density selection scheme.

Table 4: Estimation results obtained from the particle filters embedding different densities under parametric variations

		process noise ~ Gamma(0.3,0.5)		
		observation noise ~ Gaussian (0.2, 0.01)	observation noise ~ Gaussian (0.2, 0.1)	observation noise ~ Gaussian (0.2, 1)
MREIS-PF	RMSE-mean	0.1256	0.1794	0.2810
	RMSE-var	0.0069	0.0006	0.0017
PRIOR-PF	RMSE-mean	0.2368	0.2785	0.3606
	RMSE-var	0.0035	0.0025	0.0079
SPPF	RMSE-mean	0.1618	0.3341	0.3486
	RMSE-var	0.0019	0.0084	0.0086
GSPF	RMSE-mean	0.2760	0.3345	0.4169
	RMSE-var	0.0109	0.0091	0.0193
		observation noise ~ Gaussian(0.2,0.1)		
		process noise ~ Gamma (0.3,0.05)	process noise ~ Gamma (0.3, 0.1)	process noise ~ Gamma (0.3, 0.5)
MREIS-PF	RMSE-mean	0.0724	0.0880	0.1794
	RMSE-var	2.85e-5	6.52e-5	0.0006
PRIOR-PF	RMSE-mean	0.1695	0.1825	0.2785
	RMSE-var	0.0001	0.0004	0.0025
SPPF	RMSE-mean	0.3308	0.3307	0.3341
	RMSE-var	0.0104	0.0115	0.0084
GSPF	RMSE-mean	0.1737	0.1847	0.3345
	RMSE-var	0.0001	0.0004	0.0091

Table 5 shows the extended results obtained from the MREIS-PF versus other PFs in different experiment settings (different number of runs and different initial sample sizes). The process and observation noises are generated from Gamma (0.3, 0.5), and

Gaussian (0.2, 0.1) distributions, respectively. It is noticed that the increased number of simulation runs results in a very slight change in terms of both the RMSE-mean and RMSE-var values, regardless of the selected particle filters. Consequently, the proposed selection scheme for importance densities can be employed with a small number of samples in order to reduce the computational efforts (elapsed time during the estimation) while still achieving high estimation accuracies.

Table 5: Extended results obtained from particle filters embedding different densities in different experiment settings

Initial sample size	Number of simulation runs	MREIS-PF		PRIOR-PF		SPPF		GSPF	
		RMSE-mean	RMSE-var	RMSE-mean	RMSE-var	RMSE-mean	RMSE-var	RMSE-mean	RMSE-var
100	50	0.2660	0.0026	0.2818	0.0058	0.3342	0.0096	0.3423	0.0139
	100	0.2550	0.0036	0.2885	0.0061	0.3271	0.0087	0.3383	0.0076
	200	0.2316	0.0036	0.2843	0.0046	0.3090	0.0073	0.3468	0.0016
	400	0.1845	0.0007	0.2769	0.0036	0.2795	0.0015	0.3429	0.0099
200	50	0.1811	0.0005	0.2765	0.0032	0.2783	0.0016	0.3456	0.0107
	100	0.1788	0.0008	0.2757	0.0033	0.2789	0.0015	0.3453	0.0102
	200	0.1796	0.0007	0.2753	0.0031	0.2790	0.0016	0.3488	0.0111
	400	0.1813	0.0007	0.2710	0.0036	0.2789	0.0013	0.3409	0.0097
400	50	0.1809	0.0006	0.2709	0.0040	0.2787	0.0016	0.3379	0.0105
	100	0.1809	0.0007	0.2716	0.0035	0.2797	0.0017	0.3329	0.0096
	200	0.1779	0.0005	0.2704	0.0045	0.2757	0.0011	0.3411	0.0104
	400	0.1777	0.0005	0.2717	0.0031	0.2788	0.0016	0.3416	0.0114
800	50	0.1784	0.0007	0.2703	0.0030	0.2867	0.0015	0.3429	0.0118
	100	0.1771	0.0005	0.2712	0.0028	0.2778	0.0010	0.3433	0.0107
	200	0.1777	0.0006	0.2695	0.0030	0.2730	0.0022	0.3441	0.0095
	400	0.1772	0.0006	0.2649	0.0021	0.2779	0.0016	0.3425	0.0099

Results obtained from the synthetic experiments designed in this section strongly support the feasibility and validity of the proposed minimum relative entropy-based importance density selection scheme. In the next section, the validity of the PFs embedding our proposed density selection scheme is demonstrated via a case study

focusing on the problem of short-term load demand forecasting for a company located in Miami, FL.

3.4. Case Study I: Short-term Electric Power Load Forecasting

Commercial customers use considerably more electric power than residential consumers and are generally charged with extra fees for reservation to meet their demands during peak time (Florida Power & Light, 2013). Hence, utility costs in terms of the energy consumption account for a great portion of the total monthly expenditure in large-scale enterprises. In general, costs associated with energy consumption represent about 19% of total expenditures for a typical office building (E Source, 2002), and 25% of total operating costs for a large facility (Figy, 2011). In order to cut down from these significant expenses, large-scale enterprises may shop for electricity with the most favorable type of rates and payment schemes from retail electricity suppliers. In addition to diverse purchase packages with different rate schedules, utility companies offer a relatively new option for large-scale enterprises since the introduction of the distributed power generation systems. Within this new option, large-scale enterprises may choose to generate electricity using on-site generators installed near or at their consuming points by utility companies instead of purchasing the grid-supplied power. Execution of this new supply pattern provides various advantages. First, it may relieve the congestion on power lines during periods of peak demand since facilities which need large amount and highly reliable power reduce their consumption of grid-supplied power via using their on-site generators. Second, large-scale enterprises may find it economically advantageous to operate their own power generation system especially when installation costs, rents of

generators, and electricity production costs are less than purchasing costs of the grid-supplied power. However, determination of the lowest cost method of providing the required amount of electricity given the organization's production schedules relies heavily on accurate load forecasts in the forthcoming periods. On the one hand, overestimation of demands may result in financial losses due to company-owned generators being on standby while not needed, and may also lead to high production costs or even an inaccurate decision-making on the selection of the distributed generators in terms of their capacities. On the other hand, underestimation of the demands may lead to costly startup of cold generators or, in the worst case scenario, may cause blackouts if the frequency drops from the required levels. From this perspective, in this section, we select the short-term load demand forecasting problem as a case study to illustrate the feasibility of the particle filters that embed our proposed density selection scheme when applied it to a real-world state estimation problem in large-scale dynamic systems.

Short-term load demand forecasting is a stochastic process as a function of time which has been widely studied in the literature especially as part of the control and scheduling problems existent in the power systems. The main approaches used in these works can be categorized as auto-regressive integrated moving average (ARIMA) models (Hagan and Behr, 1987; Juberias et al., 1999), artificial neural networks (ANN) (Lamedica et al, 1996), regression (Pardo et al., 2002), exponential smoothing (Harvey and Koopman, 1993), state-space modeling (Gastaldi et al., 2004) or a combination of these. The ARIMA modeling and exponential smoothing are easy to implement while limitations are encountered on their applications to nonlinear and multivariate problems. Regression makes it possible to take all the explanatory variables and their impacts into

consideration. However, its performance relies heavily on the parameter estimation, and the computational complexity increases substantially when the number of the explanatory variables increases. Furthermore, even though the ANN may provide better estimation accuracies, its execution is always hampered by over-fitting and its highly complex architecture. State-space modeling, on the other hand, not only performs well in both linear and nonlinear problems, but also eases the requirements of computational efforts. To this end, we address the daily electricity demand forecasting problem in our case study by way of developing a state-space model based on a time-varying process (i.e., enterprises' production schedules) as well as external factors (i.e., daily temperature). The model is applied to the demand forecast for a private company located in Florida, which engages in the distribution and marketing of fresh-cut flowers (Bloomberg Business, 2012). The main electrical meter room is shown in Figure 8. Data of the electricity consumption for this plant has been collected from their refrigeration system using HOBO data loggers.

Figure 8: Main electrical meter room at the private company considered in this study

The dataset used in this study concerns the company's hourly electricity consumption from March 28, 2005 until March 12, 2006 with 8,400 hourly records.

Hourly temperature data during the same period are also collected to present temperature-driven impacts (i.e., heating and cooling influences). In this case, the major electricity consumption occurs in the refrigeration system, in which only cooling demands are satisfied. Therefore, we take the cooling effect of the temperature on electricity demands into account and select 64.4°F as the balance point based on the literature (Valor et al., 2001; Mirasgedis et al., 2007). Hence, electricity is needed to cool the system if the outside temperature is above this balance point.

To this end, the state-space model for electricity demand forecasting in our case study is detailed in (3.26) – (3.27),

$$D_{h,d+1} = \alpha_{h,d} D_{h,d} + U_{h,d} \tag{3.26}$$

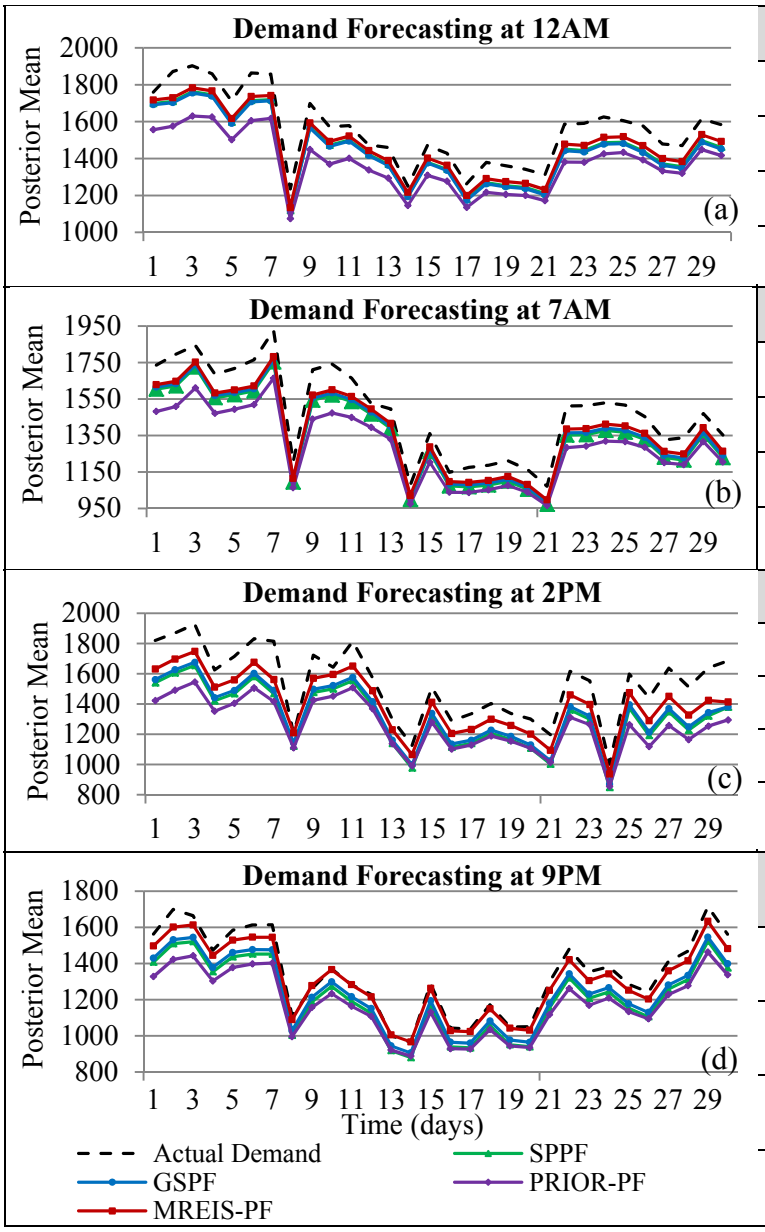
$$T_{h,d} = \beta_{h,d} D_{h,d} + V_{h,d} \tag{3.27}$$

where $D_{h,d}$ is the electricity demand at hour h of day d, measured in kilowatts, $T_{h,d}$ represents the temperature at hour h in day d, $U_{h,d}$ and $V_{h,d}$ denote the process and observation noises, $\alpha_{h,d}$ and $\beta_{h,d}$ are parameters related to the state evolution and observation functions, respectively. It is noted that these functions are updated in a time-series manner as the parameters change hourly in the day. While forecasting results may be obtained for any time spanning the 24-hr period, our analysis focuses on four time quarters, which are 12am, 7am, 2pm, and 9pm, respectively, for ease in representation.

In the experiments conducted in this section, the process and observation noises are assumed to be generated with Gamma (0.5, 0.2) and Gaussian (1, 0.01) distributions, respectively. Table 9 depicts the comparison between actual load demands and estimated

results in terms of the posterior mean under four different time quarters during 30 days. In Figure 6, the black dashed line represents the actual demands while the completed lines with different types of markers represent the estimated results obtained via particle filters embedding different density selection rules. It is noticed that the MREIS-PF outperforms all others in all four experiments conducted in different time quarters.

Table 6: Estimated versus actual daily power demands at different time quarters



PF	RMSE
MREIS-PF	195.8714
PRIOR-PF	207.4293
SPPF	250.5446
GSPF	211.1395

PF	RMSE
MREIS-PF	232.5522
PRIOR-PF	245.8831
SPPF	274.3159
GSPF	239.8054

PF	RMSE
MREIS-PF	279.1310
PRIOR-PF	330.6421
SPPF	356.1440
GSPF	317.3919

PF	RMSE
MREIS-PF	166.9243
PRIOR-PF	206.1956
SPPF	223.8799
GSPF	192.0976

In the following experiments, electricity demands at 9:00AM in a day are estimated. Particle filters are equipped with different size of initial particles to investigate its impacts on the estimation accuracies. As shown in Figure 9, for each level of particle size, the MREIS-PF outperforms the others by achieving the smallest RMSE-mean values. Furthermore, a larger initial sample size slightly improves the estimation accuracy for the PRIOR-PF, while its impact is negligible on the performance of the MREIS-PF, GSPF and SPPF.

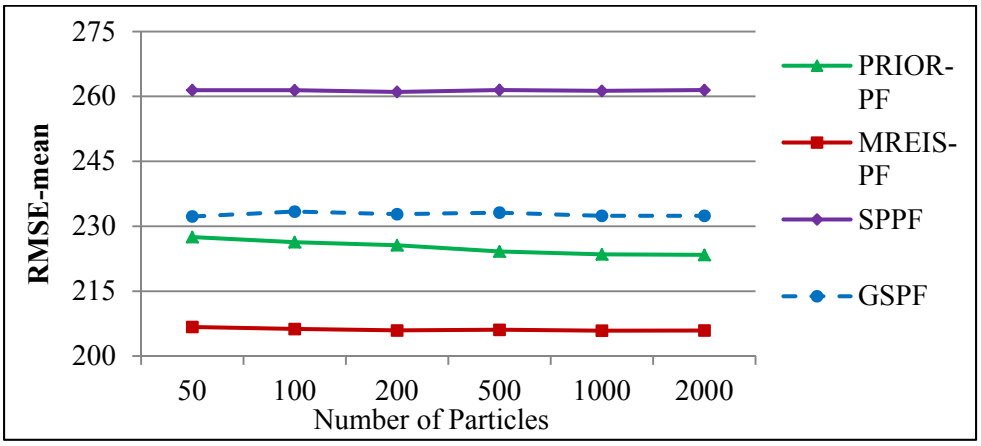


Figure 9: Mean of RMSE values for state estimates (daily power demand forecasts) as a function of the number of particles

Regarding the variance of RMSE values (see Figure 10), it decreases dramatically as the number of particles increases for results obtained via the PRIOR-PF and GSPF. On the other hand, results obtained via the MREIS-PF and SPPF present minor fluctuations on their corresponding RMSE-vars. Therefore, good estimation accuracies may be achieved via the MREIS-PF with a small size of initial particles, while considerably saving from computational efforts, especially when the system under study is very large-scale and possesses a high level of uncertainty.

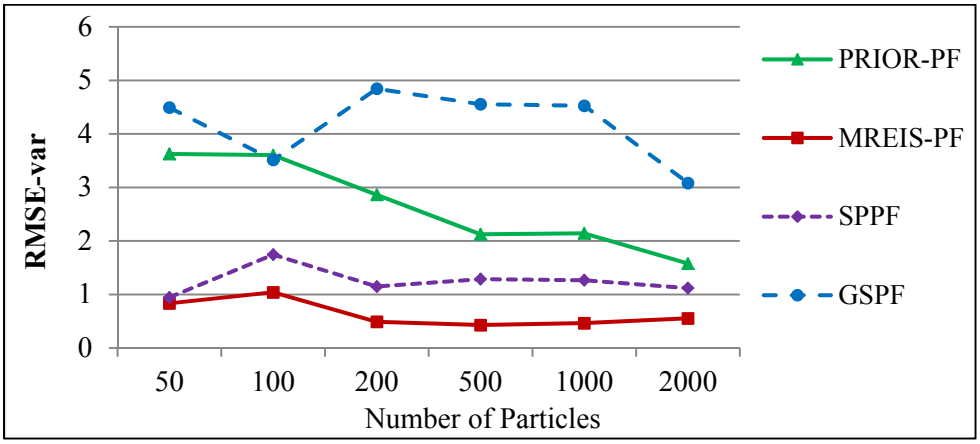


Figure 10: Variance of RMSE values for state estimates (daily power demand forecasts) as a function of the number of particles

In this study, the computational efforts required for implementing the algorithm have been measured via the time that is taken for running the simulation. As shown in Figure 11, the computational efforts required for the execution of particle filters embedding different importance density selection rules are all directly related to the size of the particle sets. Slight differences are presented among the computational efforts for implementing the MREIS-PF, PRIOR-PF and GSPF, which can be attributed to the calculation of the importance weights. For the filters having their priors as importance densities, the weights of the particles can be represented as $w_n = w_{n-1} * p(y_n|x_n)$, while for particle filters embedding the proposed importance density selection scheme, the weights w_n are equal to $w_{n-1} * \frac{p(x_n|x_{n-1})*p(y_n|x_n)}{q(x_n|x_{n-1},y_n)}$ (see (3.11)). However, the computation burden for implementing the SPPF is a polynomial function of the number of particles.

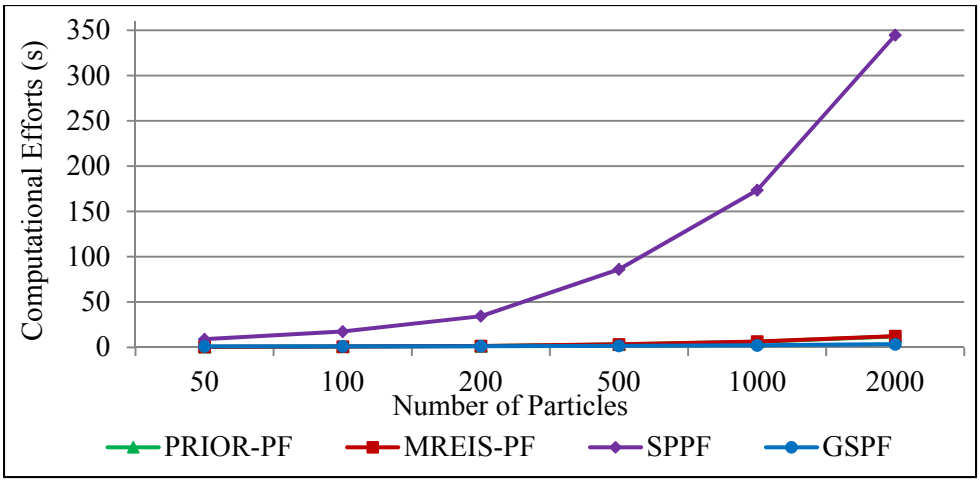


Figure 11: Computational efforts required for different importance density selection rules as a function of the number of particles

Experiments carried out in the case study demonstrate that the proposed importance density selection scheme improves the performance of generic particle filters without increasing their computational resource requirements significantly when applied to the complex and dynamic systems.

Chapter 4: Particle Filtering-based Low Elevation Target Tracking in the Presence of Multipath Interference over the Sea Surface

As mentioned earlier in Section 1.2, Chapter 1, sea surface is constantly perturbed by irregularities from the specular and diffuse reflections, and thereby should be addressed when dealing with multipath interference. In this chapter, we first introduce multipath effects of sea surface, followed by the details of the proposed state space modeling approach including the formulation of state and measurement functions and the embedded particle filtering estimation algorithm. Finally, a set of synthetic experiments are carried out to evaluate the performance of the proposed tracking mechanism under different scenarios.

4.1. Multipath Interference over the Sea Surface

Multipath effects over the sea surface usually contain two components: specular and diffuse reflections. Specular reflection is described as the reflection that occurs over a flat sea-surface, whereas diffuse reflection is the one that occurs under the situation that the returned signal reaches the rough sea-surface but cannot be absorbed or reflected completely. Both of these reflection types are depicted in Figure 12.

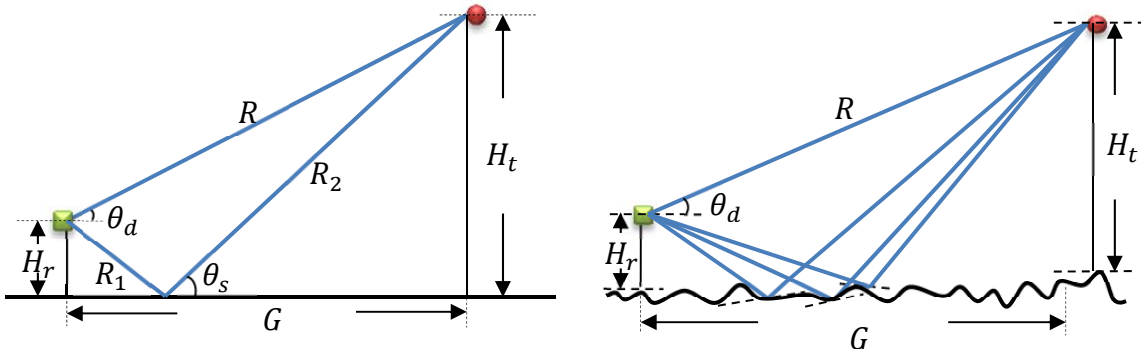


Figure 12: Components of multipath effects over the sea surface

In Figure 12, a target reflects signals in all directions, and radar receives one direct returned signal from an elevation angle θ_d and one specular reflected signal from grazing angle θ_s. Here, a specular reflection coefficient v_s can be expressed as a product of the Fresnel reflection coefficient ρ_0, specular scattering factor ρ_s, and divergence factor D ($v_s = \rho_0 D \rho_s$). Based on Beckmann and Spizzichino's work (1963), ρ_0 is a function of grazing angle and wave form polarization, as given in (4.1),

$$\rho_0 = \begin{cases} \frac{k_c sin\theta_s - \sqrt{k_c - cos^2\theta_s}}{k_c sin\theta_s + \sqrt{k_c - cos^2\theta_s}}, & \text{for vertical polarization} \\ \frac{sin\theta_s - \sqrt{k_c - cos^2\theta_s}}{sin\theta_s + \sqrt{k_c - cos^2\theta_s}}, & \text{for horizontal polarization} \end{cases} \tag{4.1}$$

where $k_c = \frac{k}{k_0} - j60\lambda\delta$, k/k_0 is the relative dielectric constant of the reflecting surface, λ is the radar wavelength, and δ is the conductivity of the sea surface. According to Beard (1961) and Blake (1980), the divergence factor D and specular scattering factor may be expressed as in (4.2) and (4.3), respectively.

$$D \approx \left[1 + \frac{2R_1R_2}{r_e G sin\theta_s}\right]^{-\frac{1}{2}} = \left[1 + \frac{2H_r(Rsin\theta_d + H_r)}{r_e R cos\theta_d sin^3\theta_s}\right]^{-\frac{1}{2}} \tag{4.2}$$

$$\rho_s = exp[-2(2\pi\mu)^2] \tag{4.3}$$

Here, $\mu = \frac{\sigma_h}{\lambda} sin\theta_s$ is the roughness factor, σ_h is the root mean square (RMS) surface height variation which is assumed to be normally distributed in this work, and r_e represents the earth radius. The diffuse reflection coefficient may be expressed as $v_d = \rho_0 \rho_d$, where ρ_d is the RMS diffuse scattering coefficient and ρ_0 has already defined

in (4.1). The scattered signal's phase is incoherent and its fluctuations are Rayleigh distributed. Theoretical deviation for computing the Rayleigh parameter of diffuse reflection coefficient is provided in Blair and Brandt-Pearce's work (2001), as presented in (4.4).

$$\rho_d = \begin{cases} 3.68\rho_0\mu\ , & 0 < \mu < 0.1 \\ (0.454 - 0.858\mu)\rho_0\ , & 0.1 \le \mu < 0.5 \\ 0.025\rho_0\ , & \mu \ge 0.5 \end{cases} \tag{4.4}$$

Particularly, the transmitted signals are always widely scattered about a mean value which varies between 0.3 and 0.35, and the majority of values is included between 0.2 and 0.4. Figure 13 shows the distributions of the values of ρ_s and ρ_d based on a course of various experiments.

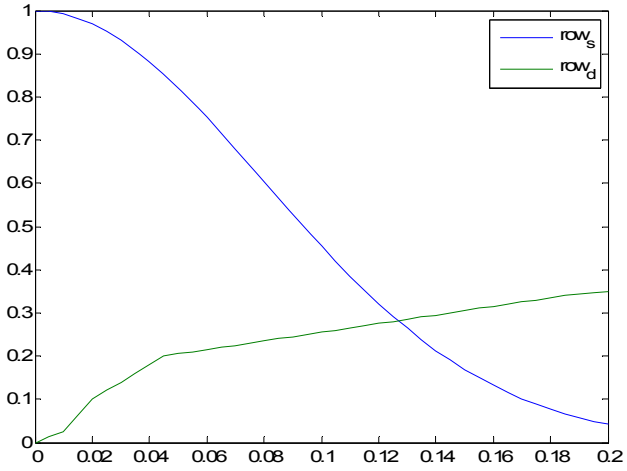


Figure 13: Specular and diffuse coefficients versus sea-surface roughness

In the presence of reflections from the sea surface, the total received signal power can be presented by amplitude of a target echo, antenna voltage gain pattern from a target

angle θ_d and θ_s, and magnitude of reflection coefficient which changes depending on the predominance of the specular and diffuse reflection. In order to determine the surface reflection coefficient, the Rayleigh criterion is conducted here describing the state of the sea surface. Particularly, Δh is defined as the difference in surface heights between two selected points on the sea, and phase shifts of the reflected signals at these two points can be obtained via $\Delta\phi = \frac{4\pi\Delta h sin\theta_s}{\lambda}$. Then, the surface reflection coefficient is presented as given in (4.5).

$$\nu = \begin{cases} \nu_s, & if\ \Delta h < \frac{1}{8}\frac{\lambda}{sin\theta_s} \\ \nu_d, & otherwise \end{cases} \tag{4.5}$$

4.2. Doppler Effects

Tracking mechanism existed in the literature always treated the objects as stationary at the given period by ignoring the Doppler Effects. However, this assumption may lead to a big estimation error, especially for the case that the relative motion between the signal transmitter and receiver is in high-speed. In order to track the target's velocity more precisely, the Doppler Effects are considered in this study. A brief introduction of the Doppler Effects is provided in this section.

Generally, if the object is moving either closer or farther away, there is a slight change in the frequency of the radio waves, as given in (4.6),

$$f = \left(\frac{c+v_r}{c+v_s}\right) f_0 \tag{4.6}$$

where f is the observed frequency of the received impulse, f_0 is the emitted frequency of the transmitter, c is the velocity of the waves in the medium, v_r is the velocity of the receiver relative to the medium, v_s is the velocity of the target relative to the medium. In this study, we assume v_r is zero, c is considered as the speed of the light, then, the relative velocity of the target with respect to the receiver can be obtained via (4.7),

$$v_s = \frac{c(f_0 - f)}{f} \tag{4.7}$$

4.3. Proposed State-Space Models Incorporating Multipath Interference

Provided the underlying information for measurement of a multipath interference, a dynamic range-velocity state-space model for target tracking over the sea surface is established in this study. Specifically, an unobserved state variable x_k is introduced to represent the time-varying dynamics of the tracking target (i.e., range and velocity), where k indicates the time. The posterior target's state is estimated via the previous state x_{k-1} and a series of observations $y_{1:k}$, which is considered as the total power of the received signals recorded by the sensor at predetermined intervals.

Assuming the signal transmitter and receiver of a radar system are located at the origin in the Cartesian coordinates and are both relatively stationary with respect to a target, we represent a target's location $P_k(R_k, \theta_k)$ via the distance between a target and transmitter (i.e., R_k), and direction of a target's movement (i.e., θ_k) (see Figure 13). Then, the state evolution functions in the developed 2D state-space model can be presented as shown in (4.8)-(4.11),

$$R_{k+1} = R_k + v_{k+1}^x \Delta t sin\theta_k + v_{k+1}^y \Delta t cos\theta_k \quad (4.8)$$

$$\theta_{k+1} = \theta_0 + sin(\theta_k + k/a) - k/b + N_k \quad (4.9)$$

$$v_{k+1}^x = \frac{c(f_0 - f_{k+1})}{f_{k+1}} sin\theta_{k+1} \quad (4.10)$$

$$v_{k+1}^y = \frac{c(f_0 - f_{k+1})}{f_{k+1}} cos\theta_{k+1} \quad (4.11)$$

where a and b are functional parameters that determine a target's trajectory, v_k^x and v_k^y denote the corresponding velocities in different directions at time k (i.e., x represents horizontal direction and y represents the vertical direction), Δt is the predetermined time interval, and N_k represents the state noise. Moreover, f_k is the observed frequency of a received impulse at time k, which is used to calculate a target's velocity based on the Doppler Effects.

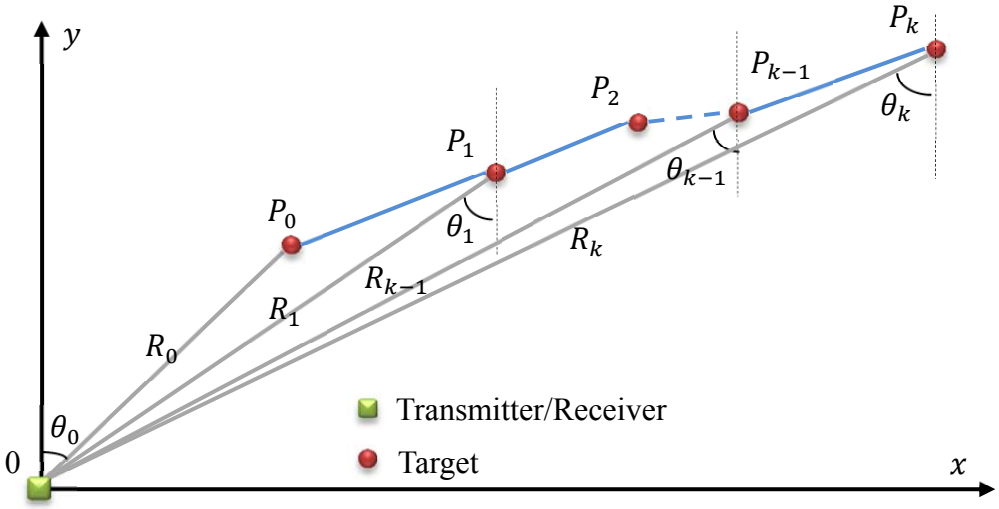


Figure 14: Tracking of a target's location shown in the coordinate system

The measurement function in the proposed tracking mechanism is related to the received signal power, sea surface state (i.e., wave height, magnitude of reflection coefficient, etc.), target's current state, and the observation noises. Particularly, the power of the received signal in the presence of sea surface multipath reflections can be presented by (4.12),

$$P = Af(\theta) + ve^{j\alpha}Af(\theta_s) \tag{4.12}$$

where A represents the amplitude of a target echo, $f(\theta)$ and $f(\theta_s)$ are the antenna voltage gain pattern from a target elevation angle θ and grazing angle θ_s. Given a monpulse radar with two beams, the sum of the signal power and the difference of the signal power can be calculated by $\sum = P_1 + P_2$ and $\Delta = P_1 - P_2$, respectively, where P_1 and P_2 are power of signals received in phase center 1 and 2 of the radar from different reflected paths, as shown in Figure 15.

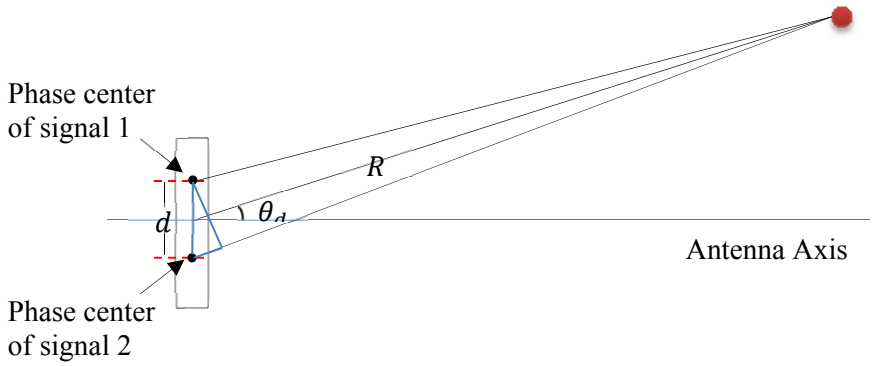


Figure 15: Monopulse radar with two beams

To this end, the measurement function is provided by the monopulse ratio as given in (4.13), where U_k is zero mean white Gaussian noise.

$$y_k = \frac{Af(\theta_k^1)+ve^{j\alpha}Af(\theta_s^1)+Af(\theta_k^2)+ve^{j\alpha}Af(\theta_s^2)}{[Af(\theta_k^1)+ve^{j\alpha}Af(\theta_s^1)]-[Af(\theta_k^2)+ve^{j\alpha}Af(\theta_s^2)]} + U_k \tag{4.13}$$

In order to better illustrate the proposed tracking mechanism, a 3D state-space model is also developed in this study. Here, the states of a target are composed of seven functions representing the range (R_k), elevation angle ($\theta_k^{xy}, \theta_k^{xz}, \theta_k^{yz}$) and velocity ($v_k^{xy}, v_k^{xz}, v_k^{yz}$) in the xy, xz and yz planes, respectively. The state functions are provided as shown in (4.12)-(4.18), where a, b and c are parameters defined for target's movement, and state noise N_k is captured by a 7x7 matrix.

$$R_{k+1} = \sqrt{(R_k sin\theta_k^{yz} + v_k^{yz}\Delta t)^2 + (R_k sin\theta_k^{xy} + v_k^{xy}\Delta t)^2 + (R_k sin\theta_k^{xz} + v_k^{xz}\Delta t)^2} \tag{4.12}$$

$$\theta_{k+1}^{yz} = \theta_0^{yz} + sin\left(\theta_k^{yz} + k/a\right) - k/b + N_k \tag{4.13}$$

$$\theta_{k+1}^{xy} = \theta_0^{xy} + csin\theta_k^{xy} + N_k \tag{4.14}$$

$$\theta_{k+1}^{xz} = arcsin(\sqrt{1 - (sin\theta_{k+1}^{yz})^2 - (sin\theta_{k+1}^{xy})^2}) \tag{4.15}$$

$$v_{k+1}^{yz} = \frac{c(f_0 - f_{k+1})}{f_{k+1}} sin\theta_{k+1}^{yz} \tag{4.16}$$

$$v_{k+1}^{xy} = \frac{c(f_0 - f_{k+1})}{f_{k+1}} sin\theta_{k+1}^{xy} \tag{4.17}$$

$$v_{k+1}^{xz} = \frac{c(f_0 - f_{k+1})}{f_{k+1}} sin\theta_{k+1}^{xz} \tag{4.18}$$

Once the state-space model is established, the particle filtering-based estimation algorithm is then incorporated to track target's posterior states. In the following section, the particle filtering algorithm utilized in this study is discussed in detail.

4.4. Estimation of a Target's State using Improved Particle Filtering Algorithm

The particle filtering algorithm has been proved to perform well in the target tracking problem in several studies as illustrated in Section 2.2., Chapter 2. Acknowledging the fact that selection of different importance densities may present variances in performances of particle filtering algorithms, in this study, two different density selection rules that are structurally dissimilar are considered.

The first one is a density rule that is most commonly used in filtering literature (Doucet and Johansen, 2008), which is based on the idea of drawing samples from the prior instead of the posterior assuming that the impact of the recent evidence y_n is trivial and can be ignored. Embedding this rule, a PF algorithm becomes easier to compute and implement, but may encounter degeneracy problems, in which all the mass may be concentrated on a few random samples with most of particles having negligible weights as n (iteration number) increases. However, as drawing samples only from the prior ignores the measurements, it leads to poor performance on the estimation especially when observations have significant effects on the posterior states.

In order to outshine these barriers, the improved importance density selection rule that is proposed in the first part of this thesis (i.e., Chapter 3) is considered as a second rule. This rule is derived taking both the prior (previous state) and the likelihood (measurement) into consideration simultaneously, thereby it has the potential to reduce

the degeneracy problems and improve the performances of the particle filters. Figure 16 presents the operational details of the proposed target tracking mechanism incorporating particle filtering algorithm.

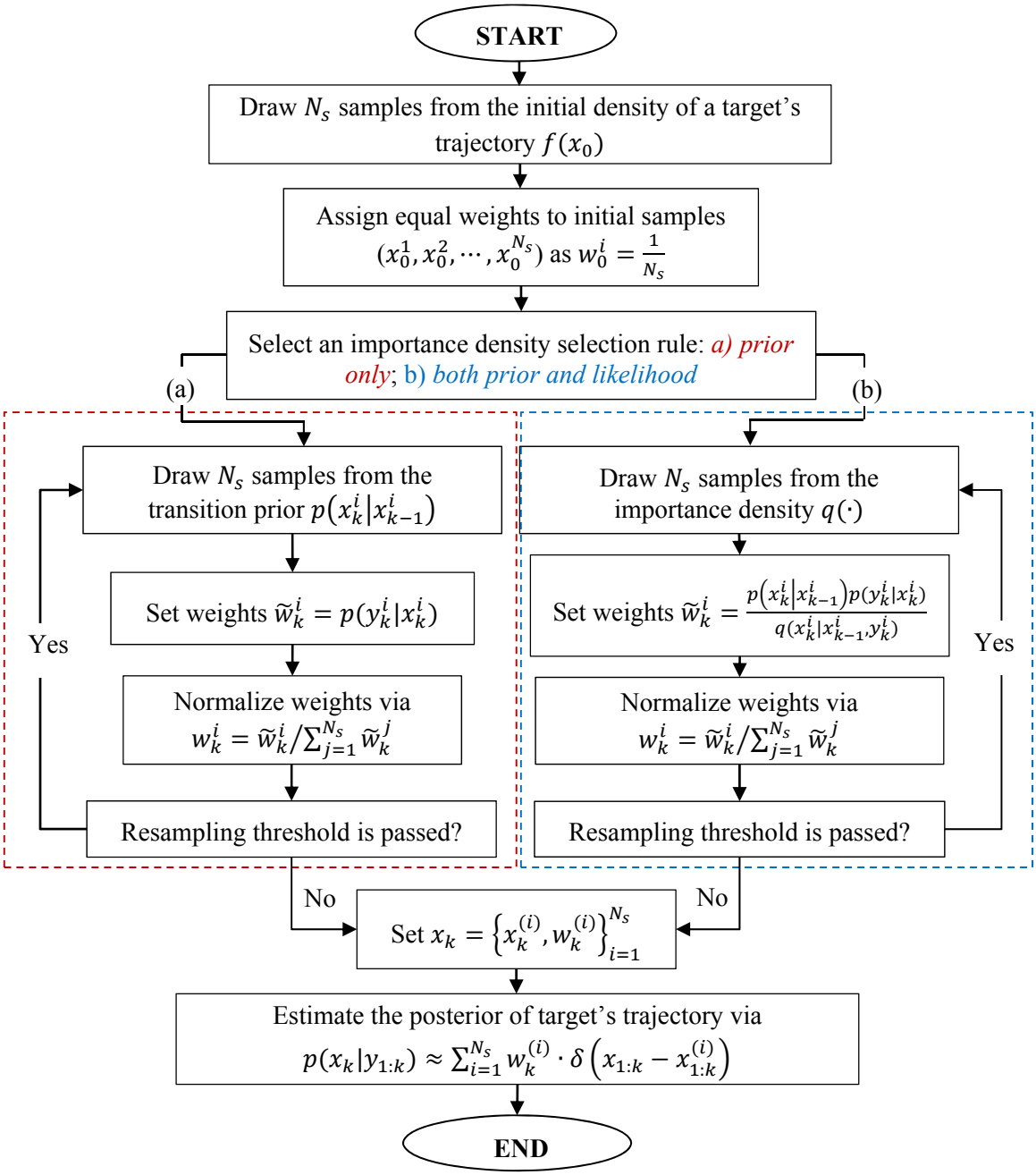


Figure 16: Operations of particle filtering algorithms embedding different density selection rules

4.5. Evaluation of the Proposed Particle Filtering-based Tracking Mechanism

In order to demonstrate the feasibility and validation of the proposed tracking mechanism in the presence of sea surface multipath, a series of real-world related case studies have been designed with different functional settings, where both specular and diffuse reflections are considered as described in Sections 4.2 and 4.3. In these cases, the accuracy of the results (target's posterior states) obtained from the proposed particle filtering algorithm (PF) with two different density selection rules are benchmarked against that of the unscented particle filtering (UPF). Experiments are carried out for 7 different scenarios. In each of these scenarios, 50 replications are conducted with 400 initial particles.

In the cases, a target's actual trajectory is characterized as a set of nonlinear state space equations, in which both process and observation noises are generated from a Gaussian distribution. The estimation accuracies are evaluated in terms of their root mean square error (RMSE) values. Moreover, the resampling threshold value is set to be $0.5N_s$ (N_s represents the number of particles) as a literature standard. Once a serious degeneracy occurs and this threshold is passed, the resampling procedure incorporated in the PF algorithms is triggered. The initial radar and environmental parameters incorporated in the model are provided in Table 7.

Table 7: Initial settings of the radar and its environment

Antenna Center Height	18 m
Polarization	Vertical
Transmitted Frequency	8GHz
Antenna Separation	1.2m
Radius of the Earth	6.371e+6 m

Seawater Temperature	10 ℃
Static Dielectric Parameter of Seawater	72.2
Relaxation Time of Seawater	1.21e-11
Ionic Conductivity of Seawater	3.6e+10

4.5.1. Scenario 1-4: Target is Flying away from the Radar with a Constant Velocity

In the first four scenarios, we aim at investigating the impacts of multipath propagation on the performance of the proposed tracking mechanism. In these scenarios, the target is flying away from the radar with a constant velocity of 68m/s under different conditions in terms of different number of reflection paths. Particularly, in the first scenario, the sea-surface is assumed to be calm and only specular reflection is considered; while in scenarios 2 - 4, both specular and diffuse reflections are included and the number of reflection paths is set as 5, 10, and 20, respectively. The initial range between the radar and target is assumed to be 2000m. Here, the results obtained from the proposed tracking mechanism with embedded particle filtering algorithms (PF_P represents the filter having its transition prior as its importance density and PF_{PAL} represents the filter having a combination of its prior and likelihood taken as its importance density) are evaluated against that of the unscented particle filtering algorithm (UPF).

Table 8 summarizes tracking results obtained from the cases for Scenarios 1 - 4. It is noticed that as the number of reflection paths increase, the accuracies of state estimation obtained from the proposed tracking mechanism are also increased in both PF_{PAL} and PF_P. Here, the multipath propagation over the sea surface have significant impacts on the radar tracking, especially when there is only specular reflection that causes significant signal fading. However, this impact is being captured by the PF_{PAL}

algorithm to a great extent in its tracking qualities compared to the UPF and PF_P algorithms. Moreover, regarding the elapsed time, which is considered as a measurement of computational resources utilization for different algorithms, both PF_P and PF_{PAL} perform much better than that of the UPF, indicating the superiority of the particle filtering algorithm in dealing with the tracking problems of large-scale systems.

Table 8: Results obtained for different numbers of reflection paths

Scenario		Horizontal Position		Vertical Position		Elevation		Elapsed Time
		RMSE-mean	RMSE-var	RMSE-mean	RMSE-var	RMSE-mean	RMSE-var	
S1 ($P = 1$)	UPF	0.8674	0.2935	2.83634	0.1634	0.0439	0.0039	20.1831
	PF_P	0.8450	0.2.557	2.7997	0.1623	0.0435	0.0029	0.1749
	PF_{PAL}	0.2847	0.0279	0.9089	0.1259	0.0141	0.0002	0.1765
S2 ($P = 5$)	UPF	0.4829	0.2817	1.2599	0.0584	0.0195	0.0005	33.2319
	PF_P	0.3443	0.0656	1.2877	0.0305	0.0195	0.0006	0.4795
	PF_{PAL}	0.2624	0.0173	0.9119	0.0106	0.0141	0.0002	0.5319
S3 ($P = 10$)	UPF	0.4122	0.2082	1.1081	0.0189	0.0171	0.0003	49.7161
	PF_P	0.3002	0.0276	1.1171	0.0133	0.0172	0.0003	0.8016
	PF_{PAL}	0.2675	0.0204	0.9428	0.0121	0.0145	0.0002	1.0296
S4 ($P = 20$)	UPF	0.3846	0.1613	1.0309	0.1324	0.0161	0.0003	74.1158
	PF_P	0.2761	0.1938	1.0255	0.1318	0.0159	0.0002	1.5629
	PF_{PAL}	0.0258	0.0253	0.9259	0.1431	0.0143	0.0002	1.5643

Detailing this demonstration of the relationship between the estimation accuracies and number of reflection paths, additional experiments are conducted with gradually increasing number of reflection paths. Figure 17 shows results obtained from UPF, PF_P, versus PF_{PAL} algorithms for estimation qualities in terms of the RMSE-mean values for various scenarios of number of reflection paths. Results presented here reveal that performances of the UPF and PF_P are highly related to the state of the sea surface while only slight fluctuations are shown for the performance of the PF_{PAL} when the number of reflection paths increases. Since the sea surface state is difficult to predict and detect, the

number of reflection paths is rarely known in the real word target tracking applications. From this perspective, the PF_{PAL} estimation algorithm proves to be more reliable and useful in estimation of the posterior distribution of the target's motion when compared to the UPF and PF_P algorithms.

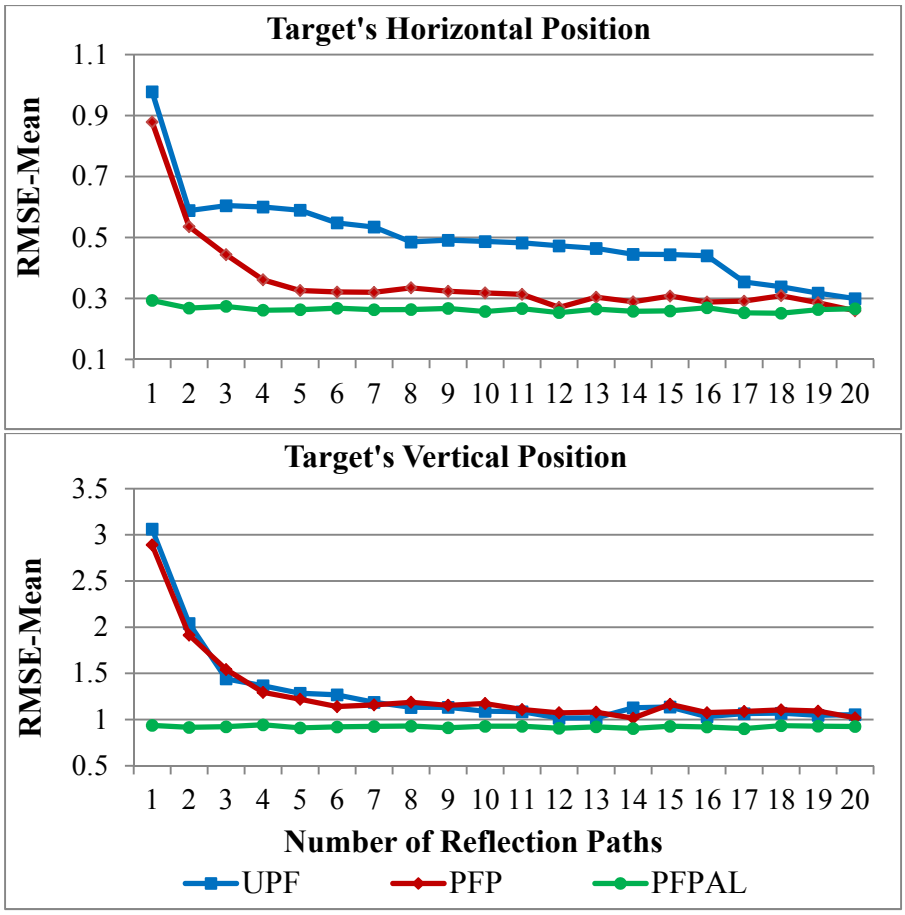


Figure 17: Mean of RMSE values obtained from UPF, PFP and PFPAL as a function of the number of reflection paths

4.5.2. Scenario 5: Target is Flying toward the Radar with a Constant Velocity

In this scenario, the proposed mechanism is tested for its capability to track a target from different directions. Here, a synthetic target is flying towards the radar with a speed of 78.5 m/s where the initial range between the target and radar is 8000m. Similar to

Scenario 2, the number of reflection paths is 5. Screenshots of the actual target trajectory and the estimates obtained via UPF, PF_P and PF_{PAL} algorithms are provided in Figures 18-20, respectively. The red lines represent the target's actual states (i.e., elevations, horizontal and vertical positions) while the blue lines are the estimated results. Results revealed that even though the proposed tracking mechanism is able to track targets from different directions with both density selection rules (PF_P and PF_{PAL}) (compared in Scenarios 2 and 5), the mechanism shows better matching qualities especially when PF_{PAL} algorithm is embedded.

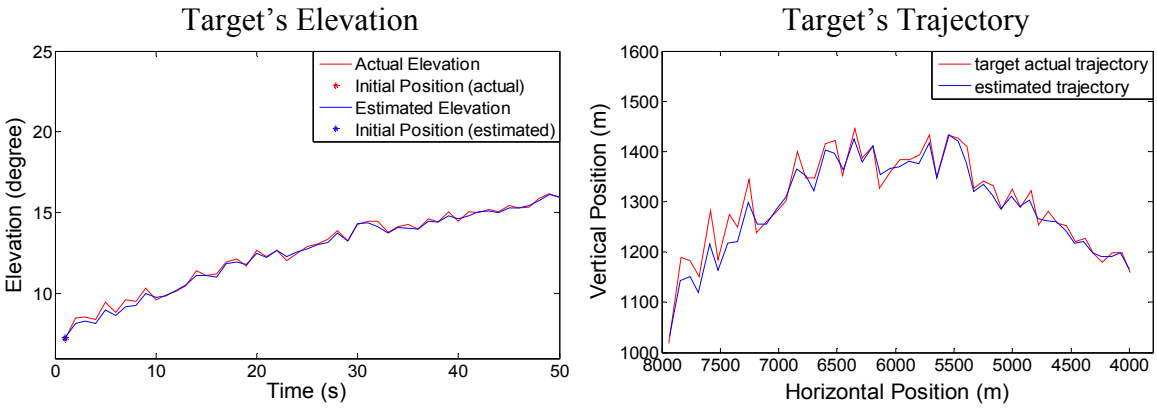


Figure 18: Estimation of a target's state (elevation and trajectory) obtained by UPF

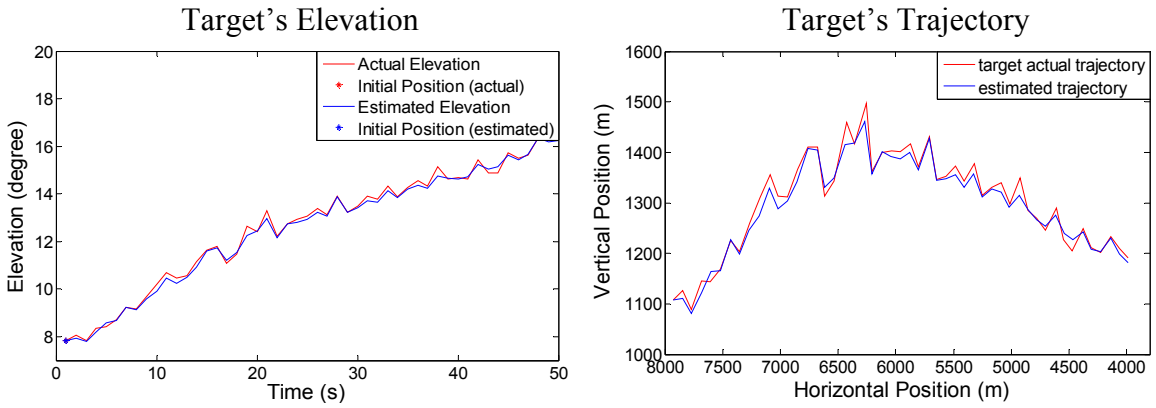


Figure 19: Estimation of a target's state (elevation and trajectory) obtained by PF_P

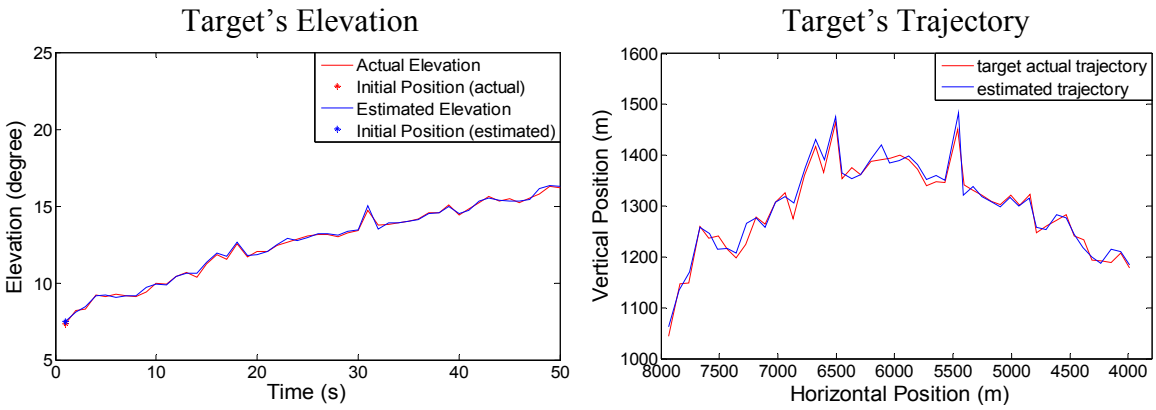


Figure 20: Estimation of a target's state (elevation and trajectory) obtained by PF_{PAL}

4.5.3. Scenario 6: target is Flying with an Accelerating Velocity

In Scenario 6, the proposed tracking mechanism embedding different estimation algorithms is applied to track a target's trajectory when a target is flying with an accelerating velocity. The motion of a target flying away from the radar is simulated and parameters for the processing noises and observation noises are kept as the same while the simulation is running. Meanwhile, different accelerations are used to demonstrate the consistency of the results under different functional settings. The results obtained in this scenario are provided in Table 9. For cases with different accelerations, the tracking mechanism incorporated with PF_{PAL} algorithm provide better estimation accuracies than that of the UPF and the PF_P algorithms in terms of the expected value and variance of the root mean square error in estimating the target's states (i.e., position and velocity). It is noticed that as the target's velocity accelerates, the estimation accuracies presented by the UPF and PF_P algorithms slightly declines. However, an increased level of the acceleration does not generate significant effects on the performance of the PF_{PAL} algorithm. Therefore, the proposed tracking mechanism embedding the PF_{PAL} as its state

estimation algorithm can be applied to the tracking problems without being restricted to the velocities.

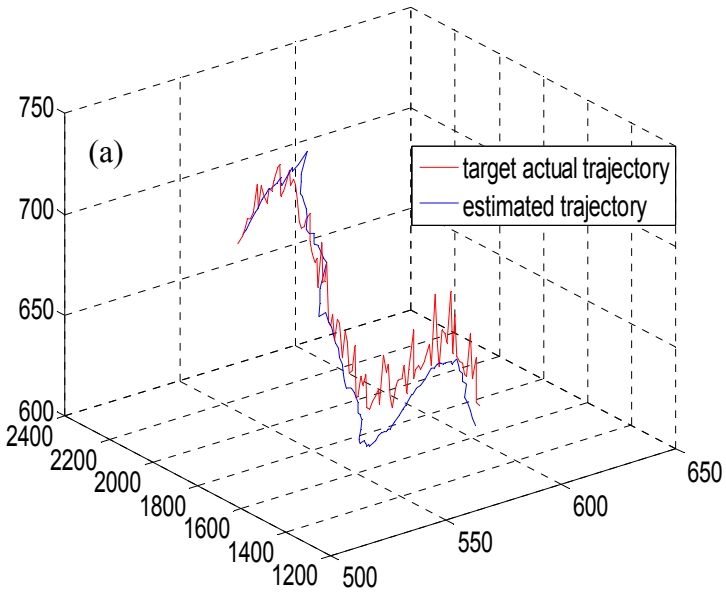
Table 9: Results obtained for different accelerations in target's velocity

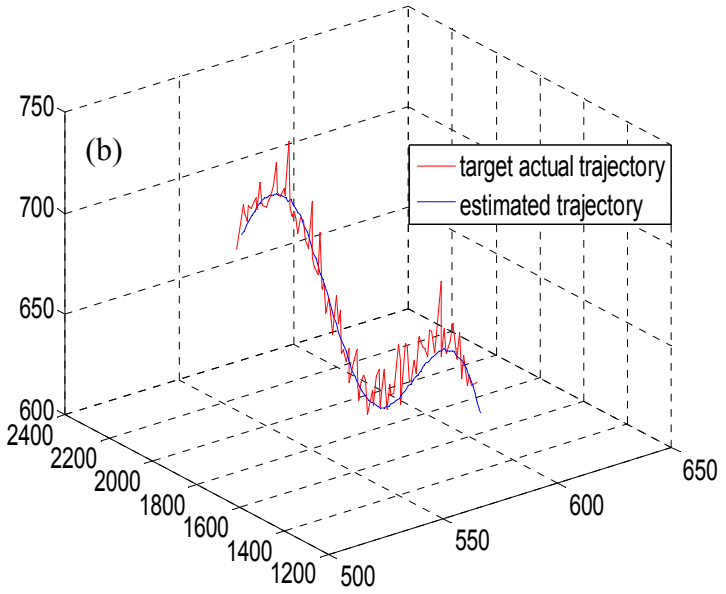
Acceleration		x position		y position		Horizontal Velocity		Vertical Velocity	
		RMSE-mean	RMSE-var	RMSE-mean	RMSE-var	RMSE-mean	RMSE-var	RMSE-mean	RMSE-var
$a = 0.05$	UPF	0.3919	0.2569	1.0810	0.0327	0.1706	0.0002	0.5549	0.0012
	PF_P	0.3166	0.0439	1.1531	0.2238	0.1810	0.0003	0.5919	0.0027
	PF_{PAL}	0.2581	0.0152	0.9238	0.0859	0.0985	0.0001	0.1071	0.0001
$a = 0.1$	UPF	0.4065	0.1379	1.1399	0.2498	0.1799	0.0003	0.5883	0.0016
	PF_P	0.2964	0.0291	1.1267	0.1595	0.1762	0.0003	0.5746	0.0045
	PF_{PAL}	0.2562	0.0151	0.9030	0.0755	0.0987	0.0001	0.1011	0.0001
$a = 0.2$	UPF	0.3774	0.0629	1.1921	0.2643	0.1931	0.0003	0.6407	0.0015
	PF_P	0.2994	0.0378	1.1394	0.2009	0.1799	0.0004	0.6050	0.0038
	PF_{PAL}	0.2521	0.0115	0.9194	0.0907	0.0966	0.0001	0.1050	0.0001
$a = 0.4$	UPF	0.4937	0.1973	1.1235	0.1496	0.1969	0.0003	0.6523	0.0057
	PF_P	0.3069	0.0324	1.1563	0.2177	0.1917	0.0004	0.6513	0.0043
	PF_{PAL}	0.2685	0.0102	0.9663	0.1170	0.1002	0.0001	0.1021	0.0001
$a = 0.8$	UPF	0.4817	0.2403	1.2158	0.2359	0.2090	0.0004	0.7246	0.0023
	PF_P	0.3209	0.0288	1.2360	0.2516	0.2073	0.0006	0.7426	0.0077
	PF_{PAL}	0.2864	0.0329	1.0174	0.1786	0.1008	0.0001	0.1043	0.0002
$a = 1.6$	UPF	0.5772	0.4841	1.2799	0.0822	0.2506	0.0010	0.9162	0.0151
	PF_P	0.3315	0.2698	1.3102	0.2089	0.2441	0.0009	0.9078	0.0139
	PF_{PAL}	0.2897	0.0202	1.0915	0.1601	0.0983	0.0002	0.1001	0.0003

4.5.4. Scenario 7: Target Tracking with a 3D State-Space Model

In the last scenario, we apply the designed 3D tracking mechanism for the simulation of a target's trajectory. Model details have been illustrated earlier in Section 4.3. The simulations are conducted for 20 replications each lasting 100 time steps (0.1s/step) with the sample size of 200, and the number of reflection paths is 10. Figure 21 depicts the screenshots of the actual versus estimated target trajectories obtained via our 3D tracking mechanism incorporating different estimation algorithms, in which

figure (a) shows the results obtained via UPF, (b) shows the results obtained via PF_P and (c) is the results obtained via PF_{PAL}. Results have revealed that the proposed tracking mechanism is capable of tracking the target with multipath interference over the sea surface. Meanwhile, among the three screenshots, figure (c) presents the highest matching qualities between the actual target's trajectory and the estimated one.





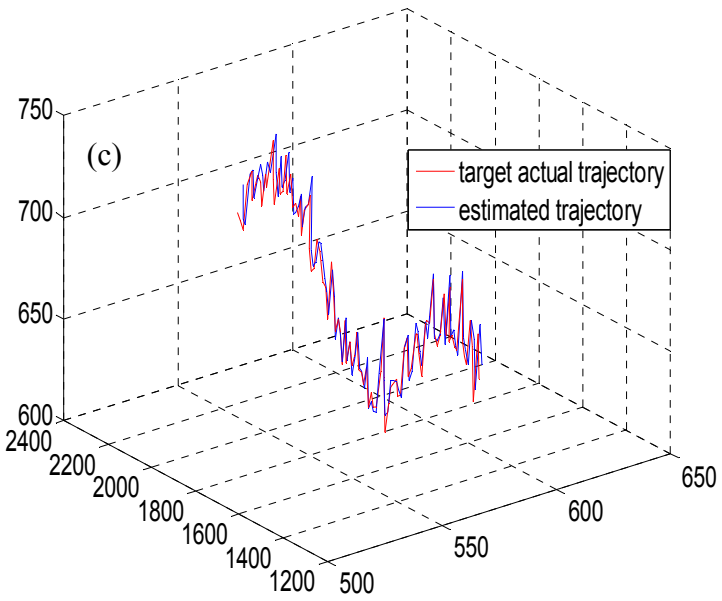


Figure 21: Actual versus estimated target trajectories obtained via the proposed 3D tracking mechanism embedding different estimation algorithms

Confirming the results obtained from the earlier scenarios, the tracking mechanism embedding PF_{PAL} as its estimation algorithm outperforms the others in terms of their estimation accuracies.

Chapter 5: A Novel Particle Filtering Algorithm with Efficient Sampling for Multi-objective Optimization

Particle filtering algorithms for state estimation of large-scale dynamic system have been illustrated in the previous two chapters. In order to extend the employment of particle filtering to the area of optimization, in this chapter, a novel method based on the particle filtering algorithm with an efficient sampling procedure for multi-objective optimization problems is presented.

5.1. Formulation of a Multi-objective Optimization Problem

Multi-objective Optimization, also known as vector optimization, is concerned with searching for optimal solutions for two or more objective simultaneously. Generally, in this kind of optimization problem, the optimal solution is not unique, which means it doesn't exist one solution that can satisfy all constrains and optimize all objectives at the same time. Thus, the non-dominated set or Pareto optimal is used to represent the optimal solutions obtained for the multi-objective optimization problems. Mathematically, a general multi-objective optimization problem can be defined as shown in (5.1) - (5.3),

$$\text{Minimize} \qquad \vec{Y} = \vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x})) \tag{5.1}$$

$$\text{s.t.} \qquad g_i(\vec{x}) = 0, \quad i = 1,2, \dots, m \tag{5.2}$$

$$h_j(\vec{x}) \leq 0, \quad for\ minimization, j = 1,2, \dots, p \tag{5.3}$$

where $\vec{x} = [x_1, x_2, \dots, x_q]^T \in R$ and $\vec{Y} = [y_1, y_2, \dots, y_n]^T$ are decision vector and objective vector, q and n represent the number of decision variables and the number of objectives, $g_i(\vec{x})$ and $h_j(\vec{x})$ are sets of constraint functions, m and p denote the number

of equality constraints and the number of inequality constraints included in the optimization problem, respectively.

The set of solutions consisting of all decision vectors which cannot be improved in any objective without degradation in other objectives are called Pareto optimal (Zitzler and Thiele, 1999). And mathematically, for a minimization problem, a decision vector $\vec{u} = [u_1, u_2, \dots, u_q]^T \in \vec{x}$ is said to dominate another decision vector $\vec{v} = [v_1, v_2, \dots, v_q]^T \in \vec{x}$ if and only if:

$$f_i(\vec{u}) \leq f_i(\vec{v}),\ for\ \ \forall\, i = 1,2, \dots n \tag{5.4}$$

$$f_j(\vec{u}) < f_j(\vec{v}),\ for\ \ \exists\, j = 1,2, \dots n \tag{5.5}$$

A solution is said to be Pareto optimal when there is no other solutions dominate it. That is, the obtained solution cannot be improved in one of the objectives without having any adverse influence on the other objectives. Then, the set of all Pareto optimal solutions is defined as the Pareto optimal set and the corresponding objective vectors are said to be on the Pareto front (Ngatchou et al., 2005).

5.2. Proposed Particle Filtering-based Optimization Algorithm

As aforementioned in Section 1.3, Chapter 1, the proposed particle filtering-based optimization algorithm is structurally formed with two procedures, namely, the dynamic weighted allocation procedure (DWA) and the performance-based sampling and resampling procedure (PSR). In this section, the details of these two procedures are presented.

5.2.1. Dynamic Weighted Allocation Procedure

The dynamic weighted allocation (DWA) procedure, which is first proposed by Jin et al. (2004), is an improvement based on the weight aggregation algorithm for multi-objective optimization problems, developing for the distribution of the weights of multiple objectives. The conventional purpose of this procedure is to transform the multi-objective optimization problems into single objective optimization problem. However, in this study, this procedure is modified and incorporated into the algorithm for quantifying the importance of each objective (determining the influence factors of different objectives) and evaluating the performance of the obtained particles. To this end, in this study, the dynamic fitness assigned to different objectives is bundled with the particles' importance weights, and is changed gradually and periodically. In this way, given a large number of iterations, the comprehensiveness and diversity of these influence factors is guaranteed. In Jin's work, the procedure is developed for two objectives. Based on his study, we modify the procedure and extend it to be feasible for multiple objectives, which is realized as shown in (5.6) - (5.9),

$$O_1(k) = A\left|sin\left(\frac{2\pi k}{Q_1}\right)\right| + \varepsilon \tag{5.6}$$

$$O_2(k) = \left(1 - O_1(k)\right)\left|sin\left(\frac{2\pi k}{Q_2}\right)\right| + \varepsilon \tag{5.7}$$

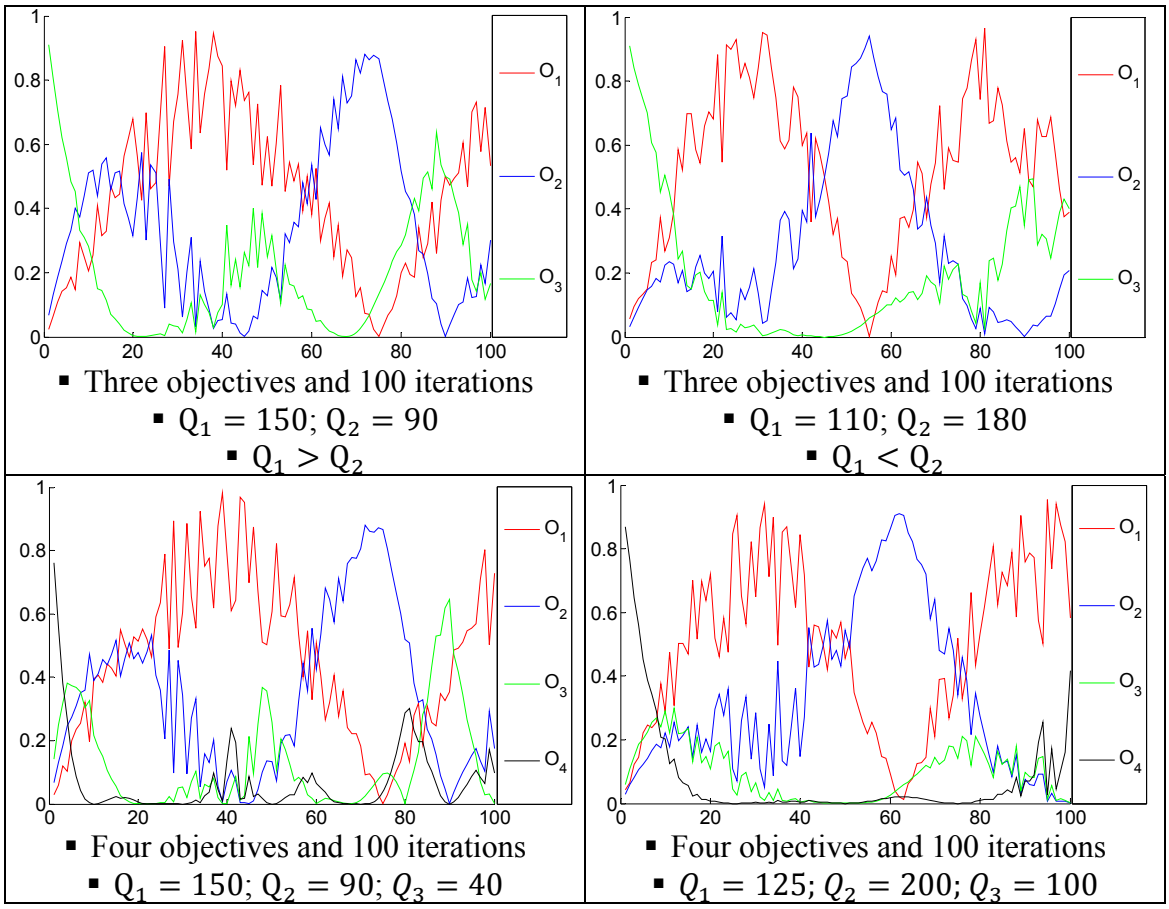
$$\vdots$$

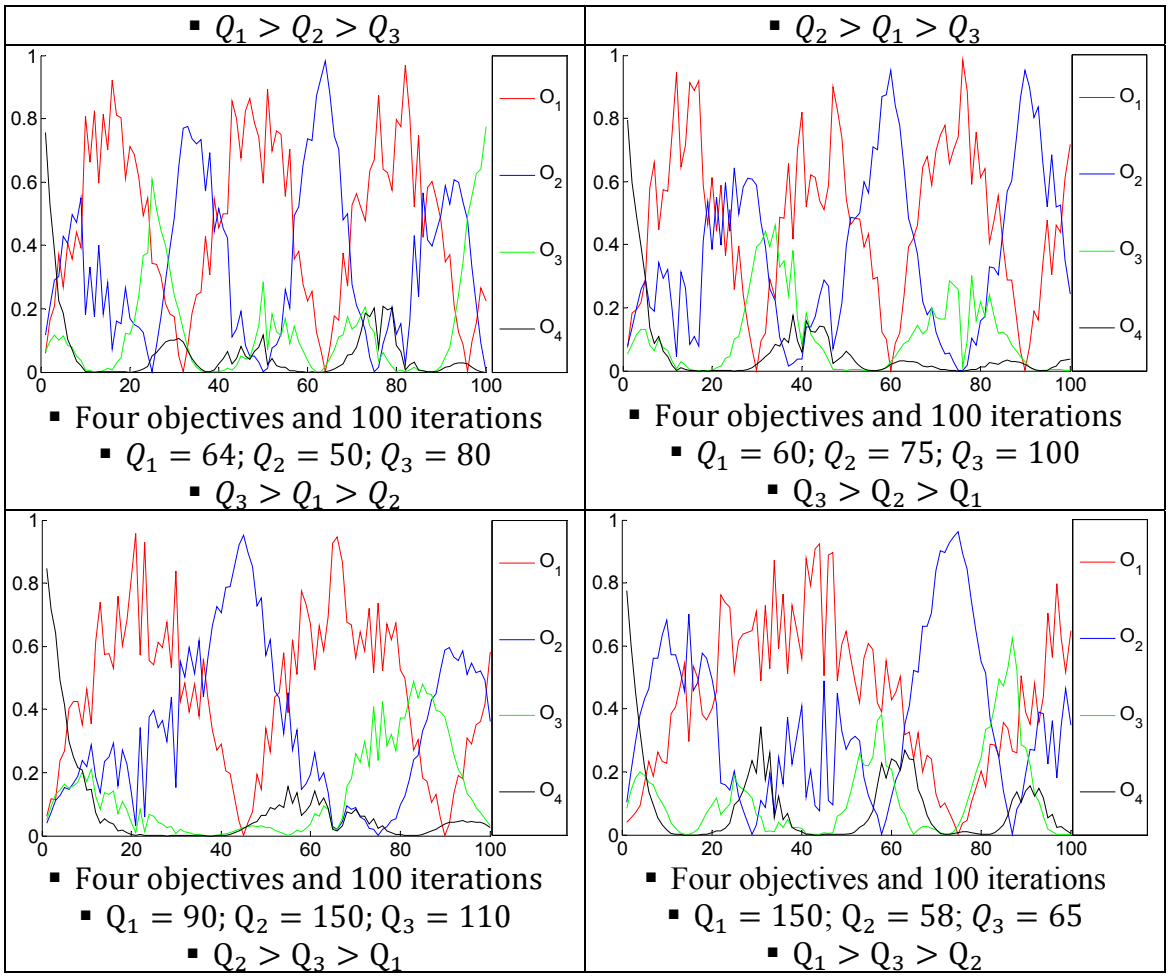
$$O_{n-1}(k) = \left(1 - \sum_{i=1}^{n-2} O_i(k)\right)\left|sin\left(\frac{2\pi k}{Q_{n-1}}\right)\right| + \varepsilon \tag{5.8}$$

$$O_n(k) = 1 - \sum_{i=1}^{n-1} O_i(k) \tag{5.9}$$

where O_n represents the influence factor of the nth objective, $k = 1,2, \dots, T$ is the number of replication, $|\cdot|$ provides the absolute value. Parameters ε and A are set as 10^{-5} and 0.99, respectively, which are enforced to avoid extreme situations in which the influence factors of the objectives are equal to 1 or 0. Here, the frequency of variation is controlled by the user-defined parameter Q_n (n represents the number of objectives). Figures shown in Table 10 provide several different combinations of influential factors with exemplary selections of Q_n for three objectives and four objectives optimization problems. From these figures, it is noticed that all possibilities in determining the influence factors of different objectives can be captured with an arbitrary selection of Q_n.

Table 10: Different combinations of influence factors of multiple objectives





Specifically, for bi-objective optimization problems, we have provided a guidance on the selection of Q_1, which is a piecewise function of the number of iterations. Correspondingly, once Q_1 is determined, the influence factors of the two objectives are calculated as shown in Figure 22.

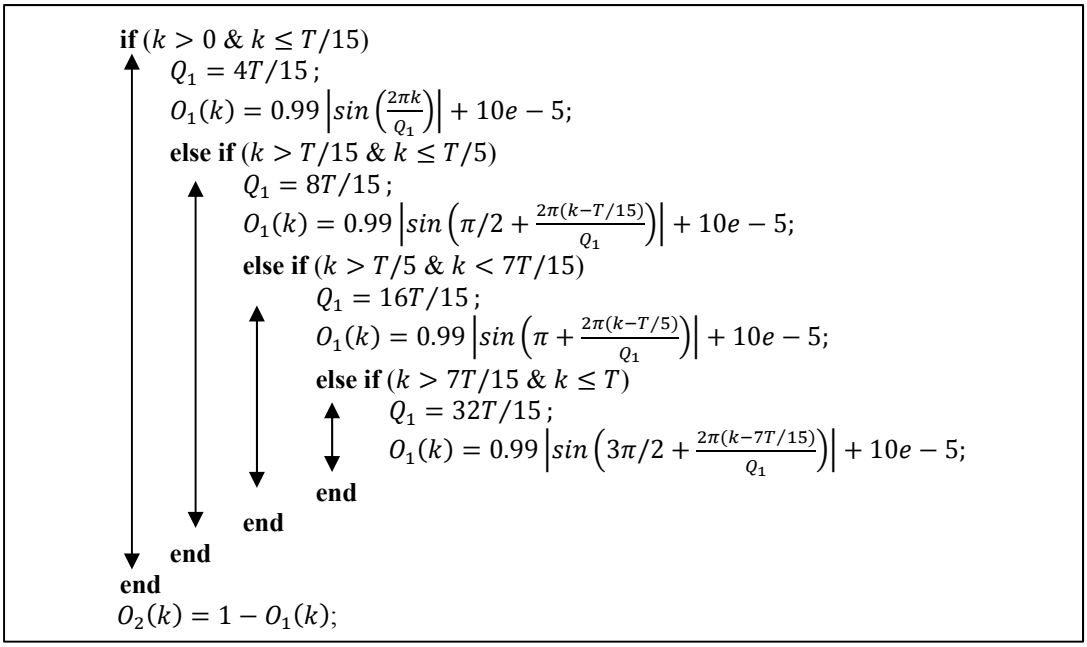


Figure 22: DWA procedure for bi-objective optimization problems

Details for the implementation of DWA are presented together with the explanation of PSR in the following subsections.

5.2.2. Performance-based Sampling and Resampling Procedure

The proposed performance-based sampling and resampling (PSR) procedure originates from the ranking and selection procedure. The idea is that the optima will be obtained if one of the alternatives is selected amongst all possible solutions and the probability that the selected alternative is truly the "best" is under control. From this perspective, the PSR procedure is equipped with two iteratively steps: 1) generate candidate solutions from a sampling distribution; 2) select the "good" alternatives and update the sampling distribution using the selected candidates for the next iteration.

First, we represent the multi-objective optimization problem using a state-space model, in which the optimal solution is treated as the posterior state to be "estimated",

and the optimal objective values observed are specified as an n-dimension measurements. The problem can be formulated mathematically as shown in (5.10) - (5.12).

$$\vec{x}_{k+1} = \vec{x}_k + \vec{v}_k, \quad k = 0,1,\dots \tag{5.10}$$

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{kn} \end{bmatrix} = \begin{bmatrix} min\, f_1(x_k^i) \\ min\, f_2(x_k^i) \\ \vdots \\ min\, f_n(x_k^i) \end{bmatrix}, \quad k = 0,1,\dots \tag{5.11}$$

$$\vec{x}^* = arg\, min_{\vec{x} \in X}\, Y \quad or \quad \vec{x}^* = arg\, max_{\vec{x} \in X}\, Y \tag{5.12}$$

where $\vec{x}_k$ represents the decision variables and state vector, $x_k^i (i = 1,2,\dots,N)$ represents the ith particle (i.e., candidate solution), $\vec{y}_k$ is the minimum values of each objective for a minimization problem, $\vec{v}_k$ is the vector of process noise according to the Gaussian distribution, X denotes the solution space, $Y = [y_{k1}, y_{k2}, \dots, y_{kn}]$ is objective vector, $\vec{x}^*$ is our target Pareto optimal solution, k represents the iteration, and n denotes the number of objectives. Once the particles are obtained, their performances are evaluated in terms of the importance weights, which are calculated via (5.13),

$$w_k^i = O_1(k) * g\left(y_{k1} \middle| x_k^i\right) * + O_2(k) * g\left(y_{k2} \middle| x_k^i\right) * + \cdots + O_n(k) * g\left(y_{kn} \middle| x_k^i\right) \tag{5.13}$$

where $g\left(y_{kn} \middle| x_k^i\right) = \frac{\sum_{i=1}^{N} (f_n(x_k^i) - y_{kn}) - (f_n(x_k^i) - y_{kn})}{\sum_{i=1}^{N} (f_n(x_k^i) - y_{kn})}$.

Here, a strategy that effectively governs the evolution of the sampling distribution to realize fast convergence is critical. In this study, the evolution is controlled by the process noise added in the state evolution function in each iteration. In particular, process

noise v_k^i is draw from a normal distribution function as $v_k^i \sim N(\mu_k^i, \sigma_k^i)$. The value of the mean μ_k^i is determined by the importance weight of particle i (i.e., w_k^i), the particle's value (i.e., x_k^i) and the "best" particle's value (i.e., x_k^{best}) as shown in (5.14). The value of the variance σ_k^i is calculated based on the solution domain as given in (5.15)

$$\mu_k^i = \left(1 - w_k^i\right) * \left(x_k^{best} - x_k^i\right) \tag{5.14}$$

$$\sigma_k^i = 1\% * \left(X_{max} - X_{min}\right) \tag{5.15}$$

With this strategy, 68% of the particles generated in the next iteration are within one standard deviation away from the particle that performs the best in the previous iteration, 95% of the particles lie within two standard deviations and about 99.7% are within three standard deviations. Therefore, as iteration increases, the distribution function becomes more and more concentrated on the promising region of the solution space, indicating a higher probability to obtain solutions with good performances. Figure 23 provides a visualized explanation of the proposed sampling procedure. The areas highlighted with green colors represent the probable regions that the particles generated in the next iteration is likely located in, and here a darker color shows a higher probability.

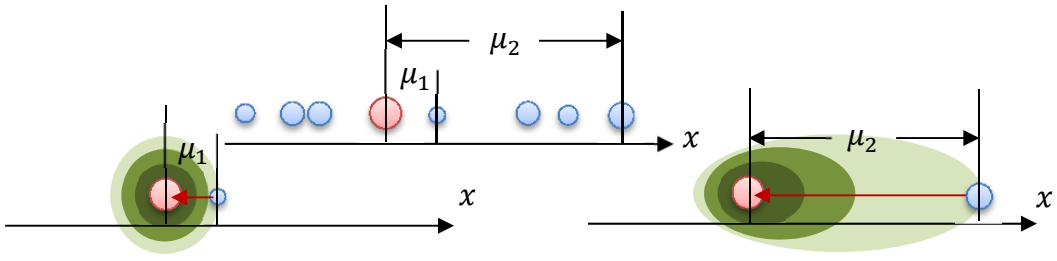


Figure 23: Evolution of the sampling distribution

Moreover, a mutation procedure is also incorporated in the resampling step to keep the algorithm from the local optimum trap. In this study, we have designed two options for resampling of the particles. In the first option, resampling is triggered in each iteration. However, only a small portion of particles (e.g., 10%) are resampled in each iteration. In the second option, resampling is triggered when the ratio of the effective sample size to the sample size exceed a threshold (i.e., λ) that is predefined by the decision makers, as shown in (5.16). The effectiveness of the two resampling procedures is discussed later.

$$r = \frac{N_e}{N} = \frac{1}{1+var(w_k^i)/E^2(w_k^i)} > \lambda \tag{5.16}$$

In order to draw samples efficiently and achieve the optima without taking considerable computational efforts, an optimal computing budget allocation (OCBA) algorithm (Chen and Lee, 2011) is also incorporated in the proposed PSR procedure. Firstly, the particles are divided into different numbers of groups. Secondly, for each group, the particles are ranked in an ascending order in terms of their importance weights. Moreover, the means and standard deviations of particles' importance weights in each group are calculated as shown in Table 11. Finally, the OCBA algorithm for determining the number of particles to be sampled for each group in the next iteration is carried out. The implementation of the OCBA algorithm is presented as shown in Figure 24, and the Matlab code for OCBA is attached in Appendix II.

Table 11: Means and Standard Deviations of Particles' Importance Weights

Iteration		Different Groups of Particles

		1	2	⋯	m
1	Mean	μ_{11}	μ_{12}	⋯	μ_{1m}
	Std.	σ_{11}	σ_{12}	⋯	σ_{1m}
2	Mean	μ_{21}	μ_{22}	⋯	μ_{2m}
	Std.	σ_{21}	σ_{22}	⋯	σ_{2m}
⋮	Mean	⋮	⋮	⋯	⋮
	Std.	⋮	⋮	⋯	⋮
k	Mean	μ_{k1}	μ_{k2}	⋯	μ_{km}
	Std.	σ_{k1}	σ_{k2}	⋯	σ_{km}

Step 1: Initialization ($k = 1$)

- Suppose that a total number of N particles (i.e., $[x^1, x^2, \dots, x^N]$)have been drawn, and their importance weights (i.e., $[w^1, w^2, \dots, w^N]$) have been calculated. Divide the particles into M groups,

$$G_m = \begin{bmatrix} x_m^{(N/M)(m-1)+1}, x_m^{(N/M)(m-1)+2}, \dots, x_m^{(N/M)m} \\ w_m^{(N/M)(m-1)+1}, w_m^{(N/M)(m-1)+2}, \dots, w_m^{(N/M)m} \end{bmatrix}$$

- Depending on the input parameters, N/M particles are generated for each group during initialization.

$$k \leftarrow 1;$$
$$n_k^1 = n_k^2 = \cdots = n_k^m = N/M$$

- Set the additional total budget (number of particles to be resampled in iteration $k+1$) N_R

While $\left(\sum_{m=1}^{M} n_m^k < N_R\right)$ repeat steps 2-4:

Step 2:

- Calculate the means of particles' importance weights for each group (i.e., $\mu_{km} = \frac{M}{N} \sum_{j=(N/M)(m-1)+1}^{(N/M)m} w_m^j$) and the corresponding standard deviations (i.e., $\sigma_{km} = \sqrt{\frac{\sum_{j=(N/M)(m-1)+1}^{(N/M)m} (w_m^j - \bar{w}_k(m))^2}{N/M-1}}$, $m = 1,2, \dots, M$)

Step 3:

- Calculate the number of particles that will be sampled in the next iteration for each particle groups ($n_{k+1}^1, n_{k+1}^2, \dots, n_{k+1}^m$) using the following equations where b is the group with the best performance:

$$\frac{n_{k+1}^i}{n_{k+1}^j} = \left(\frac{\sigma_{ki}(\mu_{kb} - \mu_{kj})}{\sigma_{kj}(\mu_{kb} - \mu_{ki})}\right)^2, \; for\; i > j, \; i, j \neq b$$

$$n_{k+1}^b = \sigma_{kb} \sqrt{\sum_{i=1, i \neq b}^{m} (n_i^{k+1}/\sigma_{ki})^2}$$

Step 4: Draw $[\max(n_{k+1}^m - n_k^m, 0)]$ (for group m) for iteration k+1; $k \leftarrow k+1$.

End

Figure 24: Implementation of the OCBA algorithm for efficient sampling

Finally, Figure 25 provides a visualized diagram of the performance-based sampling and resamping procedure (PSR) in the proposed particle filtering based multi-objective optimization algorithm.

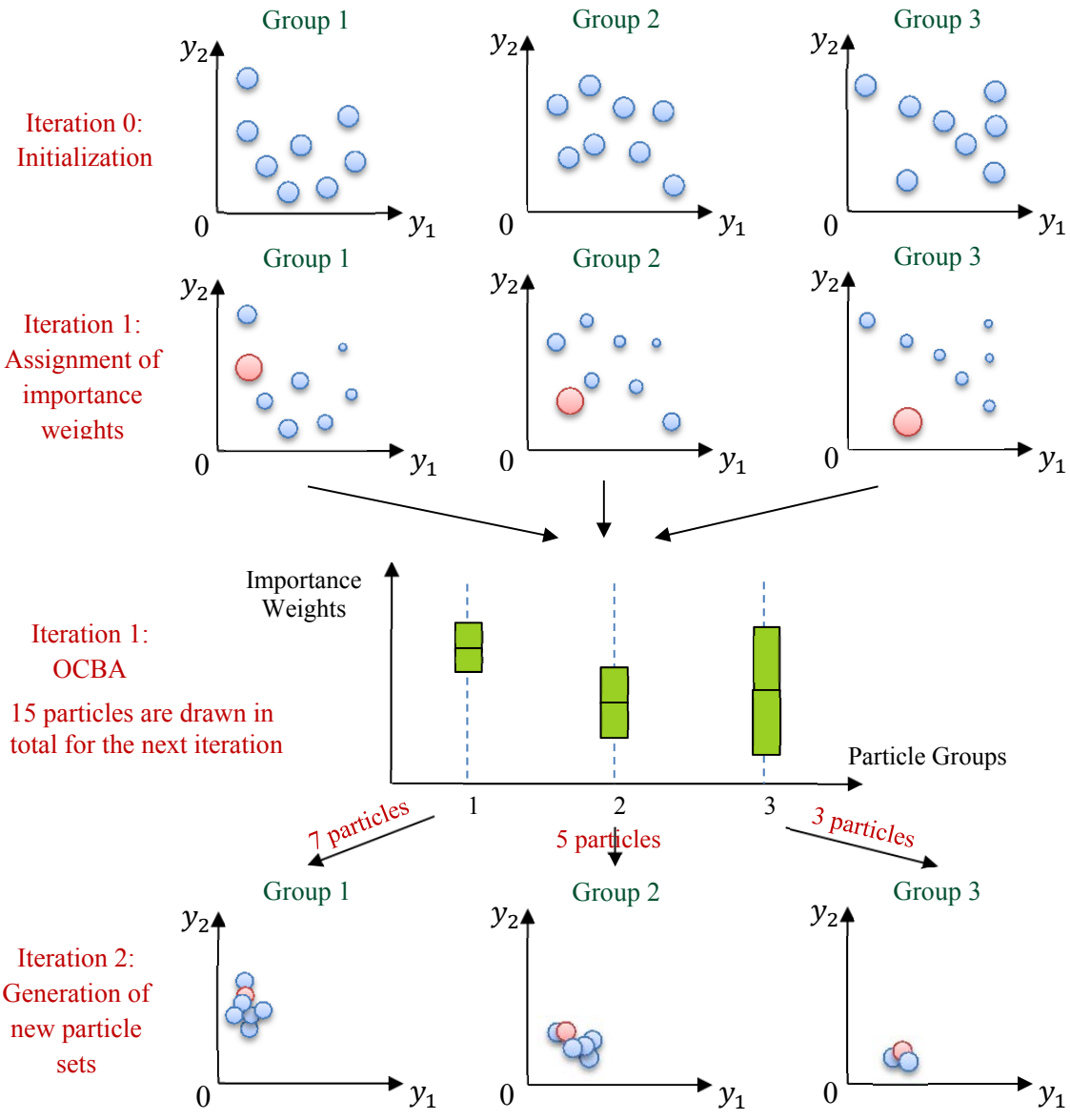


Figure 25: The performance-based sampling and resampling procedure (PSR) in the proposed optimization algorithm

In the next subsection, the performance of the proposed algorithm is evaluated through a real-world case study on an environmental and economic load dispatch problem.

5.3. Case Study II: an Environmental and Economic Load Dispatch Problem

In this subsection, the general formulation of the EELD problem, the implementation of the proposed particle filtering based optimization algorithm on the EELD problem, the designed experiments and obtained results are discussed.

5.3.1. Problem Definition and Formulation

The environmental economic load dispatch (EELD) problem is to minimize two competing objective functions, fuel cost and emission, while satisfying several equality and inequality constraints. Particularly, assume N_G generators are operated in the system, the generators' cost curves can be represented by a function related to the cost coefficients of each generator and the real power outputs of generators that are defined as decision variables. Regarding to the environmental objective, the most important emissions considered are the sulfur dioxide (SO_2) and nitrogen oxides (NO_x). Here, SO_2 are dependent on fuel consumption thus the formulation for the emission of the SO_2 is the same as the cost function used in the economic objective. However, since the nitrogen oxides emissions come from various sources and their production is related to multiple factors (i.e., boiler temperature, air content, etc.), it is much more difficult to formulate the emission of the NO_x. In this study, an approach proposed by Guerrero (2004) that

uses a combination of polynomial and exponential terms is applied to present the SO_2 and NO_x emissions. The objective functions are shown in (5.17) and (5.18), respectively,

$$F_1(P_G) = \sum_{i=1}^{N_G} a_i + b_i P_{Gi} + c_i P_{Gi}^2 \tag{5.17}$$

$$F_2(P_G) = \sum_{i=1}^{N_G} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 + \varepsilon_i e^{\rho_i P_{Gi}} \tag{5.18}$$

where a_i, b_i, and c_i represent the cost coefficients of the ith generator, α_i, β_i, γ_i, ε_i and ρ_i are emission coefficients of the ith generator, $P_G = [P_{Gi}](i = 1,2, \dots N_G)$ is a vector defined as the real power outputs of generators.

For the EELD problem, the major constraints that need to be satisfied are the generator capacity limitations and power balance constraints. More specifically, the constraint shown in (5.19) is to ensuCost re that the real power output of each generator is limited by its capacity. Constraints shown in (5.20)-(5.23) are to satisfy the power balancing requirements such that the total load provided to the system at any time is greater than the summation of what is demanded and what is lost during its transmission.

$$P_{Gi}^{min} \le P_{Gi} \le P_{Gi}^{max}, \quad \forall i \tag{5.19}$$

$$\sum_{i=1}^{N_G} P_{G_i} - P_D - P_{loss} = 0 \tag{5.20}$$

$$P_{G_i} - P_{D_i} - V_i \sum_{j=1}^{N_B} V_j [G_{ij} cos(\delta_i - \delta_j) + B_{ij} sin(\delta_i - \delta_j)] = 0, \ \forall i \tag{5.21}$$

$$Q_{G_i} - Q_{D_i} - V_i \sum_{j=1}^{N_B} V_j [G_{ij} sin(\delta_i - \delta_j) + B_{ij} cos(\delta_i - \delta_j)] = 0, \ \forall i \tag{5.22}$$

$$P_{loss} = \sum_{k=1}^{N_L} g_k [V_i^2 + V_j^2 - 2V_i V_j cos(\delta_i - \delta_j)] \tag{5.23}$$

Here, P_{Gi}^{min} and P_{Gi}^{max} are the minimum and maximum power that can be generated by the ith bus, P_D and P_{loss} represent the power demand and power loss, $V_i$and V_j are the voltage magnitudes at bus i and j, δ_i and δ_j are the voltage angles at bus i and j, G_{ij} and B_{ij} denote the transfer conductance and susceptance between bus i and bus j, respectively, $N_B$is the number of buses, N_L represents the number of transmission lines and g_k is the conductance of the kth transmission line, which connects the bus i and j. The nonlinear equality constrains of (5.21) and (5.22) are solved via the Newton-Raphson method (Grainger and Stevenson, 1994; Weber, 1997) in this study. The use of this method ensures that the output generation values obtained from the particle filtering algorithm satisfy the power balance constraints in terms of real and reactive power, respectively. Figure 26 illustrates the relationship of the candidate solutions generated from the particle filtering algorithm and the real output generation values after power balancing using the Newton-Raphson method.

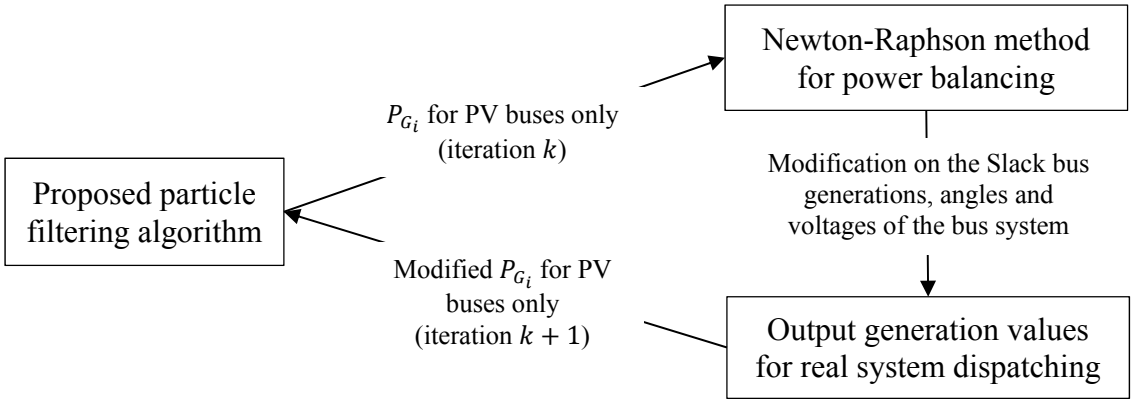


Figure 26: Relationship of the candidate solutions generated from the particle filtering algorithm and the real output generation values

Once the Pareto-optimal solution set is obtained, a fuzzy set theory (Sakawa et al., 1987) is implemented to extract one of the non-dominated solutions as the best compromise solution. Particularly, the ith solution for objective j is represented by a membership function μ_{ij}, defined as $\mu_{ij} = \frac{y_j^{max} - y_{ij}}{y_j^{max} - y_j^{min}}$ (for $\forall j$), where y_{ij} represents the jth objective value in solution i, y_j^{max} and y_j^{min} are the maximum and minimum values of the jth objective function, respectively. Then for each non-dominated solution, the normalized membership function μ_i can be calculated via (5.24), where S is the number of non-dominated solutions. Finally, the best compromise solution is defined as the one having the maximum value of μ_i.

$$\mu_i = \frac{\sum_{j=1}^{2} \mu_{ij}}{\sum_{i=1}^{S} \sum_{j=1}^{2} \mu_{ij}} \quad , \quad \forall i \tag{5.24}$$

Given the formulation of the EELD problem, the implementation of the proposed particle filtering-based optimization algorithm to obtain the Pareto-optimal solutions are provided in the next subsection.

5.3.2. Implementation of the Proposed Algorithm

The procedures to implement the algorithm are given as following:

Step 1: Initialization. To initialize the algorithm, the number of iterations K, the number of particles N, the number of groups M are set firstly. In addition, the first set of particles for each group, which is contained in an $N_G \times \frac{N}{M}$ matrix, is randomly drawn within the interval $[P_{Gi}^{min}, P_{Gi}^{max}]$. Given this, the constraints in terms of the generator

capacity limitations are satisfied. That is, the values of all particles, which are considered as the generator's real output in the optimization problem, are subjected to the generator's capacity. Once the particles are obtained, the values of the corresponding objectives are calculated for all candidate solutions (i.e., particles). Then, for each group, two matrixes are obtained as shown in Figure 27.

$$Y_{1\cdot 0} = F_{1\cdot 0}(G_i) = \begin{bmatrix} y_{G1}^1 & y_{G1}^2 & y_{G1}^3 & \cdots & y_{G1}^N \\ y_{G2}^1 & y_{G2}^2 & y_{G2}^3 & \cdots & y_{G2}^N \\ & & \vdots & & \\ y_{GN_G}^1 & y_{GN_G}^2 & y_{GN_G}^3 & \cdots & y_{GN_G}^N \end{bmatrix} \quad Y_{2\cdot 0} = F_{2\cdot 0}(G_i) = \begin{bmatrix} g_{G1}^1 & g_{G1}^2 & g_{G1}^3 & \cdots & g_{G1}^N \\ g_{G2}^1 & g_{G2}^2 & g_{G2}^3 & \cdots & g_{G2}^N \\ & & \vdots & & \\ g_{GN_G}^1 & g_{GN_G}^2 & g_{GN_G}^3 & \cdots & g_{GN_G}^N \end{bmatrix}$$

Figure 27: Objective values obtained via the initialized particles

Step 2: Feasibility Check, Calculating the Importance Weights. Since the generator capacity constraints are already satisfied in the first step, in this step, all the obtained possible solutions are checked with the power balance constraints. For particle i, if it is feasible, a parameter fea_k^i is set to 1; otherwise, fea_k^i is equal to 0. After that, the importance weights of particles $\widetilde{w}_k^i$ are calculated. Finally, these weights are normalized taking the feasibility into consideration as given in (5.25).

$$w_k^i = \frac{\widetilde{w}_k^i * fea_k^i}{\sum_{i=1}^{N/M} \widetilde{w}_k^i * fea_k^i} \tag{5.25}$$

Step 3: Generating New Candidate Solution (Importance Sampling and Resampling). Solutions obtained from the previous step will be evaluated, and the non-dominated solutions are archived. For each group, particles are ranked in an ascending

order in terms of their importance weights, and the one with the highest weight is selected. Then, the parameters of the normal distribution functions, from which the process noises are generated, are calculated. Given the updated state evolution function, a new set of particles are draw. Here, the number of particles that are obtained for each group is determined by the OCBA procedure as illustrated in Figure 24. If the resampling threshold is reached, new particles are generated according to the embedded resampling rules.

Step 4: Termination and Return the Optimal Solution. Steps 2 and 3 are repeated, and once the predefined number of iterations is reached, the algorithm stops searching for new solutions. Finally, the non-dominated solutions recorded and the corresponding objective values are exported.

5.3.3. Experiments and Results

Given the formulation of the EELD problem, the procedures to obtain the Pareto-optimal solution using the proposed particle filtering-based optimization algorithm, in this subsection, a set of experiments are designed and carried out over an IEEE-30 bus test system. As shown in Figure 28, in this system, bus 1, bus 2, bus 5, bus 8, bus 11, bus 13 are the generator buses (bus 1 is the slack bus, and the other five buses are PV buses), and the other buses are defined as the load buses (PQ buses). The IEEE-30 bus system, which behaves as an epitome of the American Electric Power System, has been heavily studied in the literature as a standard testing case for power systems (Bhagwan and Patvardhan, 1999; Huang et al., 2003; Donde et al., 2008; Panigrahi et al., 2011).

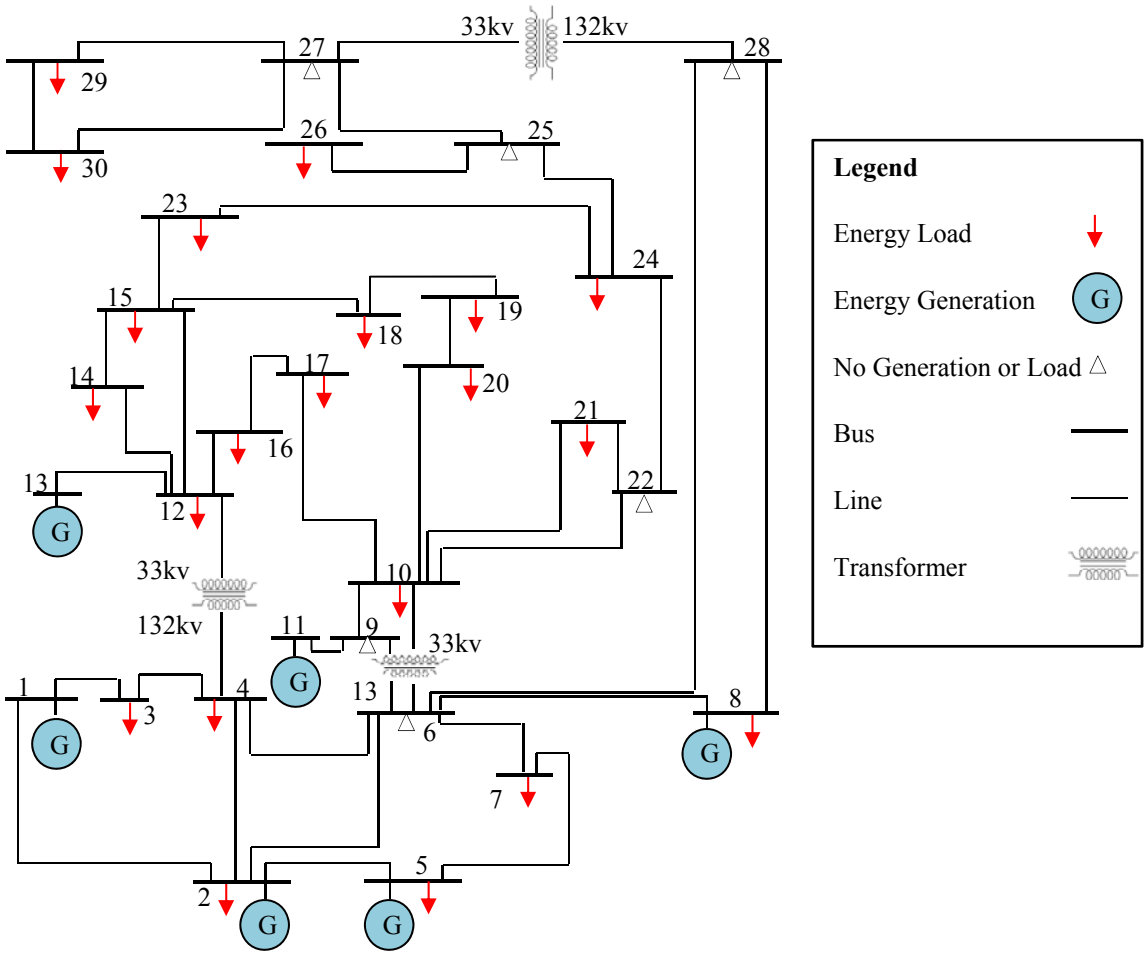


Figure 28: IEEE-30 Bus Test System

The load and line data of the IEEE-30 bus system are obtained from the Power Systems Test Case Archive (University of Washington, 2012) and are attached in Appendix III and IV. The cost and emission parameters required in (5.17) and (5.18), as well as the capacity limitations of each generator are provided in Table 12.

Table 12: Generation cost, emission, capacity coefficients

Generator	Cost			Emissions					Capacities (MW/h)	
	a	b	c	α	β	γ	δ	ϵ	P_{Gi}^{min}	P_{Gi}^{max}
1	10	200	100	4.091	-5.554	6.490	2.0e-4	2.857	0.05	0.5

2	10	150	120	2.543	-6.047	5.638	5.0e-4	3.333	0.05	0.6
3	20	180	40	4.258	-5.094	4.586	1.0e-6	8.000	0.05	1.0
4	10	100	60	5.326	-3.550	3.380	2.0e-3	2.000	0.05	1.2
5	20	180	40	4.258	-5.094	4.586	1.0e-6	8.000	0.05	0.6
6	10	150	100	6.131	-5.555	5.151	1.0e-5	6.667	0.05	1.0

The particle filtering-based multi-objective optimization algorithm proposed in this study is implemented in Matlab R2010b, and the experiments are conducted using an Intel Core 2 Duo E8600 Computer. The conducted experiments can be considered as two parts: 1) evaluating the performance of the proposed particle filtering-based multi-objective optimization algorithm (PFMO) through benchmarking against several algorithms studied in the literature; 2) evaluating the performance of the PFMO under different experimental settings (i.e., different number of particles, different number of iterations, different resampling rules) in terms of the modified convergence metric.

Part I: Results obtained via the PFMO vs results obtained via other algorithms

In this set of experiments, results obtained via the proposed algorithm for the case with 50 particles and 200 iterations, are benchmarked with results reported in the literature using different algorithms, which are Non-dominated Sorting Genetic Algorithm (NSGA) (Abdio, 2003), Niched Pareto Genetic Algorithm (NPGA) (Abdio, 2003), Strength Pareto Evolutionary Algorithm (SPEA) (Abdio, 2003), Fuzzy Clustering Particle Swarm Optimization Algorithm (FCPSO) (Agrawal et al., 2008), Multi-objective Particle Swarm Optimization Algorithm (MOPSO) (Abido, 2009), Modified Bacterial Foraging Algorithm (MBFA) (Hota et al., 2010), respectively. Figure 29 presents a

screenshot of the running algorithm, illustrating the evolution of the proposed PFMO algorithm.

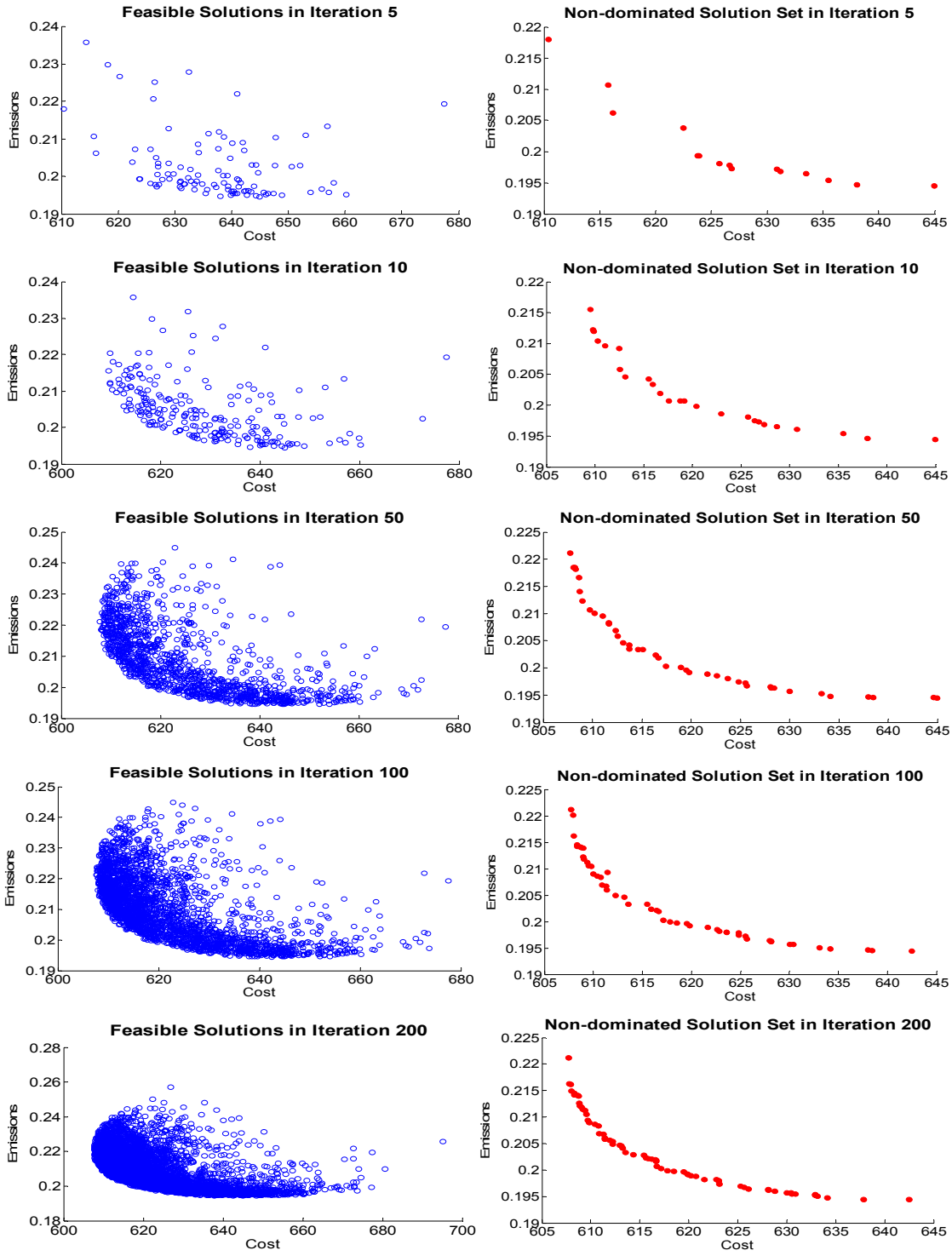


Figure 29: Obtained feasible solutions and non-dominated solutions in different iterations

The performances of these algorithms, in terms of the extreme best solutions (i.e., best generation cost and best emissions) and the best compromise solutions are compared in Tables 13-15, respectively. It can be concluded that the proposed particle filtering-based multi-objective optimization algorithm (PFMO) is very competitive against the other algorithms in obtaining both extreme best solutions, and the best compromise solutions.

Table 13: Comparison of best cost obtained via different optimization algorithms

Solutions	Best Cost						
	NSGA	NPGA	SPEA	FCPSO	MOPSO	MBFA	PFMO
P_{G1}	0.1168	0.1245	0.1086	0.1130	0.1207	0.1141	0.1061
P_{G2}	0.3165	0.2792	0.3056	0.3135	0.3131	0.3108	0.3025
P_{G3}	0.5441	0.6284	0.5818	0.5826	0.5907	0.5994	0.5962
P_{G4}	0.9447	1.0264	0.9846	0.9710	0.9769	0.9816	0.9842
P_{G5}	0.5498	0.4693	0.5288	0.5264	0.5155	0.5048	0.3697
P_{G6}	0.3964	0.3993	0.3584	0.3450	0.3504	0.3559	0.5070
Cost	**608.245**	**608.147**	**607.807**	**607.786**	**607.79**	**607.67**	**607.4310**
Emission	0.2166	0.2236	0.2202	0.2201	0.2193	0.2198	0.2202

Table 14: Comparison of best emissions obtained via different optimization algorithms

Solutions	Best Emissions						
	NSGA	NPGA	SPEA	FSPSO	MOPSO	MBFA	PFMO
P_{G1}	0.4113	0.3923	0.4043	0.4063	0.4101	0.4055	0.3830
P_{G2}	0.4591	0.4700	0.4525	0.4586	0.4594	0.4609	0.4485
P_{G3}	0.5117	0.5565	0.5525	0.5510	0.5511	0.5444	0.5660
P_{G4}	0.3724	0.3695	0.4079	0.4084	0.3919	0.3986	0.4221
P_{G5}	0.5810	0.5599	0.5468	0.5432	0.5413	0.5440	0.5180
P_{G6}	0.5304	0.5163	0.5005	0.4974	0.5111	0.5134	0.5249
Cost	647.2510	645.9840	642.6030	642.896	644.740	644.4300	640.3119
Emission	**0.1943**	**0.1942**	**0.1942**	**0.1942**	**0.1942**	**0.1942**	**0.1942**

Table 15: Comparison of best compromise solutions obtained via different optimization algorithms

Solutions	NSGA	NPGA	SPEA	MOPSO	PFMO
P_{G1}	0.2935	0.2976	0.3052	0.2367	0.2560
P_{G2}	0.3645	0.3956	0.4389	0.3616	0.3675
P_{G3}	0.5833	0.5673	0.7163	0.5887	0.5439
P_{G4}	0.6763	0.6928	0.6978	0.7041	0.6902
P_{G5}	0.5383	0.5201	0.1552	0.5635	0.4306
P_{G6}	0.4076	0.3904	0.5507	0.4087	0.5851
Cost	**617.80**	**617.79**	**629.59**	**615.00**	**616.5391**
Emission	**0.2002**	**0.2004**	**0.2079**	**0.2021**	**0.2009**

In addition to the comparison of the extreme best solutions obtained via different optimization algorithm, in this study, the performance of the proposed optimization algorithm is also demonstrated via the spread indicator. The spread (SP) indicator, also known as the diversity metric, is to evaluate the spread within the obtained non-dominated solution set (Deb et al., 2002). It is defined as shown in (5.25), where d_i is the Euclidean distance between neighboring points i and $i + 1$, $\bar{d}$ is the average of these distances, d_f and d_l represent the distance between the extreme solutions of the non-dominated solution set and boundary solutions, N indicates the size of the non-dominated solution set. A smaller value indicates a better diversity. According to the results presented in Table 16, the proposed PFMO outperforms the other algorithms in terms of the spread indicator. Performances of other algorithms in terms of the spread indicator are reported in the work conducted by Wu et al. (2010).

$$DM = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \tag{5.25}$$

Table 16: Comparison of SP-metric for different algorithms

	NSGA	NPGA	SPEA	MOPSO	PFMO
SP	0.6801	0.4281	0.3879	0.3242	0.3150

Part II: Results obtained via the PFMO under different experimental settings

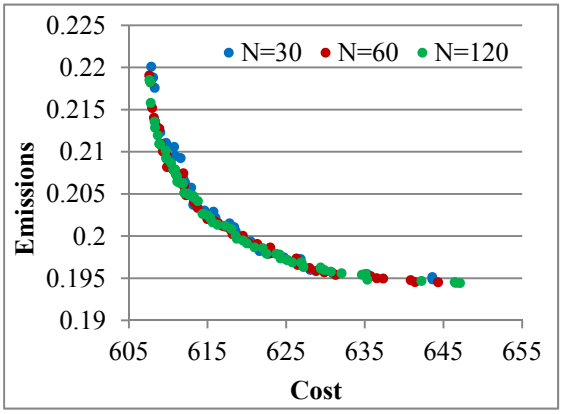
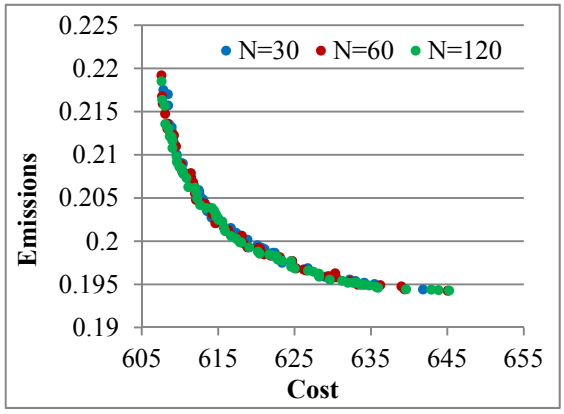
In this part, experiments are designed with various number of particles (i.e., $N = 30,60,120$) and iterations (i.e., $K = 50,100,200$). Moreover, the effectiveness of the two resampling rules embedded in the proposed algorithm is also tested here. The obtained non-dominated solutions are evaluated in terms of a modified convergence metric (MCM). The convergence metric is proposed by Khare et al. (2003), which is designed to evaluate the distance between the non-dominated solutions and the problem's global Pareto front. Generally, a smaller value indicates a better performance. However, since the global Pareto front in this EELD problem is unknown, in this study, we have modified the traditional definition of the CM as given in (5.26). In the modified version, P is the number of non-dominated solutions in the solution set, and d_i is the Euclidean distance between the solution i and its neighbor in terms of the objective values. Similarly, a smaller value is still preferable.

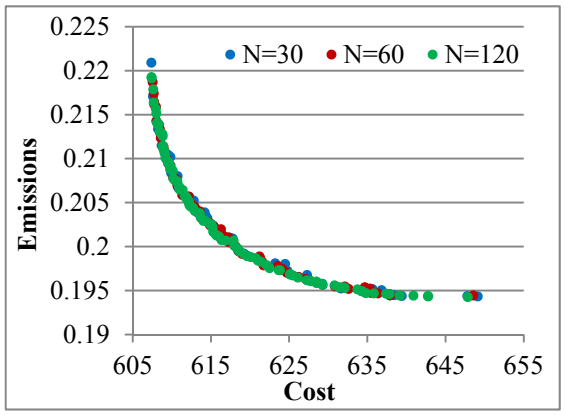
$$MCM = \frac{\sum_{i=1}^{P-1} d_i}{P} = \frac{\sum_{i=1}^{P-1}\left(\sqrt{(y_{i+1}^1 - y_i^1)^2 - (y_{i+1}^2 - y_i^2)^2}\right)}{P} \tag{5.26}$$

The non-dominated solution sets obtained for these cases and the corresponding values of the modified convergence metric are provided in Table 17, where the resultant

figures depicted that the proposed optimization algorithms have the capability to capture a great diversity of solutions. It should be also noticed here that the algorithm's performance on providing the extreme best solutions only slightly impacted by the changing number of particles, or the number of iterations. Therefore, it premises the potential to generate good results with reduced computational burden. However, the number of particles and the number of iterations do have positive impacts on the performance of the proposed algorithm if considered the convergence and diversity of the obtained non-dominated solution set.

Table 17: Comparison of non-dominated solution sets obtained with different number of particles and iterations

Number of Iterations	Non-Dominated Solution Set	Performance		
$K = 50$		$N = 30$	Best Cost	607.8350
			Best Emissions	0.1948
			MCM	1.1178
		$N = 60$	Best Cost	607.592
			Best Emissions	0.1945
			MCM	0.8346
		$N = 120$	Best Cost	607.6194
			Best Emissions	0.1943
			MCM	0.6369
$K = 100$		$N = 30$	Best Cost	607.8362
			Best Emissions	0.1944
			MCM	0.8289
		$N = 60$	Best Cost	607.5854
			Best Emissions	0.1943
			MCM	0.6937
		$N = 120$	Best Cost	607.6094
			Best Emissions	0.2185
			MCM	0.5534

$K = 200$		$N = 30$	Best Cost	607.3939
			Best Emissions	0.1943
			MCM	0.8348
		$N = 60$	Best Cost	607.5472
			Best Emissions	0.1944
			MCM	0.6119
		$N = 120$	Best Cost	607.4032
			Best Emissions	0.1942
			MCM	0.4046

Last but not least, experiments are also carried out to demonstrate the effectiveness of the resampling rules. As mentioned in Section 5.2.2, Chapter 5, two resampling rules are embedded in the proposed optimization algorithm to break the local optima. In the algorithm embedded the first resampling rule, resampling occurs in each iteration with a small portion ρ of the particles regenerated randomly, while in the algorithm embedded the second resampling rule, resampling is triggered when the condition shown in (5.16) is satisfied. Here, experiments are designed with different values of ρ and λ to investigate the impacts of the resampling procedures on the performance of the proposed optimization algorithm. The number of particles is set as 50, and the total number of iterations is 200. According to the results presented in Table 18, no significant differences are observed from these two resampling rules in terms of the obtained extreme best solutions and the convergence of the non-dominated solution set. Meanwhile, with a reasonable selection, different parameter settings of ρ for the first resampling rule and λ for the second resampling rule do not present obvious impacts on the performance of the proposed optimization algorithm. However, if ρ is set to a large number (e.g., $\rho = 90\%$), most of particles will be regenerated in each iteration.

Similarly, if λ is set to a small number (e.g., $\lambda = 0.35$), the resampling procedure will be frequently triggered. Both of these two situations will cease the convergence of the sampling distribution to a promising region; thereby degrade the performance of the proposed algorithm.

Table 18: Results obtained from the proposed algorithm embedding different resampling rules

		Best Cost	Best Emission	MCM
Resampling Rule 1	$\rho = 10\%$	607.5592	0.1942	0.4937
	$\rho = 15\%$	607.6308	0.1942	0.6100
	$\rho = 20\%$	607.7510	0.1944	0.5687
	$\rho = 90\%$	609.3385	0.1953	0.8429
Resampling Rule 2	$\lambda = 0.35$	608.2314	0.1954	0.9934
	$\lambda = 0.65$	607.5707	0.1943	0.6185
	$\lambda = 0.7$	607.4854	0.1942	0.5278
	$\lambda = 0.75$	607.5035	0.1942	0.5161

Chapter 6: Conclusions and Future Research

6.1. Conclusions

In this study, we have solved state estimation and optimization problems of large-scale dynamic systems through the use of improved particle filters. Theoretical contributions can be summarized as the derivation of the minimum relative entropy-based importance density selection scheme for improving the performance of the particle filters and the development of a particle filtering-based multi-objective optimization algorithm. Given the theoretical achievements, two real-world related state estimation problems and one multi-objective optimization problem are studied, which are short-term load demand forecasting, target tracking in the presence of multipath interference, and environmental and economic power dispatching, respectively.

In particular, in the first part of this study, a minimum relative entropy-based importance density selection scheme is proposed, considering the fact that the estimation of the posterior state relies on not only the previous state, but also on the newly arrived observation data. As opposed to the prior-sampling based particle filters, the filters incorporating this proposed scheme take both the prior and the likelihood into account when drawing samples. The proposed density selection scheme is first derived theoretically. Then, two sets of synthetic experiments are presented to test the feasibility and validity of the proposed rule. Furthermore, the validity of the proposed estimation scheme is demonstrated via a case study of short-term electricity demand forecasting for a private company located in Florida. The performance of the particle filters embedding the proposed density selection scheme and that of three other particle filters incorporating other density selection rules (i.e., PRIOR-PF, GSPF, and SPPF) are compared in terms of their estimation accuracies and computational efficiencies. Results obtained from the

experiments reveal that the proposed MREIS-PF outperforms the PRIOR-PF, GSPF, and SPPF in terms of their recorded RMSE values, especially in the cases where the observation has a strong impact on the posterior states. In addition, the initial size of particles has an insignificant effect on the performance of the MREIS-PF, and thus high-quality estimates can be provided with few samples, which in turn further reduce its computational burden. Therefore, utilization of this proposed scheme is particularly proved useful for the state estimation in large-scale dynamic systems due to its efficiency in computational resource usages.

In the second part of this study, a tracking mechanism consisting of a developed state-space model and a particle filtering estimation algorithm for low elevation target tracking over the sea-surface is proposed, in which the impacts of multipath propagation is considered from both the specular and diffuse reflection aspects. A series of real-world related cases are studied to illustrate the capability of the proposed tracking mechanism in the presence of multipath effects, and the performance of the embedded particle filtering algorithms (PF_P and PF_{PAL}) are evaluated against that of the unscented particle filtering algorithm (UPF). It is found that the tracking mechanism that embeds PF_{PAL} as its estimation algorithm outperforms the others (UPF and the PF_P) in most of the scenarios in terms of the estimations accuracies, since PF_{PAL} considers not only the prior but also the likelihood in its density selection as opposed to UPF and the PF_P. Given its superior performance, we are confident to say that the proposed tracking mechanism can be introduced as an upgrade to the existing radar systems, which would be a vital improvement in many ocean military applications including ocean surveillance, guided missile target locating, etc.

Last but not least, in the third part of this study, a novel particle filtering-based multi-objective optimization algorithm is proposed. Here, the multi-objective optimization problem is formulated as a filtering problem in which its objectives are recursively updated to approach the Pareto-optima given the prior states (i.e., candidate decision vectors) and a series of measurements (i.e., optimal objective values obtained in each iteration). This recursive updating process is achieved via two of the incorporated procedures, namely, the dynamic allocation procedure that changes the distribution of objective weights as the iterations progress and the performance-based sampling and resampling procedure that facilitates the generation of particles in each iteration. The feasibility and effectiveness of the proposed algorithm have been demonstrated on an economic and environmental power dispatch problem, where the performance of the proposed algorithm have been benchmarked against that of several algorithms investigated in the literature. Results obtained from a series of experiments have indicated that the proposed algorithm is capable of obtaining competitive non-dominated solution set without destroying the diversity of the solutions and occupying significant computational efforts.

6.2. Future Research

Future venues of this work include conducting continuous improvements on the particle filtering algorithm theoretically. First of all, further investigations will be sought after on the potential impact of the initial density function on the performance of the particle filters embedding the proposed density selection scheme, as the selection of the importance density relies not only on the product of the likelihood and the transition

prior, but also on the initial density function. Secondly, as the particle filters embedding the proposed density selection scheme fail to provide good estimation accuracy in cases where the noises are not addictive to the state evolution and measurement functions, efforts will be made to resolve this limitation. Moreover, since the selection of an appropriate resampling rule is also critical in achieving a desired accuracy level in all estimations, an optimum combination of the proposed importance density selection rule with different resampling rules may be investigated. One of my ongoing work focuses on a likelihood-base adaptive resampling rule, in which a decomposition method is utilized and the effective sample size will be calculated separately for each partition of the solution space to perform the resampling procedure more effectively. Particularly, the solution space is adaptively partitioned into different regions depending on the obtained particles, newly arrived measurements and corresponding likelihoods. Then, for each region k ($k = 1,2, \dots, K$), the effective sample size N_e^k will be calculated as described in Section 3.1, Chapter 3. A preliminary visualized diagram of the proposed resampling rule is provided in Figure 30. Moreover, modifying the noise levels in the measurement function is also an available instrument to mitigate the sample depletion. However, as the noise increases, the estimation accuracy of the particle filters decrease significantly. Therefore, efforts will be made to find a noise level that can obtain a trade-off balance between the estimation accuracy and degeneracy.

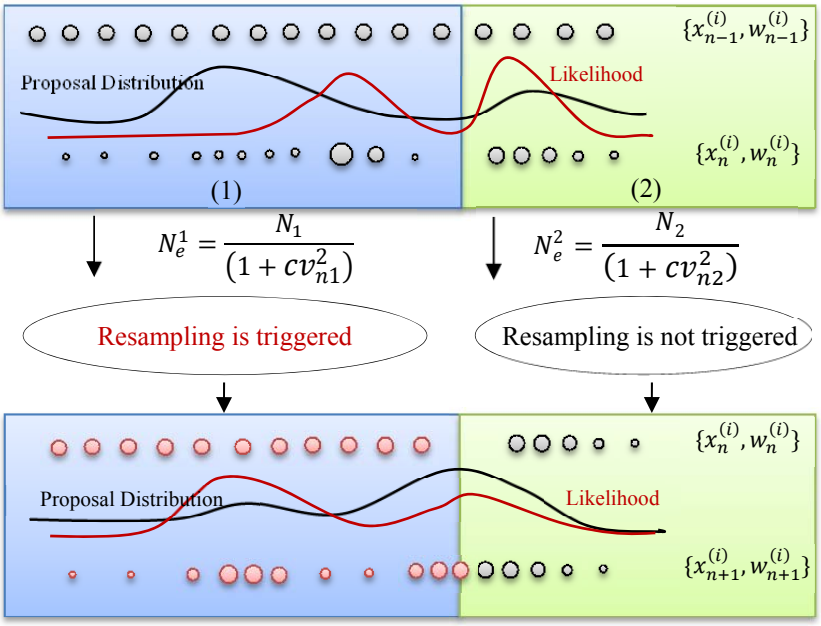


Figure 30: Visualized diagram of a likelihood-based adaptive resampling rule

The proposed particle filtering-based multi-objective optimization algorithm sheds light on developing more advanced optimization algorithm using methodologies that have been heavily studied in the area of Bayesian state estimation. In this context, a bridge that connects these two areas will be constructed. Meanwhile, there is a great room for improving the performance of the proposed optimization algorithm. It is noticed that sometimes the "best" particle obtained and evaluated via the DWA procedure does not belong to the non-dominated solution set. This may provide a misleading signal that inefficiently governs the evolution of the sampling distribution. From this perspective, I am currently working on a different procedure for evaluating the performance of the obtained particles, which may replace the dynamic weighted allocation procedure. The idea is to reformulate the multi-objective optimization problem with a different design of the state evolution and measurement function, in which the non-dominated solution obtained in each iteration will be utilized as the measurements. Once it is complete,

future venues may include an extension of the scope of its application to other popular multi-objective optimization problems.

References

Abido M.A., 2003. A Niched Pareto Genetic Algorithm for Environmental/Economic Power Dispatch, *Electric Power Systems Research*, 25(2): 97-105.

Abido M.A., 2003. A Novel Multiobjective Evolutionary Algorithm for Environmental Economic Power Dispatch, *Electric Power Systems Research*, 65(1): 71-91.

Abido M.A., 2003. Environmental Economic Power Dispatch using Multiobjective Evolutionary Algorithms, *IEEE Transactions on Power System*, 18(4): 1529-1537.

Abido M.A., 2009. Multiobjective Particle Swarm Optimization for Environmental Economic Dispatch Problem, *Electric Power Systems Research*, 79(7): 1105-1113.

Agrawal S., Panigrahi B.K., and Tiwari M.K., 2008. Multiobjective Particle Swarm Algorithm with Fuzzy Clustering for Electrical Power Dispatch, *IEEE Transactions on Evolutionary Computation*, 12(5): 529-541.

Ahn S., Yang E., Chun J., and Kim J., 2010. Low Angle Tracking using Iterative Multipath Cancellation in Sea Surface Environment, *2010 IEEE Radar Conference*, 1156-1160.

Antmann E.D., Shi X., Celik N., and Dai Y., 2011. Continuous-Discrete Simulation-based Decision Making Framework for Solid Waste Management and Recycling Programs, *Computers and Industrial Engineering*, 65(3): 438-454.

Arulampalam S., Maskell S., Gordon N., and Clapp T., 2002. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, 50(2): 174-189.

Beard C.I., 1961. Coherent and Incoherent Scattering of Microwaves from the Ocean, *IRE Transactions on Antennas and Propagation*, 9(5): 470-483.

Becerra R.L. and Coello C.C.A., 2006. Solving Hard Multiobjective Optimization Problem using ε-constraint with Cultured Differential Evolution, Computer Methods in Applied Mechanics and Engineering, 195(33): 4303-4322.

Beckmann P. and Spizzichino A., 1963. The Scattering of Electromagnetic Waves from Rough Surfaces, Pergamon Press, New York, United States.

Bhagwan D. and Patvardhan C., 1999. Solution of Economic Load Dispatch using Real Coded Hybrid Stochastic Search, *International Journal of Electrical Power & Energy Systems*, 21(3): 165-170.

Blair W.D. and Brandt-Pearce M., 2001. Statistics of Monopulse Measurements of Rayleigh Targets in the Presence of Specular and Diffuse Multipath, *In Proceedings of the 2001 IEEE Radar Conference*, 369-375.

Blake L.V., 1980. Radar Range-Performance Analysis, Artech House Radar Library, Chicago, United States.

Bloomberg Business, 2012. Company Overview of Sunburst Farms. Available online at: [http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=13412274], retrieved at February 15, 2012.

Bruder J.A. and Saffold J.A., 1991. Multipath Effects on Low-angel Tracking at Millimetre-wave Frequencies, *IEE Proceedings F: Radar and Signal Processing,* 138(2): 172-184.

Celik N. and Son Y.J., 2011. State Estimation of a Shop Floor using Improved Resampling Rules for Particle Filtering, *International Journal of Production Economics*, 134(1): 224-237.

Celik N., Sáenz J.P., and Shi X., 2012. Distributed Generation Penetration Optimization based on Particle Filtering, *In Proceedings of the 2012 Winter Simulation Conference*, 1-12.

Chen C.H. and Lee L.H., 2011. Stochastic Simulation Optimization: An Optimal Computing Budget Allocation, *World Scientific*, New York, United States.

Chen H., Liu X., Shi C., and Yao C., 2011. Power System Dynamic State Estimation based on a New Particle Filter, In Proceedings of the 2nd International Conference on Challenges in Environmental Science and Computer Engineering (CESCE), 11(B): 655-661.

Cheng Q. and Bordon P., 2008. A New Unscented Particle Filter, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3417-3420.

Chung M.S., Lim J.S., and Park D.C., 2006. Monopulse Tracking with Phased Array Search Radar in the Presence of Specular Reflection from Sea Surface, *IEEE Transactions on Aerospace and Electronic Systems*, 42(4): 1459-1463, 2006.

Clayton E.R., Weber E., and Taylor III B.W., 1982. A Goal Programming Approach to the Optimization of Multi-response Simulation Models, *IIE Transactions*, 14(4): 282-287.

Coello C.A. and Salazar L.M., 2002. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization, *In Proceedings of the 2002 IEEE Congress of Evolutionary Computation*, 1051-1056.

Coello C.A.C., 1999. A Comprehensive Survey of Evolutionary-based Multi-objective Optimization Techniques, *Knowledge and Information Systems*, 1(3): 269-308.

Collet C., Averty P., Dittmar A., 2009. Autonomic Nervous System and Subjective Ratings of Strain in Air Traffic Control, *Applied Ergonomics*, 40(1): 23-32.

Corne D.W., Knowles J.D., and Oates M.J., 2000. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization, *Parallel Problem Solving from Nature – PPSN VI*, Springer, Berlin, Germany.

Deb K., Pratap A., Agarwal S., and Meyarivan T., 2002. A Fast and Elitist Multiobjective Optimization Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2): 182-197.

Djurić P.M., Vemula M., and Bugallo M.F., 2008. Target Tracking by Particle Filtering in Binary Sensor Networks, *IEEE Transactions on Signal Processing*, 56(6): 2229-2238.

Donde V., Lopez V., Lesieutre B., Pinar A., Yang C., and Meza J., 2008. Sever Multiple Contingency Screening in Electric Power Systems, *IEEE Transactions on Power Systems*, 23(2): 406-417.

Doucet A. and Johansen A.M., 2008. A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later, *In Handbook of Nonlinear Filtering*, Cambridge: Cambridge University Press.

Doucet A., Godsill S., and Andrieu C., 2000. On Sequential Simulation-based Methods for Bayesian Filtering, Statistics and Computing, 10: 197-208.

E Source, 2012. Managing Energy Costs in Office Building. Available online at: [https://www.nationalgridus.com/non_html/shared_energyeff_office.pdf], retrieved at February 10, 2012.

Eusuff M. and Lansey K., 2003. Optimization of Water Distribution Network Design using the Shuffled Frog Leaping Algorithm, *Journal of Waste Resources Planning and Management*, 129(3): 210-225.

Fieldsend J.E. and Singh S., 2002. A Multi-objective Algorithm based upon Particle Swarm Optimization, an Efficient Data Structure and Turbulence, *In Proceedings of the 2002 UK Workshop on Computational Intelligence*, 24-44.

Figy S., 2011. FM Issue: Cutting Costs with Monthly Utility Bill Analysis. Available online at: [http://www.todaysfacilitymanager.com/articles/fm-issue-cutting-costs-with-monthly-utility-bill-analysis.php], retrieved at February 8, 2012.

Florida Power & Light Company, 2012. Special Electric Service Rate for Businesses. Available online at: [http://www.fpl.com/landing/pdf/special_rate.pdf], retrieved at February 7, 2012.

Fonseca C. and Fleming P.J., 1993. Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion, Generalization, *In Proceedings of the 5th International Conference on Genetic Algorithms*, 416-423.

Germann U., Galli G., Boscacci M., and Bolliger M., 2006. Radar Precipitation Measurement in a Mountainous Region, *Quarterly Journal of the Royal Meteorological Society, Part A*, 132(168): 1669-1692.

Gordon N.J., Salmond D.J., and Smith A.F.M., 1993. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, *IEE-Proceedings-F*, 140(2): 107-113.

Grainger J.J. and Stevenson W.D., 1994. Power System Analysis, McGraw Hill, Inc., New York, United States.

Hagan M.T. and Behr S.M., 1987. The Time Series Approach to Short Term Load Forecasting, *IEEE Transactions on Power Systems*, 2(3): 785-791.

Hajela P. and Lin C., 1992. Genetic Search Strategies in Multicriterion Optimal Design, *Structural Optimization*, 4(2): 99-107.

Han X., Zhang H., Meng H., and Wang X., 2010. Hybrid Method of DOA Estimation for Low-angle Target Tracking, *2010 International Conference on Electrical and Control Engineering (ICECE)*, 4687-4690.

Harvey A. and Koopman S.J., 1993. Forecasting Hourly Electricity Demand using Time-varying Splines, *Journal of the American Statistical Association*, 88(424): 1228-1236.

Horn J., Nafpliotis N., and Goldberg D.E., 1994. A Niched Pareto Genetic Algorithm for Multi-objective Optimization, *In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 1: 82-87.

Hota P.K., Barisal A.K., and Chakrabarti R., 2010. Economic Emission Load Dispatch through Fuzzy based Bacterial Foraging Algorithm, *Electrical Power and Energy Systems*, 32(7): 794-803.

Huang G.M., Lei J., and Abur A., 2003. A Heuristic Approach for Power System Measurement Placement Design, *In Proceedings of the 2003 International Symposium on Circuits and Systems*, 3:407-410.

Inaba T. and Araki K., 2004. Space-Frequency Maximal Ratio Combining for Low-Elevation Radar Target, *Electronics and Communications in Japan, Part I: Communications*, 85(5): 75-85.

Jao J.K., 1994. A Matched Array Beamforming Technique for Low Angle Radar Tracking in Multipath, *Record of the 1994 IEEE National Radar Conference*, 171-176.

Jatoth R.K., Kumar R., and Kumar T.K., 2013. Performance Comparison of Hybrid GA-PSO based Tuned IMMS for Maneuver Target Tracking, *International Journal of Intelligent Systems and Applications*, 5(12): 120-134.

Ji C., Zhang Y., Tong M., and Yang S., 2008. Particle Filter with Swarm Move for Optimization, *In Proceedings of 10th International Conference on Parallel Problem Solving From Nature (PPSN)*, 909-918.

Jin Y., Olhofer M., and Sendhoff B., 2001. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How? *In Proceedings of Genetic and Evolutionary Computation Conference*, 1042-1049.

Jishy K. and Lehmann F., 2009. Particle Filter Algorithm for Tracking Manoeuvring Target using FM Signal, *IEEE International Symposium on Signal Processing and Information Technology*, 355-359.

Johansen A.M. and Doucet A., 2008. A Note on Auxiliary Particle Filters, *Statistics and Probability Letters*, 78(12): 1498-1504.

Juberias G., Yunta R., Garcia M.J., and Mendivil C., 1999. A New ARIMA Model for Hourly Load Forecasting, *In Proceedings of IEEE Transmission and Distribution Conference*, 1: 314-319.

Kehoe B., Warrier D., Patil S., and Goldberg K., 2014. Cloud-based Grasp Analysis and Planning for Toleranced Parts using Parallelized Monte Carlo Sampling," *IEEE Transactions on Automation Science and Engineering*, in press.

Kennedy J. and Eberhart R.C., 1995. Particle Swarm Optimization, *In Proceedings of International Conference on Neural Networks*, 1942-1948.

Khan Z., Balch T., and Dellaert F., 2005. MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11): 1805-1918.

Khare V.R., Yao X., and Deb K., 2003. Performance Scaling of Multi-objective Evolutionary Algorithms, Lecture Notes in Computer Science 2632: Evolutionary Multi-Criterion Optimization, 367-390, Springer, Berlin, Germany.

Kollias P., Clothiaux E.E., Miller M.A., Albrecht B.A., Stephens G.L., and Ackerman T.P., 2007. Millimeter-Wavelength Radars: New Frontier in Atmospheric Cloud and Precipitation Research, *Bulletin of the American Meteorological Society*, 88: 1608–1624.

Kotecha J.H. and Djuric P.M., 2003. Gaussian Particle Filtering, *IEEE Transactions on Signal Processing*, 51(10): 2592-2601.

Kramer K.A. and Stubberud S.C., 2008. Tracking of Multiple Target Types with a Single Neural Extended Kalman Filter, *Studies in Computational Intelligence (SCI)*, 109: 495-512.

Lamedica R., Prudenzi A., Sforna M., Caciotta M., and Cencellli V.O., 1996. A Neural Network based Technique for Short-term Forecasting of Anomalous Load Periods, *IEEE Transactions on Power Systems*, 11(4): 1749-1756.

Le Berre M., Hnaien F., and Snoussi H., 2011. Multi-objective Optimization in Wireless Sensors Networks, *2011 International Conference on Microelectronics (ICM)*, 1-4.

Li X.R. and Bar-Shalom Y., 1993. Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking, IEEE Transactions on Control Systems Technology, 1(3): 186-194.

Liu J.S. and Chen R., 1995. Blind Deconvolution via Sequential Imputations, *Journal of the American Statistical Association*, 90(430): 567-576.

Lopes H.F. and Tsay R.S., 2011. Particle Filters and Bayesian Inference in Financial Econometrics, *Journal of Forecasting*, 30(1): 168-209.

Lopez-Ibanez M. and Stuetzle T., 2012. The Automatic Design of Multi-Objective Ant Colony Optimization Algorithms, *IEEE Transactions on Evolutionary Computation*, 16(6): 861-875.

Mavrotas G., 2009. Effective Implementation of the ε-constraint Method in Multi-Objective Mathematical Programming Problems, *Applied Mathematics and Computation*, 213: 455-465.

Melzi M., Ouldali A., and Messaoudi Z., 2010. Multiple Target Tracking using the Extended Kalman Particle Probability Hypothesis Density Filter, *In Proceedings of the 18th European Signal Processing Conference*, 1821-1826.

Mirasgedis S., Sarafidis Y., Georgopoulou E., Kotroni V., Lagouvardos K., and Lalas D.P., 2007. Modeling Framework for Estimating Impacts of Climate Change on Electricity Demand at Regional Level: Case of Greece, *Energy Conversion and Management*, 48(5): 1737-1750.

Monson C.K. and Seppi K.D., 2004. The Kalman Swarm, *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 140-150.

Moral P.D., Doucet A., and Jasra A., 2006. Sequential Monte Carlo Samplers, *Journal of the Royal Statistical Society*, 68(3): 411-436.

Ngatchou P., Zarei A., and El-Sharkawi M.A., 2005. Pareto Multi Objective Optimization, *In Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, 84-91.

Okuda S., Arai Y., and Kouguchi N., 2005. Sea Surface Wave Information using GPS Sea Reflected Signal-Wave Height, *In Proceedings of the 18th International Technical Meeting of the Satellite Division*, 810-815.

Okuma K., Taleghani A., de Freitas N., Little J.J., and Lowe D.G., 2004. A Boosted Particle Filter: Multi-Target Detection and Tracking, *In Proceedings of the European Conference on Computer Vision*, 28-39.

Panigrahi B.K., Pandi V.R., Sharma R., and Das S., 2011. Multiobjective Bacteria Foraging Algorithm for Electrical Load Dispatch Problem, *Energy Conversion and Management*, 52(2): 1334-1342.

Pardo A., Meneu V., and Valor E., 2002. Temperature and Seasonality Influences on Spanish Electricity Load, *Energy Economics*, 24(1): 55-70.

Pardoe D. and Stone P., 2007. An Autonomous Agent for Supply Chain Management, *Handbooks in Information Systems Series: Business Computing*, 141–72, Elsevier, Bingley, United Kingdom.

Pitt M.K. and Shephard N., 1999, Filtering via Simulation: Auxiliary Particle Filters, *Journal of the American Statistical Association*, 94 (446): 590-599.

Qing Y., Xiong W., and Gang Li, 2010. Multi-Population Binary Ant Colony Algorithm with Concrete Behaviors for Multi-Objective Optimization Problem, *In Proceedings of the 2nd IEEE International Conference on Information Management and Engineering*, 274-281.

Radebaugh J., Lorenz R.D., Kirk R.L., Lunine J.I., Stofan E.R., Lopes R.M.C., Wall S.D., 2007. Mountains on Titan Observed by Cassini Radar, Icarus, 192(1): 77-91.

Ramachandra K.V., Mohan B.R., and Geetha B.R., 1993. A Three-State Kalman Tracker using Position and Rate Measurements, *IEEE Transactions on Aerospace and Electronic Systems*, 29(1): 215-222.

Raquel C.R. and Nava P.C. Jr., 2005. An Effective Use of Crowding Distance in Multi-objective Particle Swarm Optimization, *In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, 257-264.

Renfrew M., Bai Z., and Cavusoglu M.C., 2013. Particle Filter based Active Localization of Target and Needle in Robotic Image-Guided Intervention Systems, *2013 IEEE Conference on Automation Science and Engineering (CASE)*, 448-454.

Sakawa M., Yano H., and Yumine T., 1987. An Interactive Fuzzy Satisficing Method for Multiobjective Linear-programming Problems and its Application, *IEEE Transactions on Systems, Man, and Cybernetics*, 17(4): 654-661.

Schaffer D., 1985. Multiple Optimization with Vector Evaluated Genetic Algorithms, *In Proceedings of the 1st International Conference on Genetic Algorithms*, 93-100.

Schirru A., Pampuri S., and de Nicolao G., 2010. Particle Filtering of Hidden Gamma Processes for Robust Predictive Maintenance in Semiconductor Manufacturing, *2010 IEEE Conference on Automation Science and Engineering (CASE)*, 51-56.

Sen S., Tang G., and Nehorai A., 2011. Multi-objective Optimization of OFDM Radar Waveform for Target Detection, *IEEE Transactions on Signal Processing*, 59(2): 639-652.

Sherman S.M., 1971. Complex Indicated Angles Applied to Unresolved Targets and Multipath, *IEEE Transactions on Aerospace and Electronic Systems*, AES-7(1): 160-170.

Sinha A., Bar-Shalom Y., Blair W.D., and Kirubarajan T., 2003. Radar Measurement Extraction in the Presence of Sea-surface Multipath, *IEEE Transactions on Aerospace and Electronic Systems*, 39(2): 550-567.

Sisson S.A., Fan Y., and Tanaka M.M., 2007, Sequential Monte Carlo without Likelihoods, *In Proceedings of the National Academy of Sciences of the United States of America*, 104(6): 1760-1765.

Specht D.F., 1991. A General Regression Neural Network, *IEEE Transactions on Neural Networks*, 2(6), 568-576.

Srinivas N. and Deb K., 1994. Multi-objective Function Optimization using Non-dominated Sorting Genetic Algorithms, *Evolutionary Computation*, 2 (3): 221-248.

Tabucanon M.T., 1988. Multiple Criteria Decision Making Industry, Elsevier Science Publishers, Amsterdam, Netherlands.

Tamiz M., Jones D., and Romero C., 1998. Goal Programming for Decision Making: An Overview of the Current State-of-the-Art, *European Journal of Operational Research*, 111(3): 569-581.

Tay S., Maussang F., Coatanhay A., Chonavel T. and Garello R., 2011. Near Sea Surface Target Tracking by Extended Kalman Filtering of the GPS Reflected Signals, *In Proceedings of IEEE Oceans*, 1-4.

Thanos A.E., Moore D., Shi X., and Celik N., 2014. System of Systems Modeling and Simulation for Microgrids using DDDAMS, *Modeling and Simulation Support for System of Systems Engineering Applications*, Wiley and Sons Inc., Hoboken, New York, USA.

Tiwari S., Koch P., Fadel G., Deb K., 2008. AMGA: an Archive-based Micro Genetic Algorithm for Multi-objective Optimization, *In Proceedings of Genetic and Evolutionary Computation Conference*, 729-736.

Toffolo A., and Lazzaretto A., 2002. Evolutionary Algorithms for Multi-objective Energetic and Economic Optimization in the Thermal System Design, *Energy*, 27(6): 549-567.

Translation Bureau, 2013. Radar Definition, Public Works and Government Services Canada. Available online at: [http://www.btb.termiumplus.gc.ca/tpv2alpha/alpha-fra.html?lang=fra&i=1&index=ent&index=srchtxt=radar&comencsrch.x=0&comencsrch.y=0], retrieved December 10, 2013.

University of Washington, 2012. Power Systems Test Case Archive – UWEE. Available online at: [http://www.ee.washington.edu/research/pstca/], retrieved March 13, 2014.

Valor E., Menue V., and Caselles V., 2001. Daily Air Temperature and Electricity Load in Spain, *Journal of Applied Meteorology*, 40: 1413-1421.

van der Merwe R., de Freitas N., Doucet A., and Wan E., 2001. The Unscented Particle Filter, *In Proceedings of the Neural Information Processing Systems Conference*, 584-590.

Weber J.D., 1997. Implementation of a Newton-based Optimal Power Flow into a Power System Simulation Environment, Master thesis, University of Illinois at Urbana-Champaign, Illinois, United States.

Wen Q. and Qiong P., 2007. An Improved Particle Filter Algorithm based on Neural Network for Target Tracking, *Intelligent Information Processing III: International Federation for Information Processing (IFIP)*, 228: 297–305.

Wu Z., and Su T., 2010. Radar Target Detect using Particle Filter, *In Proceedings of the 2010 IEEE Radar Conference*, 955-958.

Wu L.H., Wang Y.N., Yuan X.F., Zhou S.W., 2010. Environmental/Economic Power Dispatch Problem using Multi-objective Differential Evolution Algorithm, *Electric Power Systems Research*, 80: 1171-1181.

L.H. Wu[a,*], Y.N. Wang[a], X.F. Yuan[a], S.W. Zhou[b]

Yedukondalu K., Sarma A.D., and Srinivas V.S., 2011. Estimation and Mitigation of GPS Multipath Interference using Adaptive Filtering, *Progress in Electromagnetics Research M*, 21: 133-148.

Zhang Q. and Li H., 2007. MOEA/D: A Multi-objective Evolutionary Algorithm based on Decomposition, *IEEE Transactions on Evolutionary Computation*, 11: 712-731.

Zhou E., Fu M., and Marcus S. I., 2008. A Particle Filtering for Randomized Optimization Algorithms, *In Proceedings of the 40th Conference on Winter Simulation*, 647-654.

Zitzler E. and Thiele L., 1999. Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, 3(4): 257-271.

Zitzler E., Laumanns M., and Thiele L., 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems*, International Center for Numerical Methods in Engineering, Barcelona, Spain.

Appendix I: Proof to Obtain a Closed Form Formula of the Relative Entropy between $q(x)$ and $f(x)$

The function of relative entropy between $q(x)$ and $f(x)$ is presented as $g(\omega,\eta) = \omega ln(\omega/\eta)$ by treating the density functions $q(x)$ and $f(x)$ as variables ω and η. Then, the Taylor series expansion of $g(\omega,\eta)$ for the neighborhood of point (a,b) can be written as in (1),

$$\begin{aligned} g(\omega,\eta) &= g(a,b) + \tfrac{1}{1!}\left[d\omega g_{\omega}(a,b) + d\eta g_{\eta}(a,b)\right] \\ &\quad + \tfrac{1}{2!}\left[(d\omega)^2 g_{\omega\omega}(a,b) + 2g_{\omega\eta}(a,b)d\omega d\eta + (d\eta)^2 g_{\eta\eta}(a,b)\right] \\ &\quad + \cdots + \tfrac{1}{n!}\left[\textstyle\sum_{k=0}^{n}\binom{n}{k}\frac{\partial^n g}{\partial\omega^{n-k}\partial\eta^k}|_{(a,b)}(d\omega)^{n-k}(d\eta)^k\right] \\ &= aln(a/b) + (1 + lna - lnb)(\omega - a) + \left(-\tfrac{a}{b}\right)(\eta - b) \\ &\quad + \tfrac{1}{2!}[\tfrac{1}{a}(\omega - a)^2 + \left(-\tfrac{2}{b}\right)(\omega - a)(\eta - b) + \tfrac{a}{b^2}(\eta - b)^2] + \varepsilon \\ &= \omega + \omega lna - \omega lnb - \tfrac{a}{b}\eta + \tfrac{1}{2a}\omega^2 - \tfrac{1}{b}\omega\eta + \tfrac{a}{2b^2}\eta^2 + \varepsilon \end{aligned} \quad (1)$$

At this point, the closed form of the relative entropy between $q(x)$ and $f(x)$ can be represented as in (2) by replacing the neighborhood of point (a,b) with $(\vartheta(x_0)$,$\mu(x_0))$ and variables ω and η with the density functions $q(x)$ and $f(x)$.

$$D_{KL}(Q||F) \approx \textstyle\sum_x \left\{\frac{q(x)^2}{2\vartheta(x_0)} + q(x)\left[1 + ln\frac{\vartheta(x_0)}{\mu(x_0)} - \frac{f(x)}{\mu(x_0)}\right] + \frac{\vartheta(x_0)f(x)^2 - \vartheta(x_0)\mu(x_0)q(x)}{2\mu(x_0)^2}\right\} \quad (2)$$

Similar to the derivation conducted in Chapter 3, Section 3.2.3, the equation presented in (2) are set to zero to obtain the minimum value of the relative entropy. By solving (3), the $q(x)$ can be presented as a function of $f(x)$, $\vartheta(x_0)$, and the initial density $\mu(x_0)$ as shown in (4).

$$\frac{q(x)^2}{2\vartheta(x_0)} + q(x)\left[1 + ln\frac{\vartheta(x_0)}{\mu(x_0)} - \frac{f(x)}{\mu(x_0)}\right] + \frac{\vartheta(x_0)f(x)^2 - \vartheta(x_0)\mu(x_0)f(x)}{2\mu(x_0)^2} = 0 \quad (3)$$

$$q(x) = \frac{-\left[1+ln\frac{\vartheta(x_0)}{\mu(x_0)} - \frac{f(x)}{\mu(x_0)}\right] \pm \sqrt{\left[1+ln\frac{\vartheta(x_0)}{\mu(x_0)} - \frac{f(x)}{\mu(x_0)}\right]^2 - \frac{f(x)^2 - \mu(x_0)f(x)}{\mu(x_0)^2}}}{\frac{1}{\vartheta(x_0)}} \quad (4)$$

Then, the density function $\vartheta(x_0)$ can be obtained by setting $\left[1 + \ln\frac{\vartheta(x_0)}{\mu(x_0)} - \frac{f(x)}{\mu(x_0)}\right]^2 - \frac{f(x)^2 - \mu(x_0)f(x)}{\mu(x_0)^2} = 0$, as shown in (5). Here, before substituting (5) into (4), we have to make efforts on checking the feasibility of the quadratic and determining whether the positive or the negative value obtained from the quadratic should be selected. Once it is done, the expression of the $q(x)$ obtained becomes extremely complex, making it very difficult to draw samples from.

$$\vartheta(x_0) = \mu(x_0)e^{\pm\sqrt{\frac{f(x) - \mu(x_0)f(x)}{\mu(x_0)^2}} + \frac{f(x)}{\mu(x_0)} - 1} \quad (5)$$

Appendix II: OCBA Matlab Code

```
function [ groupIndex, outIndex ] = resample_ocba( s_mean, s_var, nd,
n, add_budget, an, type)

%This function determines the allocation of the samples drawn in the
resampling procedure
%s_mean[i]: sample mean of design i, i=0,1,...,ND-1
%s_var[i]: sample variance of design i, i=0,1,...,ND-1
%nd: the number of designs
%n[i]: number of simulation replications of design i, i=0,1,...,ND-1
%add_budget; the additional simulation budget
%an[i]: additional number of simulation replications assigned to design
i, i=0,1,...,ND-1
%type: type of opitmization problem. type=1, MIN problem; type=2, MAX
problem

% ------------------defining variables and initial setting----------

morerun = zeros(1,nd);
t_s_mean = zeros(1,nd);
ratio = ones(1,nd);

%---------------------------------------------------------------------
if(type==1) %MIN problem
    for i=1:nd,
        t_s_mean(i)=s_mean(i);
    end
else %MAX problem
    for i=1:nd,
        t_s_mean(i)=(-1)*s_mean(i);
    end
end

t_budget=add_budget;

for i=1:nd,
    t_budget=t_budget+n(i);
end

b=best(t_s_mean,nd);  % call function "best"
s=second_best(t_s_mean,nd,b);  % call function "second_best"

ratio(s)=1.0;

for i = 1:nd
    if(i~=s && i~=b)
        temp =(t_s_mean(b)-t_s_mean(s))./(t_s_mean(b)-t_s_mean(i));
        ratio(i)=temp.*temp.*s_var(i)./s_var(s);
    end
end
% calculate ratio of Ni/Ns

temp=0;
for i = 1:nd
    if(i~=b)
        temp = temp + ratio(i).*ratio(i)./s_var(i);
```

```
    end
end

ratio(b)=sqrt(s_var(b).*temp); % calculate NB

for i = 1:nd
    t1_budget=t_budget;
end

while (more_alloc)
    more_alloc=0;
    ratio_s=0.0;
    for i = 1:nd,
        if(morerun(i))
            ratio_s = ratio_s + ratio(i);
        end
    end

    for i = 1:nd,
        if(morerun(i))
            an(i) = t1_budget./ratio_s.*ratio(i);
            if(an(i)<n(i))
                an(i)=n(i);
                morerun(i)=0;
                more_alloc=1;
            end
        end
        if(more_alloc)
            t1_budget=t_budget;
            for j = 1:nd
                if(~morerun(j))
                     t1_budget = t1_budget-an(j);
                end
            end
        end
   end
end %end of WHILE

t1_budget=an(0);
for i = 1:nd
    t1_budget = t1_budget+an(i);
end

an(b) = an(b)+t_budget-t1_budget; % give the difference to design b

for i = 1:nd
    an(i)= an(i)-n(i);
end
```

Appendix III: IEEE-30 Bus Load Data

Bus	Type	Vsp	theta	PG_i	QG_i	PL_i	QL_i	Q_{min}	Q_{max}
1	1	1.060	0	0	0	0	0	0	0
2	2	1.043	0	0	0	21.7	12.7	-40	50
3	3	1.021	0	0	0	2.4	1.2	0	0
4	3	1.012	0	0	0	7.6	1.6	0	0
5	2	1.010	0	0	0	94.2	19	-40	40
6	3	1.010	0	0	0	0	0	0	0
7	3	1.002	0	0	0	22.8	10.9	0	0
8	2	1.010	0	0	0	30	30	-10	40
9	3	1.051	0	0	0	0	0	0	0
10	3	1.045	0	0	0.19	5.8	2	0	0
11	2	1.082	0	0	0	0	0	-6	24
12	3	1.057	0	0	0	11.2	7.5	0	0
13	2	1.071	0	0	0	0	0	-6	24
14	3	1.042	0	0	0	6.2	1.6	0	0
15	3	1.038	0	0	0	8.2	2.5	0	0
16	3	1.045	0	0	0	3.5	1.8	0	0
17	3	1.040	0	0	0	9	5.8	0	0
18	3	1.028	0	0	0	3.2	0.9	0	0
19	3	1.026	0	0	0	9.5	3.4	0	0
20	3	1.030	0	0	0	2.2	0.7	0	0
21	3	1.033	0	0	0	17.5	11.2	0	0
22	3	1.033	0	0	0	0	0	0	0
23	3	1.027	0	0	0	3.2	1.6	0	0
24	3	1.021	0	0	0.043	8.7	6.7	0	0
25	3	1.017	0	0	0	0	0	0	0
26	3	1.000	0	0	0	3.5	2.3	0	0
27	3	1.023	0	0	0	0	0	0	0
28	3	1.007	0	0	0	0	0	0	0
29	3	1.003	0	0	0	2.4	0.9	0	0
30	3	0.992	0	0	0	10.6	1.9	0	0

*Type: 1 - Slack bus
2 - PV bus
3 - PQ bus

Appendix IV: IEEE-30 Bus Line Data

Line #	From Bus	To Bus	Resistance (pu)	Reactance (pu)	Susceptance (pu)	Rating (MVA)	B/2 (pu)	X'mer Tap (a)
1	1	2	0.0192	0.0575	0.0264	130	0.0264	1
2	1	3	0.0452	0.1852	0.0204	130	0.0204	1
3	2	4	0.0570	0.1737	0.0184	65	0.0184	1
4	3	4	0.0132	0.0379	0.0042	130	0.0042	1
5	2	5	0.0472	0.1983	0.0209	130	0.0209	1
6	2	6	0.0581	0.1763	0.0187	65	0.0187	1
7	4	6	0.0119	0.0414	0.0045	90	0.0045	1
8	5	7	0.0460	0.1160	0.0102	70	0.0102	1
9	6	7	0.0267	0.0820	0.0085	130	0.0085	1
10	6	8	0.0120	0.0420	0.0045	32	0.0045	1
11	6	9	0.0000	0.2080	0.0000	65	0.0000	0.978
12	6	10	0.0000	0.5560	0.0000	32	0.0000	0.969
13	9	11	0.0000	0.2080	0.0000	65	0.0000	1
14	9	10	0.0000	0.1100	0.0000	65	0.0000	1
15	4	12	0.0000	0.2560	0.0000	65	0.0000	0.932
16	12	13	0.0000	0.1400	0.0000	65	0.0000	1
17	12	14	0.1231	0.2559	0.0000	32	0.0000	1
18	12	15	0.0662	0.1304	0.0000	32	0.0000	1
19	12	16	0.0945	0.1987	0.0000	16	0.0000	1
20	14	15	0.2210	0.1997	0.0000	16	0.0000	1
21	16	17	0.0824	0.1923	0.0000	16	0.0000	1
22	15	18	0.1070	0.2185	0.0000	16	0.0000	1
23	18	19	0.0639	0.1292	0.0000	16	0.0000	1
24	19	20	0.0340	0.0680	0.0000	16	0.0000	1
25	10	20	0.0936	0.2090	0.0000	32	0.0000	1
26	10	17	0.0324	0.0845	0.0000	32	0.0000	1
27	10	21	0.0348	0.0749	0.0000	32	0.0000	1
28	10	22	0.0727	0.1499	0.0000	32	0.0000	1
29	21	22	0.0116	0.0236	0.0000	32	0.0000	1
30	15	23	0.1000	0.2020	0.0000	16	0.0000	1
31	22	24	0.1150	0.1790	0.0000	16	0.0000	1
32	23	24	0.1320	0.2700	0.0000	16	0.0000	1
33	24	25	0.1885	0.3292	0.0000	16	0.0000	1
34	25	26	0.2544	0.3800	0.0000	16	0.0000	1
35	25	27	0.1093	0.2087	0.0000	16	0.0000	1
36	28	27	0.0000	0.3960	0.0000	65	0.0000	0.968
37	27	29	0.2198	0.4153	0.0000	16	0.0000	1
38	27	30	0.3202	0.6027	0.0000	16	0.0000	1
39	29	30	0.2399	0.4533	0.0000	16	0.0000	1
40	8	28	0.0636	0.2000	0.0214	32	0.0214	1
41	6	28	0.0169	0.0599	0.0065	32	0.0065	1