# High-level Information Fusion for Constrained SMC Methods and Applications

by

Sepideh Seifzadeh

A thesis
presented to University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

## Abstract

Information Fusion is a field that studies processes utilizing data from various input sources, and techniques exploiting this data to produce estimates and knowledge about objects and situations. On the other hand, human computation is a new and evolving research area that uses human intelligence to solve computational problems that are beyond the scope of existing artificial intelligence algorithms. In previous systems, humans' role was mostly restricted for analysing a finished fusion product; however, in the current systems the role of humans is an integral element in a distributed framework, where many tasks can be accomplished by either humans or machines. Moreover, some information can be provided only by humans not machines, because the observational capabilities and opportunities for traditional electronic (hard) sensors are limited.

A source-reliability-adaptive distributed non-linear estimation method applicable to a number of distributed state estimation problems is proposed. The proposed method requires only local data exchange among neighbouring sensor nodes. It therefore provides enhanced reliability, scalability, and ease of deployment. In particular, by taking into account the estimation reliability of each sensor node at any point in time, it yields a more robust distributed estimation. To perform the Multi-Model Particle Filtering (MMPF) in an adaptive distributed manner, a Gaussian approximation of the particle cloud obtained at each sensor node, along with a weighted Consensus Propagation (CP)-based distributed data aggregation scheme, are deployed to dynamically re-weight the particle clouds.

The filtering is a soft-data-constrained variant of multi-model particle filter, and is capable of processing both soft human-generated data and conventional hard sensory data. If permanent noise occurs in the estimation provided by a sensor node, due to either a faulty sensing device or misleading soft data, the contribution of that node in the weighted consensus process is immediately reduced in order to alleviate its effect on the estimation provided by the neighbouring nodes and the entire network. The robustness of the proposed source-reliability-adaptive distributed estimation method is demonstrated through simulation results for agile target tracking scenarios. Agility here refers to cases in which the observed dynamics of targets deviate from the given probabilistic characterization.

Furthermore, the same concept is applied to model soft data constrained multiple-model Probability Hypothesis Density (PHD) filter that can track agile multiple targets with non-linear dynamics, which is a challenging problem. In this case, a Sequential Monte Carlo-Probability Hypothesis Density (SMC-PHD) filter deploys a Random Set (RS) theoretic formulation, along with Sequential Monte Carlo approximation, a variant of Bayes filtering. In general, the performance of Bayesian filtering-based methods can be enhanced by using

extra information incorporated as specific constraints into the filtering process. Following the same principle, the new approach uses a constrained variant of the SMC-PHD filter, in which a fuzzy logic approach is used to transform the inherently vague human-generated data into a set of constraints. These constraints are then enforced on the filtering process by applying them as coefficients to the particles' weights. Because the human generated Soft Data (SD), reports on target-agility level, the proposed constrained-filtering approach is capable of dealing with multiple agile target tracking scenarios.

**Acknowledgments**

First, I would like to express my gratitude to my supervisor Dr. Fakhri Karray for his continual support and help during my PhD studies. This work has only been made possible with his patience, guidance, and encouragement throughout the entire research work conducted for this dissertation. I would also like to express my gratitude to the members of my committee, Dr. Mohamed-Yahia Dabbagh, Dr. Oleg Michailovich, Dr. William Melek, and Dr. Simon Yang for all of their valuable comments and suggestions which helped improve the quality of this work. I should also thank Dr. Bahador Khaleghi for the helpful brainstorming discussions and valuable feedbacks. Last but not the least, I would like to thank my parents and my wonderful sisters for their motivation and support throughout my graduate studies.

## Dedication

I would like to dedicate my work to my mother and the memory of my father.

# Contents

# List of Abbreviations

SMCL: Sequential Monte Carlo

PF: Particle Filter

KF: Kalman Filter

EKF: Extended Kalman Filter

UKF: Unscented Kalman Filter

MSE: Mean Square Error

ARMSE: Average Root Mean Square Error

LRF: Laser Range Finder

MHT: Multi Hypothesis Tracking

MLE: Maximum Likelihood Estimator

PDF: Probability Density Function

EM: Expectation Maximization

SDC: Soft-Data-Constrained

MM: Multi-Model

PHD: Probability Hypothesis Density

IMM: Interactive Multiple Model

RS: Random Set

FIS: Fuzzy Inference System

CP: Consensus Propogation

WCP: Weighted Consensus Propogation

MMPF: Multiple-Model Particle Filter

CPF: Centralized Particle Filter

DPF: Distributed Particle Filter

TPM: Transition Probability Matrix

RCL: Reported Certainty Level

RAL: Reported Agility Level

KLD: KullbackLeibler divergence

OSPA: Optimal Sub-Pattern Assignment

ECW: Expected Cluster Weight

SAD: Stochastic Agility Discount

kNN: k Nearest Neighbour

AMM: Autonomous Multi-Model

JDL: Joint Directors of Laboratories

HCI: Human Computer Interaction

DSET: Dempster-Shafer evidence theory

GM: Gaussian Mixture

DOD: Department of Defense

HLIF: High Level Information Fusion

SIS: Sequential Importance Sampling

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Problem Statement

In cases where standard hard sensors are incapable of providing required information they have been designed and built for, and the required data is only available through soft data reports, soft data become very valuable. This is the most significant advantage over the data from hard sensors. For example, humans judge whether or not a particular type of relationship exists between some entities; whereas, hard sensors cannot provide information about the existence of a relationship, since they are designed primarily to measure attributes and features of entities [1]. Therefore proposing a framework to gather and model soft data and an appropriate way of fusing it to the estimation process can tackle the gaps arising from the limitation of the conventional (hard) sensors.

As shown in Figure 1.1, target tracker can track the conventional target which is behaving based on the pre-defined Transition Probability Matrix (TPM); however, for an unconventional target with an unknown behaviour characteristics, the tracker is unable to perform tracking. Incorporating high level information into the estimation process as shown in this figure enables the system to track agile targets with unknown behaviour characteristics.

Soft/hard data fusion is mostly aimed at large networks of information processing; therefore, a distributed scheme of proposed framework is required. In general, achieving network reliability is problematic, because the majority of fusion operators assume that the models producing beliefs are equally reliable and play a symmetrical role in the fusion process. In the distributed state-estimation process, it is important to share information

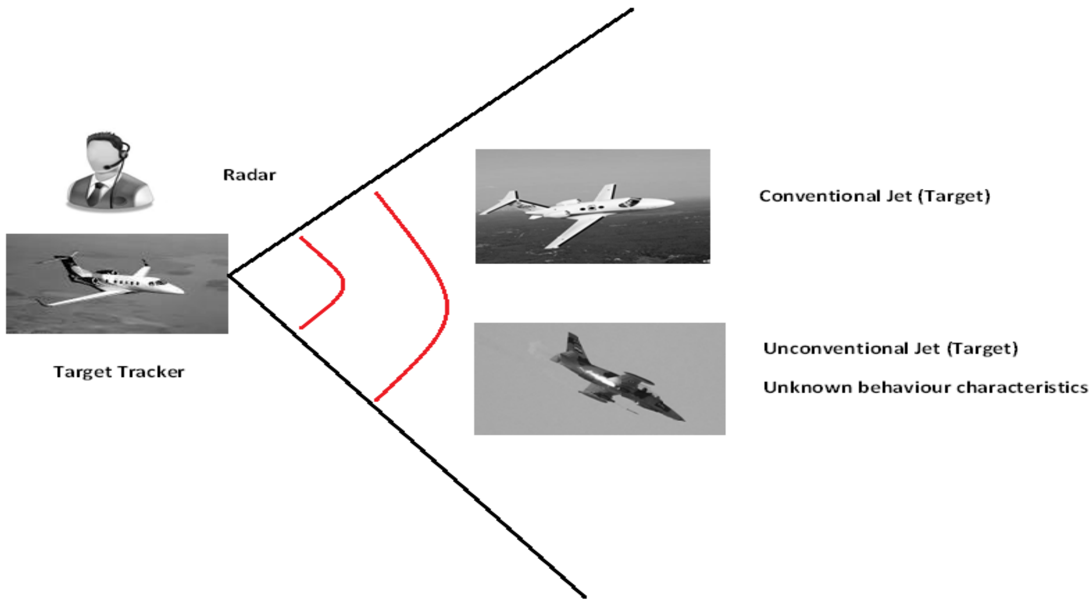


Figure 1.1: High level information fusion to enable agile target tracking

with a reliability factor, to reduce negative effects on neighbouring nodes, as these effects gradually affect the overall estimation throughout the network. In order to adaptively recognize and evaluate which source's information is valid, its history and its neighbouring nodes' estimation is required to evaluate its reliability factor.

Multi-target tracking is an ongoing research in the field; however, the problem in which maneuvering multiple-targets have unknown behaviour characteristics has not been yet tackled in the literature.

## 1.2 Motivations and Contributions

Human computation is a new and evolving research area that uses human intelligence to solve computational problems that are beyond the scope of existing approaches. In a modular and extensible distributed framework, many tasks can be accomplished by either human or machine performers [2], and some information can be provided only by humans.

Soft data can be obtained from direct human reports as well as from open source information on the Internet (e.g., MySpace, YouTube, Facebook, eBay, Craigslist, Wikipedia, Blogger, Photobucket and Flikr). This information can significantly augment data obtained from traditional sensors such as unattended ground sensors, radar, airborne vehicles and others. Burke et al. [3] have described the concept of participatory sensing, in which a community of observers might be tasked to provide information for applications such as urban planning and pubic health. Other examples using human cognition to provide information for different applications are detailed in [4].

Traditionally, fusion process designs have exploited the error characteristics of the (hard) input data; for example, knowledge of calibrated detection probability has been a crucial parameter affecting general operations. However, there is no such data for uncalibrated human observers or social media sources, so the quality of soft data should be assessed using a different approach and must be accounted for in design of various fusion operations.

In this thesis, a novel technique has been proposed to incorporate the high level information into the filtering process to improve the estimation results and recover from the failure situations, as well as quality assessment of soft data before integration. In some cases when agility exists, due to the lack of necessary information, the estimation is virtually unfeasible using conventional methods; however, the proposed technique provides this capability. Furthermore, the proposed method is implemented in a distributed sensor networks, and an adaptive source reliability approach is proposed to handle the source reliability throughout the estimation process. In next subsections, different aspects of the proposed method are highlighted.

### 1.2.1 Distributed Soft-Data-Constrained Filter

Soft/hard data fusion is mostly aimed at large networks of information processing; therefore, a new approach, Soft-Data-Constrained Multi-Model Particle Filtering (SDCMMPF), is proposed to enable distributed estimation of soft, as well as hard data. The proposed distributed non-linear estimation method which is based on SDCMMPF is applicable to a number of distributed state estimation problems. This method needs only local data exchange among neighbouring sensor nodes and thus provides enhanced reliability, scalability, and ease of deployment. To make the Multi-Model Particle Filtering (MMPF) approach work in a distributed manner, a Gaussian approximation of the particle cloud obtained at each sensor node and a Consensus Propagation (CP) based distributed data aggregation scheme are used to dynamically re-weight the particles' weights. The proposed method can recover from failure situations and is robust to noise, since it keeps the same population of particles and uses the aggregated global estimate to infer the constraints. The constraints are enforced by adjusting particles' weights and assigning a higher mass to those closer to the global estimate represented by the nodes in the entire sensor network after each communication step. Each sensor node experiences gradual change; i.e., if a noise occurs in the system, the node, its neighbours, and consequently the overall network are less affected than with other approaches, and thus recovers faster.

In order to evaluate the proposed method, a case in which the dynamics of the maneuvering target might deviate from the probabilistic characterization presented by the transition probability matrix is studied. This problem is referred to as agile target tracking, and it considers agility levels to be directly associated with the likelihood of unpredictable target maneuvers. The uncertainty regarding the target mode and its transitions are typically characterized in a Markovian manner, using the so-called transition probability matrix [5]. If a target is agile, its next mode, which is assumed to be one of the available modes, is unpredictable. Agile target tracking is an important and challenging problem, which, to the best of our knowledge, has rarely been addressed in the literature. This lack of interest is partly due to the difficulty of obtaining data regarding the agility level of targets when using the conventional sensory mechanisms. On the other hand, a relatively recent trend in data fusion community aims at exploiting data provided by humans [1, 6, 7]. Our observation is that human agents have advanced cognitive abilities, which allow them to provide valuable information regarding intricate target behaviors, including the agility. Accordingly, the proposed approach deploys human-generated data about targets' agility levels to improve tracking performance.

### 1.2.2 Source Reliability

This thesis also proposes a source-reliability based distributed adaptation scheme whose objective is to enable the algorithm to deal with distributed tracking scenarios in a robust manner by incorporating source reliability into the distributed fusion process.

The key observation is that in the presence of a permanently faulty hard sensor data or a misleading soft data, the sensor's local particle cloud would be significantly biased with respect to its aggregated global estimate. Accordingly, discounting the contribution of such a node to the global aggregation process could enhance the robustness of the distributed estimation process. The underlying distributed data aggregation scheme deployed to estimate the global Gaussian at each sensor node is a weighted variant of the consensus propagation [8]. This procedure is repeated iteratively to allow information sharing among neighbouring nodes, and consequently the entire network, with the objective of diverting particle clouds of all sensor nodes towards a global aggregate.

### 1.2.3 Multiple Agile Target Tracking

In the literature, numerous methods have been proposed to tackle the problem of tracking multiple targets [9]. The Probability Hypothesis Density (PHD) filter [10] is one of the most popular multi target tracking approaches. This thesis considers a problem in which the dynamics of maneuvering multiple-targets might be agile. The efficiency of the proposed method is verified through simulations for multi agile target tracking scenario that can process both soft and hard data in sensor networks.

### 1.2.4 Contributions

Motivated by the limitations of the existing methods discussed above, this thesis has the following objectives:

1. to model soft data using Fuzzy logic approach in order to capture its vagueness and to present an appropriate syntax and semantics that will allow modeling of soft (human-generated) data for target tracking applications,

2. to introduce a constrained variant of multi-model particle filter that can incorporate external information in order to refine the estimation,

3. to use the proposed method in a distributed framework so as to share the information among neighbouring nodes and therefore the entire network, since soft/hard data fusion is mostly aimed at large networks of information processing.

4. to model an adaptive reliability schema in the distributed framework,

5. to deploy PHD filter in order to extend the proposed target tracking framework to handle multi agile target tracking scenarios.

## 1.3 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 presents the background and a literature review of the existing approaches. Chapter 3 introduces the proposed method, the soft data modeling, and how to incorporate the soft data as dynamic constraints into the filtering process. The proposed method is further extended in Chapter 4 to enable distributed data aggregation, thus enhancing the scalability and robustness of the proposed framework Lastly, a series of single-target tracking experiments evaluates the proposed framework. The proposed adaptive source reliability scheme is detailed in Chapter 5. Chapter 6 further extends the target tracking problem into multi-target case by proposing a novel approach using PHD filter to account for multi agile target tracking scenarios. Chapters 3, 4, 5, and 6 begin with a brief review of the literature relevant to the proposed approach and end with a series of experiments that assess these approaches. Chapter 7 provides a discussion of the results and concludes the thesis, along with some directions for the future research.

# Chapter 2

# Background and Literature Review

## 2.1 Introduction

Multisensor data fusion is a technique used to enable information from several sources to be combined to form a unified picture. Data fusion systems are now widely used in various areas such as sensor networks, robotics, video and image processing, and intelligent system design, to name a few. Therefore, knowing different techniques and approaches, and their advantages and disadvantages, is valuable when considering a technique. Soft data, a recent integral component in information fusion is proposed to enhance the estimation process. It has different aspects to study, such as its collection requirements, challenges, and strategies, how it can be integrated into the fusion process, and its advantages over hard sensors. This section studies multisensor data fusion, and investigates the data fusion task, including its potential advantages, challenges, existing methodologies, and recent advances. Several reviews of the data fusion literature exists [11, 12, 13, 14, 15, 16].

Most data fusion research has been dedicated to problems associated with the first level of the Joint Directors of Laboratories (JDL) model [13]. As work on low-level fusion becomes well established and approaches mature field, research on high level fusion tasks is an active field of research and is gaining more attention. The JDL model shown in Figure 2.1 [17] has been used by data fusion community for fusion problems. Other fusion models exist, including the DDF model [18], the Omnibus model [19] and the perceptual reasoning model [20].

There are 4 levels of information fusion based on the JDL model as follows:

Level 0: Sub Object and Object Assessment, which involves processing data from sensors (e.g. signals, images, vector quantities, or scalar data).

Level 1: Object refinement, which combines data from multiple sensors or sources to obtain the most reliable estimate of the object's location, characterization, and identity.

Level 2: Situation Refinement, which uses the results from level 1 processing and develops a contextual interpretation of their meaning

Level 3: Thread Assessment, which considers the projection of the current situation into future to determine the potential impact or threats associated with the current situation.

Level 4: Process Refinement, which is also referred to as resource management observes the other levels of processing and seeks to make the fusion process better, i.e., more accurate, more timely, and more specific by redirecting the sensors and information sources, selecting which algorithm is appropriate for the current situation or available data and by changing the control parameters on the fusion algorithms.

Level 5: Processing, which involves Human Computer Interaction (HCI) and optimizes how the data fusion system interacts with one or more human users.

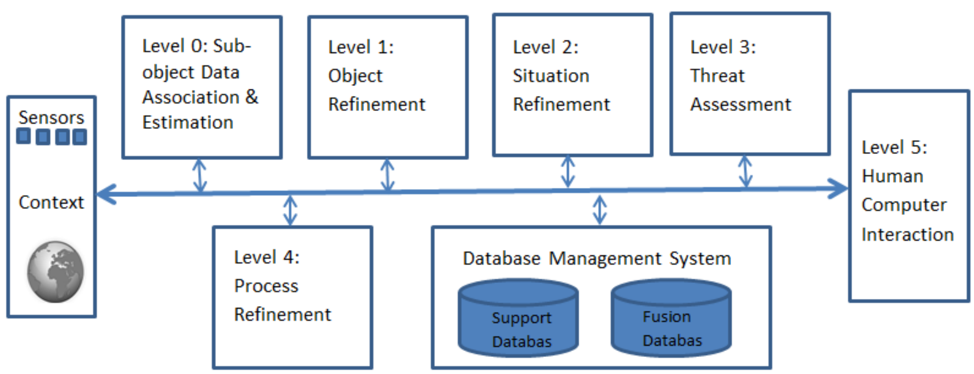


Figure 2.1: US Department of Defense (DoD) Joint Director of Laboratories Data Fusion Model [17]

High Level Information Fusion (HLIF) is Levels 2 and up. The primary objective of Level 1 fusion is tracking and identification of individual targets. By observing the kinematic (position and velocity) and classification (e.g. size) states, hard information sensors provide essential reports that can be associated to form tracks and estimate states of individual targets.

The rest of this chapter is organized as follows. An overview of data fusion problems, techniques and challanges are provided in section 2.2. Section 2.3 provides an overview of data fusion techniques. Section 2.4 provides an overview of soft data, such as its collection requirements, challenges and strategies; as well as, the fusion techniques for soft data fusion.

## 2.2 Data Fusion

In [21], the authors present a review and discussion of many data fusion definitions. Based on the identified strengths and weaknesses of previous works, they have proposed a principle-based definition of information fusion: "Information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making." Data fusion is a multi-disciplinary research area borrowing ideas from many different fields such as signal processing, information theory, statistical estimation and inference.

Generally, performing data fusion has several advantages [22, 12]; mainly involving enhancements in data authenticity such as improved detection, confidence, and reliability, as well as reduction in data ambiguity, and data availability. It can also provide specific benefits for some applications, such as wireless sensor networks, which are often composed of a large number of sensor nodes, and have scalability issues caused by potential collisions and transmissions of redundant data. Regarding energy restrictions, communication should be reduced to increase the lifetime of sensor nodes. When data fusion is performed during the routing process, i.e., when sensor data is fused and only the results are being forwarded, the number of messages are reduced, collisions are avoided, and energy is saved. In the proposed method, since all necessary information is shared only with neighbouring nodes, there is no need for nodes to broadcast their messages or send long-range messages, and yet share their information according to their reliability with the whole network.

The major novelty of this work is the ability to express all aspects of multi-source information processing, i.e., both data processing as well as estimation processing. Furthermore, it allows for consistent combination of the processing elements and providing a good performance. Such formalization enables traditional methods to increase their performance and helps to build an automatic development of fusion systems.

### 2.2.1 Challenging Problems in Multi-sensor Data Fusion

There are a number of challenges which occur in data fusion tasks [8]. The majority arise from the data to be fused, the imperfection and diversity of the sensors, as well as the nature of the application:

- Data imperfection: data provided by sensors has always uncertainty in the measurements; therefore, it is affected by some level of impreciseness. Data fusion algorithms should be able to express such imperfections effectively.

- Outliers and spurious data: the uncertainties in sensors are sometimes due to the ambiguities and inconsistencies of the environment, and from the inability to distinguish between them [23]. Data fusion algorithms should be able to exploit the redundant data to reduce such effects.

- Data alignment/registration: sensor data must be transformed from each sensor's local frame into a common frame before fusion occurs. Such an alignment problem is often referred to as sensor registration and deals with the calibration error induced by individual sensor nodes.

- Data association: multi-target tracking problems introduce much greater complexity to the fusion system than single-target tracking does [24]. One new difficulty is data association, which may come in two forms: measurement-to-track, which refers to the problem of identifying from which target, each measurement originates; and track-to-track association, which deals with distinguishing and combining tracks that are estimating the state of the same real-world target [13].

- Processing framework: data fusion processing can be performed in a centralized or decentralized manner. The latter is usually preferable in wireless sensor networks, as it allows each sensor node to process locally, collected data. This is much more efficient compared to the communication burden required by a centralized approach in which all measurements have to be sent to a central processing node for fusion.

- Data dimensionality: the measurement data may be preprocessed, either locally at each of the sensor nodes, or globally at the fusion center where it is compressed into lower dimensional data, assuming a certain level of compression loss is allowed. This preprocessing enables saving on the communication bandwidth and power required for transmitting data, in the case of local preprocessing [25], or limiting the computational load of the central fusion node, in the case of global preprocessing [26].

## 2.3 Data Fusion Techniques

Uncertainty is an important factor in the sensing and data fusion process. An explicit measure of the uncertainty must be provided to enable sensory information to be fused in an efficient and predictable manner. There are many methods of representing uncertainty, such as probabilistic models; however, this model often cannot capture all the information that is required to define and describe the operation of a sensing and data fusion system. Thus, a number of alternative modeling techniques have been proposed to deal with the perceived limitations of probabilistic methods [27], that are listed below:

- Complexity: the need to specify a large number of probabilities to be able to apply probabilistic reasoning methods correctly.
- Inconsistency: the difficulties involved in specifying a consistent set of beliefs in terms of probability and using these to obtain consistent deductions about states of interest.
- Precision of Models: the need to be precise in the specification of probabilities for quantities about which little is known.
- Uncertainty about uncertainty: the difficulty in assigning probability in the face of uncertainty, or ignorance about the source of information.

The alternative techniques used in literature for data fusion are [28], Fuzzy set theory [29], possibility theory [30], rough set theory [31], and Dempster-Shafer evidence theory (DSET) [32]. Most of these approaches are capable of representing specific aspect of imperfect data.

Even with the aforementioned limitations, probabilistic modeling techniques play an essential role in developing data fusion methods. Almost all conventional data fusion algorithms have probabilistic models as a central component in their development. Therefore, we have chosen interactive multi-model particle filter, which is a variant of Bayes theorem, as the underlying method for the estimation process. Fuzzy logic is used for soft data processing and to model the human-generated data, since it is a method capable of modeling the uncertainty about uncertainty of the soft data. In general, probabilistic measures are appropriate when dealing with ill-defined (random) variables hitting well-defined sets, whereas Fuzzy measures enable calculation of the membership of well-known variables in ill-defined (vague) sets [33].

This section provides an overview of some methods used in this thesis. Probabilistic distribution expresses data uncertainty and is used as an underlying data fusion technique;

Fuzzy set theory can represent vagueness of data; therefore, it is used to model the vague human-generated soft data. Finally, there is a description of Random set theoretic fusion approach which is used for multi-target tracking scenarios.

### 2.3.1 Fuzzy Logic-Based Fusion

Fuzzy logic is a popular method for representing uncertainty, particularly in applications such as supervisory control and high-level data fusion tasks. It is often claimed that Fuzzy logic provides an ideal tool for inexact reasoning, particularly in rule-based systems.

In contrast to the probability and evidence theories, which are well suited to modeling the uncertainty of membership of a target in a well-defined class of objects, Fuzzy sets theory is well suited to modeling the Fuzzy membership of a target in an ill-defined class. Yet, similar to probability theory that requires prior knowledge of probability distributions, Fuzzy sets theory requires prior membership functions for different Fuzzy sets [34]. A brief description of the main definitions and operations of Fuzzy logic is as follows:

Fuzzy logic introduces the novel notion of partial set membership, which enables imprecise (rather than crisp) reasoning [29].

Considering a Universal set $X$ consisting elements $x : X = \{x\}$, if $A$ is a subset of $X$ $A \subseteq X$, $\mu_A(x)$ is called membership function; which defines the degree of membership of element $x \in X$ in set $A$, this membership degree ranges between 0 and 1. Composition rules for Fuzzy sets follow the composition processes for normal crisp sets as:

AND is implemented as minimum

$\mu_{A \bigcap B}(x) = min[\mu_A(x), \mu_B(x)]$,

OR is implemented as maximum

$\mu_{A \bigcup B}(x) = max[\mu_A(x), \mu_B(x)]$,

NOT is implemented as compliment

$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

The normal properties associated with binary logic such as commutativity, associativity, incompetence, distributivity, De Morgan's law and absorption hold except the law of excluded middle:

$A \bigcup \bar{A} \neq X$

$A \bigcap \bar{A} \neq \emptyset$

### 2.3.2 Probabilistic Data Fusion

Estimation is the single most important problem in sensor data fusion. An estimator is a decision rule which takes as an argument a sequence of observations and then computes a value for the parameter or state of interest. There are a number of filters that are different variants of Bayes filter and are used for this purpose such as Kalman filter, which is a recursive linear estimator, successively calculating an estimate for a continuous valued state, that evolves over time, on the basis of periodic observations. Some of the non-linear data fusion methods are, Unscented Kalman Filter (UKF) [35], Extended Kalman Filter (EKF) [36], and Particle Filter(PF).

Extended KF and Unscented KF are based on the first-order and second-order approximations as a Taylor series expansion about the current estimate, respectively. However, both of these methods can only handle non-linearities to a limited extent.

The Monte Carlo simulation-based techniques such as Sequential Monte Carlo (SMC) [37] and Markov Chain Monte Carlo (MCMC) [38] are among the most powerful and popular methods of approximating probabilities. They are also very flexible as they do not make any assumptions regarding the probability densities to be approximated. Particle filter is a recursive implementation of the SMC algorithm [39], providing an alternative for Kalman filtering when dealing with non-Gaussian noise and non-linearity in the system. The proposed method in this work is based on particle filtering algorithm; therefore, a brief overview of this method and multi-model variant of it is presented in this section. Probabilistic methods rely on the probability distribution/density functions to express data uncertainty. At the core of these methods lies the Bayes estimator, which enables fusion of pieces of data. Let's say we have two random variables $x$ and $z$ with joint probability density $p(x, z)$. The Bayes theorem is as follows:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}, \tag{2.1}$$

where $p(x)$ is a prior probability density function and $p(z|x)$ is the conditional probability density function, which describes, for each fixed value of $x \in X$ the likelihood that observation $z \in Z$ will be made, i.e., probability of $z$ given $x$. $p(x|z)$ is the posterior distribution which describes the likelihoods associated with $x$ given the observation $z$. $p(z)$ is the marginal distribution which is used to normalize the posterior.

The value of Bayes theorem provides direct means of combining observed information with prior beliefs about the state of the world and it lies at the heart of many data fusion algorithm. It can be applied directly to integrate the observation of multiple sources of information, $Z^n \triangleq \{z_1 \in Z_1, ..., z_n \in Z_n\}$. The posterior distribution will be constructed using this information as follows:

$$P(x|Z^n) = \frac{p(Z^n|X)p(x)}{p(Z^n)}, \tag{2.2}$$

It requires that the joint distributions of all possible combinations of observations conditioned on the underlying state to be known completely. However, it is reasonable to assume that information from each source is independent of the information obtained from the other sources. Therefore, we have:

$$p(x|Z^n) = \frac{p(x)\Pi_i^n p(z_i|x)}{p(Z^n)}; \tag{2.3}$$

therefore, the posterior distribution on $x$ in this case is the product of prior likelihood from each source.

### 2.3.3 Particle Filter

Particle filter is a powerful sampling-based inference/learning algorithm that is an implementation of the Bayes filter which is a general probabilistic approach for estimating an unknown probability density function recursively over time. Particle filter has been appeared in several fields under such names as "condensation", "Sequential Monte Carlo" and "survival of the fittest" [40, 41]. It uses Sequential Monte Carlo methods [42] and allows treatment of any type of probability distribution and nonlinearity. The advantage of using particle filtering method is that there is no need of assumption of the underlying distribution. For many applications, it is important to account for nonlinearity and non-Gaussianity to model the underlying dynamics accurately. Therefore, in this work PF, which is a variant of Bayesian algorithm, is selected as an underlying algorithm of the proposed approach.

Extended Kalman Filter (EKF) which is also an implementation of the Bayes filter can deal with nonlinear dynamics as well, but still needs to assume the existence of Gaussian noise [43]; however, PF can deal with linear/non-linear dynamics with arbitrary noise. In PF, Monte Carlo methods are used to update the density representation over time. Such a representation is an approximation, but it is nonparametric; therefore, it can represent a much broader space of distributions than Gaussian.

### 2.3.4 Multi-Model Particle Filtering

Among the methodologies developed in the literature on maneuvering target tracking, the multi-model techniques are probably the most popular [44]. The multi-model tracking methods belong to the family of hybrid estimation techniques; in which, the target state includes both continuous and discrete components. The base target state is the component that varies continuously as in conventional tracking systems. The discrete component however, has a stair-case type trajectory, i.e., it may either jump or remain unchanged, which is commonly known as the target mode. The conventional solution to multi-model tracking is to follow the estimation after the decision approach, i.e., first deciding on the best mode of the target and then applying a single filtering process using the chosen mode as if it is the correct one [44].

In [45] the multi-model tracking algorithms have been categorized into three generations, in which each new generation is deemed to be fundamentally different in terms of operation, structure, and capabilities. The first generation, typically referred to as the Autonomous Multi-Model (AMM) filtering, has the distinctive characteristic that each one of the elemental filters operates independently [46]. The second generation, Interacting Multiple Models (IMM), improves upon the first by enabling the elemental filters to operate in a more efficient cooperative manner through effective interactions [47]. Finally, the third and the most recent generation adds the benefits of variable structure filtering, e.g., constant adaptation of mode transition probabilities to further enhance the performance [48]. Accordingly, the constrained Multi-Model Particle Filter (MMPF) method proposed in thesis can be considered as a variant of the variable structure MMPF algorithms.

Scenarios in which target experiences sudden motions and distractions have been tackled by Wang *et al.* [49], by incorporating the efficiency of the mean-shift algorithm with the multihypothesis characteristics of particle filtering technique. Another approach proposed by Hou *et al.* [50] in this trend, is a robust adaptive control to solve the consensus problem of multiagent systems, in which the agent's dynamics includes the uncertainties and external disturbances, which is more practical in real-world applications. In this approach due to the approximation capability of neural networks, the uncertain dynamics is compensated by the adaptive neural network scheme. The effects of the approximation error and external disturbances are counteracted by employing the robustness signal. A fast target maneuver detection and highly accurate tracking technique using neural Fuzzy network based on a Kalman filter is proposed in [54].

However, in cases in which there exists a transition matrix for a predefined dynamics of the system, MMPF can be used. MMPF has been proposed by several authors [51, 52, 53, 5] to perform nonlinear filtering with switching dynamic modes.

MMPF operates as a general discrete-time hybrid system, which is modeled by the following target dynamic and measurement models:

$$x_t = f_{t-1}(x_{t-1}, r_t) + \varepsilon_{t-1}(r_t) \tag{2.4}$$

$$z_t = h_t(x_t, r_t) + \delta_t, \tag{2.5}$$

where $f$ and $h$ are the state transition and measurement functions; and the covariance matrices of the process noise, $\varepsilon_{t-1}$, and measurement noise, $\delta_t$ are $Q_{t-1}(r_t)$ and $R_t$, respectively. The $r_t \in S = \{1, ..., s\}$ is the regime (mode) variable in effect during the sampling process, and the target state is represented as an augmented hybrid-state vector defined as $y_t = [x_t^T r_t]^T$. First, the next set of particle modes is predicted, $\{r_t^n\}_{n=1}^N$, based on the particles' previous modes, $\{r_{t-1}^n\}_{n=1}^N$, and the Transition Probability Matrix $\Pi$, where each element of TPM define the transition probability from mode i to mode j. If $r_{t-1}^n = i$, then $r_t^n$ should be set to $j$ with a probability equal to $\pi_{ij}$; therefore, if $r_{t-1}^n = i$ and we have a random sample $u_n \sim U(0,1)$, then $r_t^n$ is set to $m \in S$ such that:

$$\sum_{j=1}^{m-1} \pi_{ij} < u_n \leq \sum_{j=1}^{m} \pi_{ij} \tag{2.6}$$

The cumulative distribution function of discrete random variable $r_t$ given $r_{t-1} = i$ is given by $\sum_{j=1}^{m} \pi_{ij}$. The next step involves Sequential Importance Sampling (SIS), in which first the next state is predicted based on the previous state $x_{t-1}$ and observations up to time $t-1$, denoted as $z_{1:t-1}$.

$$\begin{aligned} p(x_t, r_t = j | z_{1:t-1}) = \\ \sum_i \pi_{ij} \int p(x_t | x_{t-1}, r_t = j) \times \\ p(x_{t-1}, r_{t-1} = i | z_{1:t-1}) dx_{t-1} \end{aligned} \tag{2.7}$$

In the updating part, this prediction is updated based on the current measurement at time $t$, $z_t$, in which the measurement is obtained.

Table 2.1: Algorithm 1: Generic MMPF

| |
|---|
| $[\{y_t^n, w_t^n\}_{n=1}^N] = \text{MMPF}\ [\{y_{t-1}^n, w_{t-1}^n\}_{n=1}^N, z_t]$ |
| Step 1: Regime transition (RT): |
| $[\{r_t^n\}_{n=1}^N] = \text{RT}\ [\{r_{t-1}^n\}_{n=1}^N, \Pi]$ |
| Step 2: Regime conditioned SIS: |
| $[\{x_t^n, w_t^n\}_{n=1}^N] = \text{RC-SIS}\ [\{x_{t-1}^n, r_k^n, w_{t-1}^n\}_{n=1}^N, z_t]$ |
| Step 3: $\hat{N}_{eff} = \frac{1}{\sum_{n=1}^N (w_t^n)^2}$ |
| Step 4: If $\hat{N}_{eff} < N_{thr}$ |
| Resampling |
| End If |

$$
\begin{aligned}
&p(x_t, r_t = j|z_{1:t}) = \\
&\frac{p(z_t|x_t, r_t = j)p(x_t, r_t = j|z_{1:t-1})}{\sum_j \int p(z_t|x_t, r_t = j)p(x_t, r_t = j|z_{1:t-1})dx_t}
\end{aligned}
\tag{2.8}
$$

After updating step, in resmapling step the weights of the particles are normalized. Resampling is used to avoid the problem of degeneracy of the PF algorithm.

### 2.3.5 Random Set Theoretic Fusion

The capability of random set theory has been studied in literature [55, 56, 57]. The most notable work on promoting random set theory as a unified fusion framework has been done by Mahler [24, 57, 58, 59]. In particular, in his book [59] he attempts to present a detailed exposition of random set theory and its application to data fusion problems. Random set theory is usually deemed as an ideal framework for extending the popular Bayes filter from single-target (modeled by a random variable) into multi-target (modeled by a random set).

Using random set theory, states and measurements, are modeled as random sets of finite size instead of conventional vectors; therefore, priors and likelihood functions constructed are capable of modeling a wide range of different phenomena. For instance, phenomena related to the system dynamics such as target disappearance/appearance, extended/unresolved targets, and target spawning, as well as measurement-related phenomena such as missed detection and false alarms can be explicitly represented.

Different approximation techniques are devised to compute the Bayes update equation, as one cannot expect to solve for this multi-target tracking analytically as was not the case for single-target Bayes filter. The moment matching techniques have been very successful in approximating the single-target Bayes filter. For instance, Kalman filter relies on propagating the first two moments (i.e. mean and covariance) while alpha-beta filters match only the first moment. In case of multi-target tracking, the first moment is the Probability Hypothesis Density (PHD), which is used to develop a filter with the same title, i.e., PHD filter [60]. There is also a higher order extension of this filter called Cardinalized Probability Hypothesis Density (CPHD) filter [61], [62], which propagates the PHD as well as the full probability distribution of the random variable representing the number of targets. Both PHD and CPHD filters involve integrals that prevent direct implementation of a closed form solution. As a result two approximation methods, namely, Gaussian Mixture (GM) and Sequential Monte Carlo (SMC), have been used in the literature to further ease the implementation stage for these filters [63, 64]. One important advantage of the (C)PHD family of filters is to avoid the data association problem, but this also means that maintaining track continuity can become a challenging task.

### 2.3.6 Distributed Data Fusion

In distributed Bayes theorem, the sensor models in form of likelihood functions, are maintained locally at each sensor node. When an observation is made, these likelihoods are instantiated to provide a likelihood function describing a probability distribution over the true state of the world. Importantly, it is this likelihood that is transmitted to the fusion centre not the raw observations. Then, the central fusion center computes the normalised product of communicated likelihoods and prior to yield a posterior distribution.

A second approach to distributed data fusion using Bayes theorem is that each sensor node computes a likelihood and combines it locally with the prior from the previous timestep, and sends its local posterior distribution on the state to the central fusion centre. The fusion centre then recovers the new observation information by dividing each posterior by the global prior and then taking a normalised product to produce a new global posterior.

This posterior is then communicated back to the sensors and the cycle repeats in recursive form.

The literature offers some designs that use distributed Bayes filters in sensor networks, either in a centralized or a distributed manner. In the first category, all the nodes send their information to a base station (center-based fusion) [65, 66] that performs all calculations and sends the final estimation back to all nodes in the network. This approach requires extensive communication and is very costly. Huge numbers of communications require a large amount of energy and offer a possible failure point at the central node. To overcome these issues, distributed strategies are an alternative providing a more general and robust solution, with fewer communications and the possibility of parallel processing [67, 68, 69, 70]. In [71], the Kalman filtering iterations are parallelized over a set of sensors; however, it still requires a fusion center to combine the estimates. Distributed particle filtering approaches are more effective for large-scale, nonlinear and non-Gaussian distributed estimation problems [72], [73].

Based on the type of data communicated between nodes, Distributed Particle Filtering (DPF) is classified into two types [74, 75]: statistics dissemination-based, in which processed data is exchanged between nodes [76, 77], and measurement dissemination-based, in which raw measurements are exchanged [78, 79]. Different statistic dissemination-based methods exist, varying in their scheduling and communication topology. The proposed method lies in the category of statistics dissemination-based methods, and the communication among nodes is consensus-based, which means that all nodes in the network process the data simultaneously [80, 81].

The distributed state estimation methods mainly rely on distributed data aggregation schemes as their underlying enabling technology. The most common distributed data aggregation schemes are gossip-based consensus filters [84], message passing (belief propagation) algorithms [85], and data diffusion processes [86]. A distributed consensus filter is proposed in [81] where each sensor can communicate with the neighboring sensors, and only a small fraction of sensors need to measure the target information, with which the whole network can be controlled. Consensus algorithms are used for distributed computations [87, 80]. According to the context, consensus means a global agreement on some quantity that depends on the data of all nodes [74].

Each of these algorithmic categories has its own benefits and disadvantages. Additionally, a few hybrid approaches aim at providing a framework for developing algorithms based on existing methods and try to minimize their inherent restrictions as much as possible.

## 2.4 From Soft Data to Intelligence

### 2.4.1 Soft Data Collection Requirements, Challenges, and Strategies

It is very difficult to develop techniques for combining human-supplied data with traditional sensory data; some issues including how to quantify the uncertainty of human data, how to model humans as sensors, how to task humans as sources of information, and even how to elicit information. One of the challenges of soft/hard information fusion is knowledge elicitation, which refers to the general problem of obtaining information from human observers or experts. This process may include soliciting information about their cognitive processes, beliefs, methods, observations and the uncertainty associated with their beliefs and observations. Extensive research has been performed in this area [1]. Based on O'Hagan *et al.* [88], this problem is divided into two related areas as follows:

- Elicitation of general information about a cognitive decision process (e.g. methods to perform knowledge engineering to understand how an analyst processes data, makes inferences and decisions for a general problem, such as in situation analysis). This issue is related to the original concept in expert system development of creating a knowledge base made up of facts, rules, scripts, frames or analogical representations of the inference process to describe how an expert such as a maintenance technician, physician, or intelligence analyst processes data to make inferences. McNeese *et al.* [89, 90, 91, 92] have conducted knowledge elicitation in different applications.

- The second area of knowledge elicitation concerns how to elicit information from observers about an evolving situation, event, or activity, i.e, how we can obtain observations or reports with associated probabilities, confidence factors or other measures of uncertainty.

The other challenge of soft/hard information fusion involves with developing the data formats. One of the key challenges that Rimland *et al.* [2] have encountered is the development of data formats, protocols, and methodologies to establish an information architecture and framework for the effective capture, representation, transmission, and storage of the vastly heterogeneous data and accompanying meta-data; including the capabilities and characteristics of human observers, uncertainty of human observations, "soft" contextual data, and information pedigree.

### 2.4.2 Constrained Bayesian Filtering

Attempts to improve tracking by integrating the external knowledge as constraints can be traced back to the early 1990s [106]. The literature related to constrained Bayesian filtering contains a wide spectrum of techniques, including pseudo-measurement [107], clipping [108], projection [109], and optimization-based methodologies [110]. The formalized constraints themselves can be of various types and forms, such as linear, non-linear, soft, hard, equality, and inequality [111]. Constrained variants of the particle filtering method have also been proposed in the literature, and assume a variety of domain-specific constraints [112, 113]. Simon [100] noted that for linear systems with linear constraints all of the existing approaches result in the same optimal state estimate. On the other hand, for non-linear cases, the number of state estimation techniques can be overwhelming, as the constrained filtering problem can be viewed from many different perspectives. Research on the theory and implementation of constrained particle filters remains an active field of research.

Using external knowledge and model it as the constants to be applied to the filtering process has been an active research recently. Papi *et al.* [113] have mentioned that practical application of Particle Filter (PF) for the nonlinear target tracking application requires available external knowledge to be formalized in terms of constraints on target dynamics to increase the tracking performance. They have studied the case of perfectly known hard constraints and have shown that if constraints are known and correctly modeled, then the PF converges to the correct a posteriori Probability Density Function (PDF). They have studied the case of soft constraints and pointed out that the lack of information on when and how the target violates the constraints makes the filtering problem much more difficult; however, detecting the violation of constraints is possible if the knowledge is processed using an Interactive Multiple Models (IMM) scheme. Hall *et al.* [114] briefly review ongoing work on dynamic fusion of hard/soft data by pointing out its motivation, advantages and challenges. Another related trend focuses on the so-called human centred data fusion paradigm and emphasizes the human role in data fusion [66, 74].

In contrast to the conventional data provided by well-calibrated sensors, also referred to as hard data, human-generated data, known as soft data [110], [104], are typically unstructured, vague, and subjective. For example, in [115], Cano *et al.* have proposed integrating the expert knowledge as external information in order to reduce the entropy of the posterior. The posterior is updated based on the expert's knowledge, and it is assumed that the expert always gives a definite answer, without any error. Their approach is based on Sequential Monte Carlo (SMC), which requests expert information about the direct probabilistic relationships between variables, which cannot be reliably discerned with the help of the data.

### 2.4.3 Soft Data Fusion

Once the importance of soft data and how to collect information from the correct sources is known, the challenging part is the fusion process, i.e., how to combine the information provided by the soft data and hard data. In contrast to the conventional data provided by well-calibrated sensors, also referred to as hard data, human-generated data, known as soft data [1, 33], are typically unstructured, vague, and subjective. On the other hand, the main advantage of humans is their ability to perform complex cognitive tasks. They can provide high level data regarding the targets that could be very difficult, if not impossible, to obtain using hard conventional sensors. A tremendous amount of research has been done on data fusion using conventional sensors. In contrast, limited work has studied the fusion of data produced by human and non-human sensors. Hall et al. [1] provide a brief review of ongoing work on dynamic fusion of hard/soft data, identifying its motivation and advantages, challenges, and requirements. A recent preliminary research in this area is the work on generating a dataset for hard/soft data fusion intended to serve as a foundation and a verification/validation resource for future research [93]. Very recently, a Dempster-Shafer theoretic framework for soft/hard data fusion was proposed that relies on a novel conditional approach to updating, as well as a new model to convert propositional logic statements from text into forms usable by Dempster-Shafer theory [94].

Another trend of work along this area is focused on the so called *human centered data fusion* paradigm and puts emphasis on the human role in data fusion process [17, 95]. This new paradigm considers humans as active participants in the data fusion process and not merely as soft sensors but also as hybrid computers and *ad-hoc* teams (hive mind). In spite of these accomplishments, research on hard/soft data fusion, as well as human-centered fusion is still in its fledging stage and should provide rich opportunities for further theoretical advancement and practical demonstrations [33].

## 2.5 Summary

Different data fusion techniques along with their advantages and disadvantages were presented in this chapter. This is followed by a description of soft data, how it differs from the conventional sensors, and few of the challenges and requirements of soft data collection. The last part details how, this information should be integrated into the fusion process.

In the next chapter, the proposed approach along with the proposed procedures used to model soft data while incorporating it in the fusion process are discussed and the experiments for different scenarios are demonstrated to evaluate the efficiency of the approach in different situations.

# Chapter 3

# Soft-Data-Constrained Multi-Model Particle Filter

## 3.1 Introduction

The performance of Bayesian filtering based methods can be enhanced by using extra information incorporated as specific constraints into the filtering process. Following the same principle, in the proposed method the inherently vague human-generated data are modeled using a Fuzzy Inference System (FIS). The soft data are then transformed into a set of constraints, which enable the MMPF method to deal with tracking situations involving potentially highly agile targets.

The problem of tracking targets whose dynamics include multiple-switching regimes, also known as maneuvering target tracking, has been studied extensively as reflected in the review paper series by Li & Jilkov [96, 97, 98, 99, 44]. The maneuvering target tracking problem in the presence of non-linearity has also been studied, and several methods such as function approximation, sampling-based moment approximation, and stochastic model approximation have been proposed to tackle this issue [86]. The sampling-based moment approximation methods using the sequential Monte Carlo approach, also known as particle filtering, are widely deployed to deal with non-linear tracking problems. An extension of the particle filtering methodology according to the multi-model principles, i.e., multi-model particle filter, provides a powerful and flexible solution to non-linear maneuvering target tracking problems. The uncertainty regarding the target mode and its transitions are typically characterized in a Markovian manner, using the so-called transition probability matrix [5].

In this work, we consider the problem where dynamics of the maneuvering target might deviate from the probabilistic characterization represented by the transition probability matrix. This problem is referred to as agile target tracking, and considers agility levels to be directly associated with the likelihood of unpredictable target maneuvers. If a target is agile, its next mode, which is assumed to be one of the available models, is considered unpredictable. Agile target tracking is an important and challenging problem. Human agents have advanced cognitive abilities, which allow them to provide valuable information regarding intricate target behaviors, including agility. Accordingly, the proposed approach deploys human-generated data about target's agility levels to improve tracking performance.

A popular approach to enhance the performance of Bayesian filtering methods using extra information and subsequent incorporation of specific constraints into the filtering process is introduced in [100]. Following the same principle, we propose a soft-data-constrained MMPF method; where the inherently vague (subjective) soft data provided by human agents are modeled using a Fuzzy inference system. They are then transformed into a set of constraints, which are enforced to enable dealing with tracking situations involving potentially highly agile targets.

The rest of this chapter is organized as follows. Soft data modeling and soft data fusion are discussed in section 3.2. Section 3.3 presents the experimental results conducted to compare our proposed method with the regular MMPF.

## 3.2 Soft Data Modeling

In developing the architecture of the proposed framework, some factors are important to be considered at the beginning of the process, the two most important ones are as follows:

The first one is to define at what point during the process, the hard and soft data streams should be fused, i.e., what is the architectural framework. It is often argued that the data should be joined together at the closest point to its origin, i.e., where the data are in a "raw" nature. This approach is often advanced on the basis of an information-theoretic argument, which claims that any operation on unfused data loses valuable information. However, there are at least two factors that prevent the choice of this option. One is that, while there is access to raw data for hard sensors, such as primitive perceptual data of say Radar or an imaging sensor (blobs of some type in the pixelized data), there is no equivalent raw data access on the soft data side. That is, there is no access to the primitive perceptual and early-cognitive operations in the mind of the human observer, access on the soft side is at the reported-entity level.

Processing and manipulating raw data from hard sensors is quite well-known; however, accessing and processing such raw data during human observation has a very high technical risk and is the second mitigating factor. In general, there is enough difficulty even in processing the entity-level data from human observers; therefore, fusing the data at the entity level is usually preferred. Performing fusion at this level is inherent to soft reporting, but this choice imputes a need to process the hard data stream to the entity level, i.e., the hard data should be operated on to the point of generating entity-level estimates. This can be done either from a single hard sensor or be the result of multiple hard sensor fusion operations to the state estimation level. The entity-level is considered quite natural for the domain of intelligence analysis [101].

On the soft data side, a major difficulty and design choice is defining a robust natural-language-processing (NLP) capability, also called text-extraction methods. The problem is that the realization of a natural language understanding capability has been a goal of ongoing research for many years. On the other hand, by performing automated fusion, it is expected to achieve the best possible way to extract rich semantic meaning from the reported linguistic data.

To model the soft data report, which is supplied by a human observer, the reports are assumed to comply to a specific syntax and semantics, which are predefined by an ontology. Please note that an appropriate Natural Language Processing (NLP) method can be used to format raw soft data according to this syntax. The syntax for the soft data report is shown in Figure 3.1. As shown in this figure, each report is a natural language expression that reports on the agility level of the target along with the level of certainty presumed by the reporter. In other words, each report is an expression comprised of a target-identification term, target ID, a qualifier term to express the level of certainty (RCL) presented by the report, and a term to represent the reported agility level (RAL) of the target.

| | |
|---|---|
| | **Report=<Target ID> is <qualifier> <level of agility> agile** |
| **Expression** | **Expression=<Target ID> is <RCL> <RAL> agile**<br><br>**<Target ID> ∈ {robot}**<br>**<RCL> ∈ {slightly, perhaps, certainly}**<br>**<RAL> ∈ {extremely, highly, marginally/not}** |

Figure 3.1: The syntax considered for the soft data reports

Table 3.1: Fuzzy rules for the case of: $RCL$="certainly" &$RAL$="extremely"

| |
|---|
| if ($ECW$ is high) & ($SAD$ is high) then ($C$ is med) |
| if ($ECW$ is high) & ($SAD$ is med) then ($C$ is low) |
| if ($ECW$ is high) & ($SAD$ is low) then ($C$ is vlow) |
| if ($ECW$ is low) & ($SAD$ is high) then ($C$ is med) |
| if ($ECW$ is low) & ($SAD$ is med) then ($C$ is high) |
| if ($ECW$ is low) & ($SAD$ is low) then ($C$ is vhigh) |

### 3.2.1 Soft Data Modeling Using Fuzzy Logic

Fuzzy inference systems are used to capture the uncertainty arising from the vagueness of soft data. One could argue in favour of probabilistic approaches as an alternative to Fuzzy logic for soft data processing. However, probabilistic measures are appropriate when dealing with ill-defined (random) variables hitting well-defined sets, whereas Fuzzy measures enable calculating the membership of well-known variables in ill-defined (vague) sets [33, 102]. The Fuzzy method can be two dimensional (2-D), with one dimension for the universe of discourse of the variable and the other for its membership degree, or 3-D with an extra dimension for spatial information [103]. In this work, a 2-D Fuzzy system is used to model the human report [104, 105]. The modeling of human report using a Fuzzy system further described in the next subsection.

The semantics used to interpret the given soft data is given as follows, with three different categories for the Reported Agility Level (RAL) and Reported Certainty Level (RCL). For the RAL, the report can be judged extremely, highly, or marginally agile and for the RCL, it can be considered as certainly, almost, or perhaps certain. As a result, we have modeled nine different FISs that have different rules and different membership functions for the output variable. Based on the agility level reported, RAL, and the certainty level of the report, RCL, one of the Fuzzy models is chosen. A simple decision tree is deployed to accomplish the Fuzzy model selection. Based on the inputs RCL={"slightly", "perhaps", "certainly"} and RAL={"extremely", "highly", "marginally"}, the appropriate model is selected. After choosing the FIS, the agility level of the target along with the respective value of the transition probability matrix are the inputs to the Fuzzy system that outputs the constraints.

Table 3.1 shows the rules defined when the reported RAL is "extremely". The rules of the FIS change based on different values reported for RAL; therefore, we have nine different sets of Fuzzy rules. This table demonstrate the rules defined for different values of RAL. In this table, the terms "vlow", "med" and "vhigh" represent very low, medium

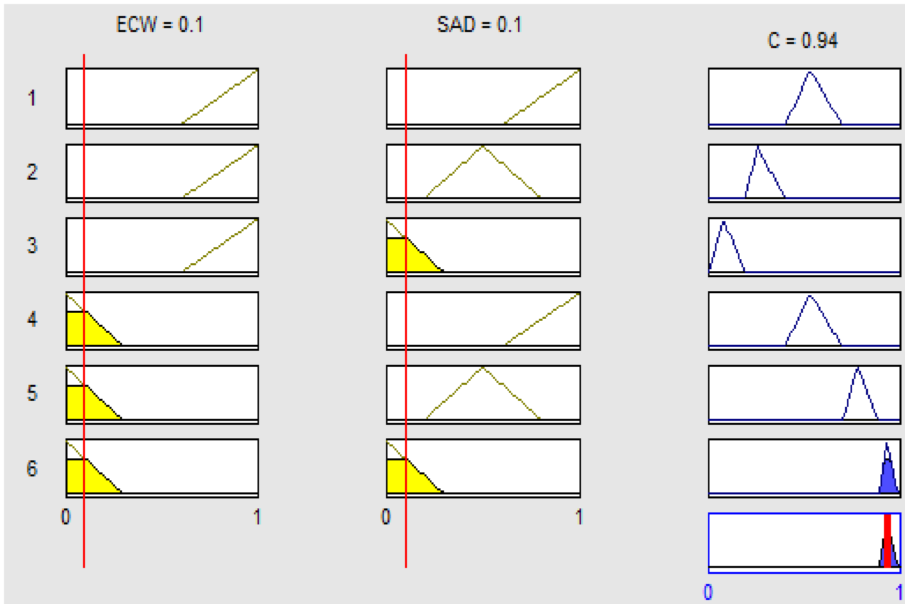

Figure 3.2: Exemplar Fuzzy inference systems for soft data report as: "target is certainly extremely agile"

and very high respectively. The value reported for RCL affects the membership function of the output values $C_{mm'}$. As the certainty level increases, the membership functions get narrower.

Figure 3.2 show the Fuzzy inference systems for two cases in which the soft data is reported as "target is certainly extremely agile" (3.2). The membership functions of the output are defined based on the value reported for the RCL. As shown in Figure 3.2, the membership functions are narrower and have less overlap to reflect a higher certainty level of the report; whereas, in the case of a less certain report, they are wider and have more overlap.

The rules of the FIS are adapted based on the value reported for the RAL. Table 3.1 (which corresponds to Figure 3.3) demonstrates a set of rules defined for two different RALs. The rules in this table are defined when the RAL is "extremely". Also, Figure 3.2 corresponds to the Fuzzy rules presented in Table 3.1, with the inputs (ECW is low) and (SAD is low). The discussion presented in next section elaborates on how the Fuzzy rules for each Fuzzy Inference System (FIS) are adapted to achieve the desired constrained filtering behaviour. Figures 3.3 to 3.6 show the effect of soft data report's certainty level on the output constraint (shown as c on z-axes in the figure). These figures demonstrate

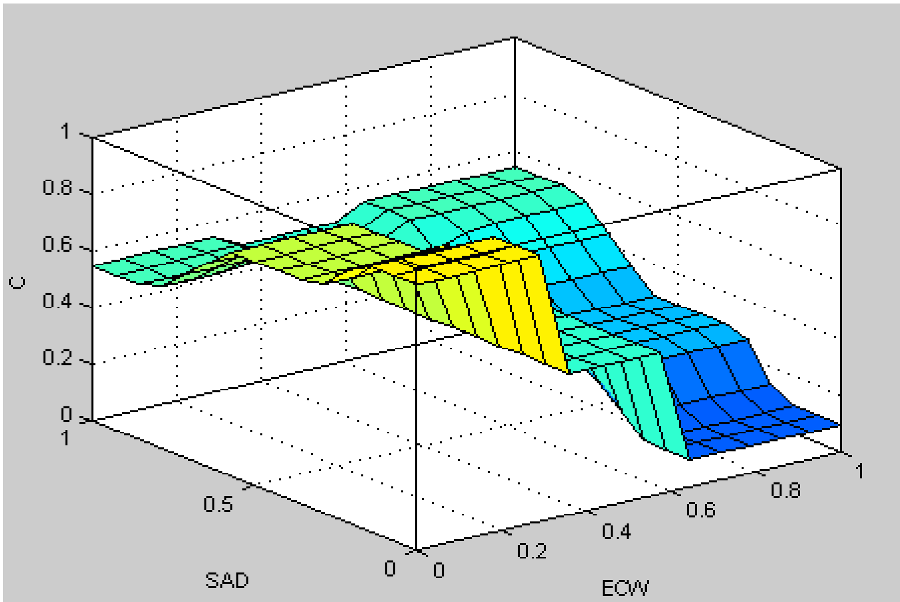


Figure 3.3: The effect of soft data report's certainty level on Fuzzy inference system, with soft data report as: "target is certainly extremely agile"

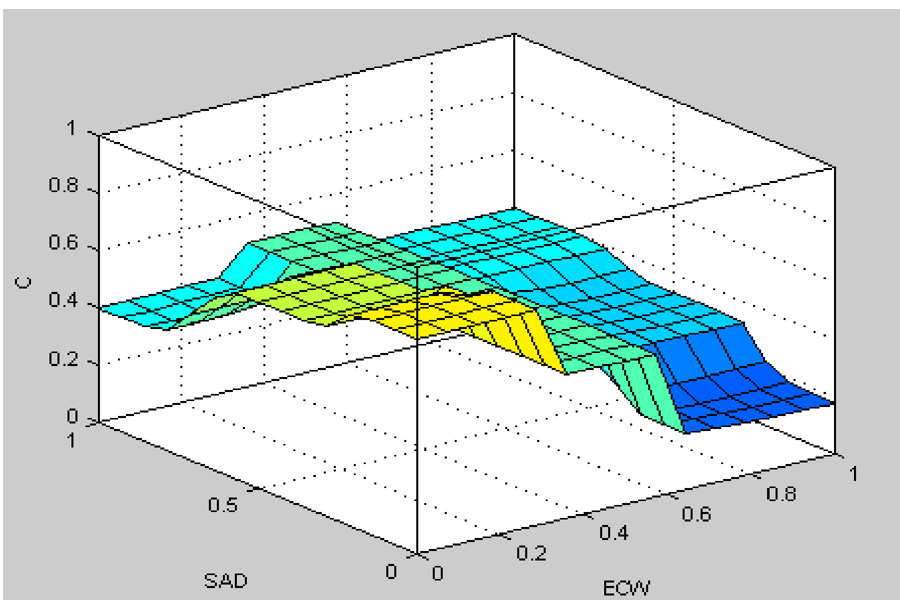


Figure 3.4: The effect of soft data report's certainty level on Fuzzy inference system, with soft data report as: "target is perhaps extremely agile"

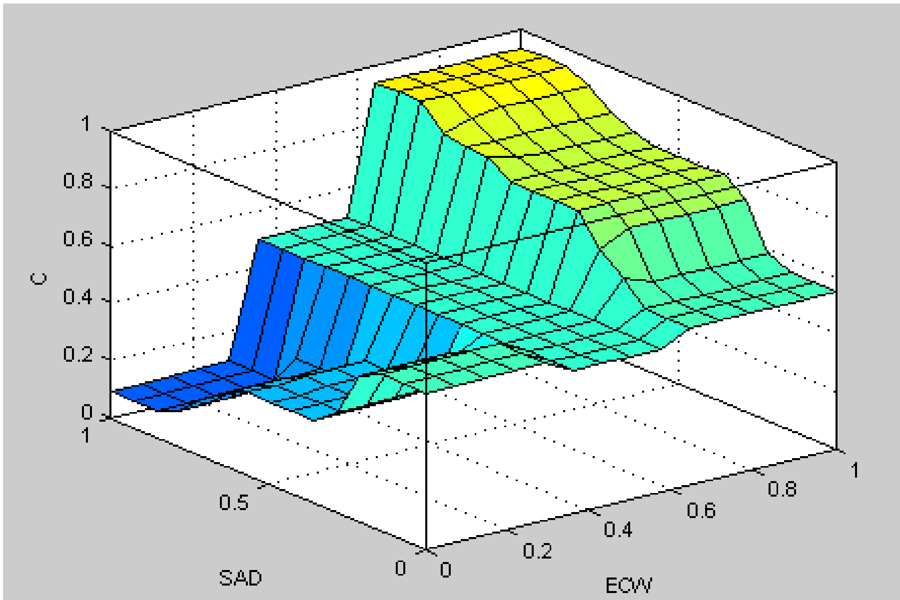


Figure 3.5: The effect of soft data report's certainty level on Fuzzy inference system, with soft data report as: "target is certainly marginally agile"

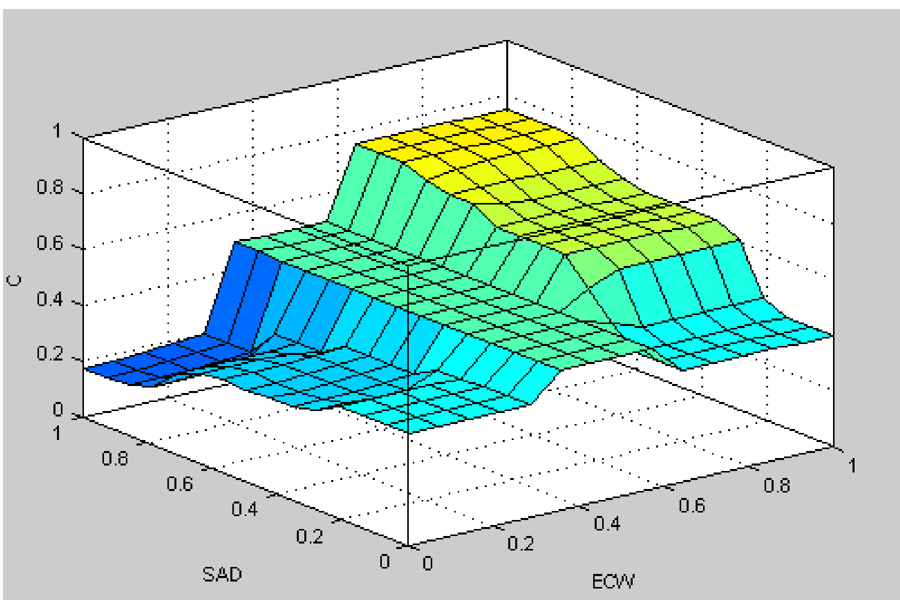


Figure 3.6: The effect of soft data report's certainty level on Fuzzy inference system, with soft data report as: "target is perhaps marginally agile"

different outputs (z-axis) resulted based on different input values SAD (y-axis) and ECW (x-axis).

### 3.2.2 Soft Data Fusion

#### Incorporating Soft Data As Dynamic Constraints

Algorithm 2 in Table 3.2 depicts the pseudocode of the proposed Soft-Data-Constrained Multi-Model Particle Filter (SDC MMPF).

The algorithm starts with *step 1*, which consists of the uniform initialization of the particle clouds; that is, each of the possible target modes is represented by the same number of particles, $\frac{N}{M}$. Each particle having the same weight computed as $\frac{1}{N}$, in which, $N$ and $M$ are the total number of the particles and the number of target modes, respectively. Our contributions reside in *step 2*, the mode prediction step. The generic MMPF prediction aims at simulating the target mode transition probabilities dictated by matrix $\Pi$. Due to the stochastic nature of this process, the predictions of MMPF are always slightly different from those that result from $\Pi$.

For agile targets, the higher the agility level, the less the likelihood of the next target mode being the same as the mode predicted by $\Pi$. Accordingly, our main objective is to reinforce the aforementioned stochastic difference, if the target is reported to be agile and vice versa; that is, discouraging this difference for targets with no/marginal agility. To achieve this goal, in *step 2.1*, particles are clustered based on their current mode. The next target mode, $r_t$, is predicted using the previous mode, $r_{t-1}$, along with the transition matrix, $\Pi$ in *step 2.2.a*. At the same time, next target mode, $r_t^*$ is predicted using $r_{t-1}$ along with the regular MMPF in *step 2.2.b*. Next, the difference between these two particle clouds, i.e., $r_t$ and $r_t^*$, is calculated using KLD measure in *step 2.2.c*, followed by normalizing this value in *step 2.2.d*.

Table 3.2: Algorithm 2: SDCMMPF

---

$[\{y_t^n, w_t^n\}_{n=1}^N]$ =SDCMMPF $[\{y_{t-1}^n, w_{t-1}^n\}_{n=1}^N, z_t, SD]$

**Step 1:** Initialization of the particles

**Step 2:** Mode prediction

2.1 Cluster the input particle cloud into $M$ particle clouds, $PC_m$: $m = 1, ..., M$

2.2 For $PC_m : m = 1, ..., M$ (Figure 3.7):

a. Predict $r_t^*$ using transition probability matrix $\Pi$

b. Predict $r_t$ using the generic MMPF

c. Compute $KLDs$

d. Normalization

2.3 Model given soft data (Figure 3.1)

a. Define rules based on $RAL$

b. Model membership functions for the outputs $C_{m,m'}$ based on $RCL$

2.4 For each of the particle clouds $PC_m : m = 1, ..., M$:

a. Further cluster the cloud into $PC_{m,m'}$: $m' = 1, ..., M$

b. For $PC_{m,m'}$: $m' = 1, ..., M$

i. Select $FIS$ based on given $RAL$ & $RCL$

ii. $C_{m,m'} = FIS(ECW, SAD)$ $ECW \ \alpha \ KLD$ & $SAD \ \alpha \ \pi_{m,m'}$

iii. Apply $C_{m,m'}$ to the respective particles

**Step 3:** State prediction

**Step 4:** Updating

**Step 5:** Resampling

**Step 6:** Go to step 2

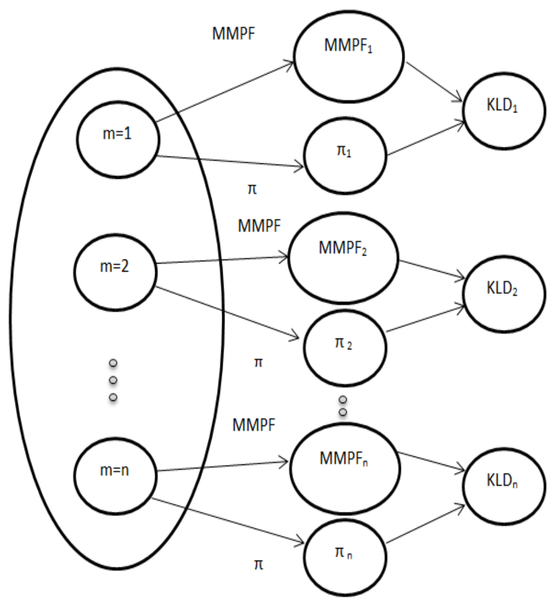


Figure 3.7: KLD measure calculation procedure including: predicting clusters using both MMPF & Π and computing KLD between the two predicted clusters

The KLD measure between the two distributions which are predicted by transition probability matrix and the MMPF is calculated as shown in Figure 3.7 using KLD measure [116]. Consider two distributions $p$ and $q$ and let the two sets $\{X_1, ..., X_n\}$ and $\{Y_1, ..., Y_m\}$ be i.i.d samples drawn independently from the $p$ and $q$, respectively. In [117], an asymptotically unbiased and mean-square consistent estimator $\hat{D}_{KL}(p, q)$ of $D_{KL}(p, q)$, based on the k-Nearest Neighbor (NN) density estimation [118], is defined as in the following:

In *step 2.3*, soft data is modeled using a Fuzzy inference system as discussed in the previous sub-section. One of the FISs is selected based on the RCL and RAL of the human report in *step 2.3.a*, and then ECW along with the SAD are the input to the selected FIS, shown in *step 2.3.b*. As shown in *step 2.4*, each of the particle clouds $PC_m, m = 1, ..., M$ are further clustered into $M$ sub-clusters at *step 2.4.a*. The constraint weights $C_{m,m'}$ are calculated in *step 2.4.b* as follows: If the target agility level, which is input by the user, is high and the estimation of the target location is close to that predicted by the transition probability matrix, i.e, a small KLD measure, then the particles that follow the behavior defined by Π should get low weights and gradually disappear. On the other hand, the rest of the particles, which are not behaving similar to model Π, should get higher weights, in order to duplicate (survive) more in the resampling step. The soft-data-inspired dynamic constraints affect the particles' weights before the resampling step, i.e., the weights are imposed in PF onto prior particles; therefore, the weighting of the particles is as follows:

$$w_t^n = w_{t-1}^n \; p(z|x) \; C_{m,m'} \tag{3.1}$$

in which the constant weights $C_{m,m'}$, defined as $C_{m,m'} = p(SD|x)$, are calculated in *step 2.4.i* of the algorithm. The following section presents the results of the experiments conducted to compare the performance of the proposed method with the generic MMPF.

## 3.3 Single Target Tracking Experiments

Three categories of experiments have been conducted. The first category evaluated the effect of human-agent reports on different target-agility levels as shown in Figures 3.8 to 3.13 and is discussed in Section 5.2.2. In the second category, the aim was to measure the impact of the level of uncertainty in these human-agent reports as shown in Figures 3.14 to 3.17 and is discussed in Section 5.2.3. The third category was conducted to show the robustness of the proposed method with respect to the varying agility levels, the results are shown in Figures 3.18 to 3.19 and are discussed in Section 5.2.4.

### 3.3.1 Experiments

The baseline used to assess the experimental results achieved by the proposed method is the generic MMPF, and the metric for evaluating the performance is the Mean Squared Error (MSE) between the estimated and the original target trajectories. In the experiments, the transition probability matrix, $\Pi$, was used and is defined as follows:

$$\Pi = \begin{bmatrix} 0.05 & 0.15 & 0.8 \\ 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \end{bmatrix}$$

The following modes indicate the behavior of a target: mode 1, moving straight East; mode 2, moving straight South East (SE); and finally, mode 3, moving North East (NE). For instance, to transit from mode 3 to mode 2, the target turns $-90^o$ and continues moving straight. Based on the transition matrix, when a target is at mode 1, it will most probably transit next to mode 3, and so on. Each target has periodic behavior with three maneuvers per period. In order to simulate medium agility, 1 or 2 (out of 3) maneuvers do not take place according to $\Pi$. When there is a high level of agility, no maneuvers take place according to $\Pi$. For example, as shown in Figures 3.12 and 3.16, the target is at mode 1 and remains at that mode.

### 3.3.2 Category I: Impact of Target Agility Level

In this category of experiments, three different scenarios were designed and performed and reported on by a human agent. In scenario I.a, shown in Figure 3.8, the report was *"robot is certainly marginally/not agile."* In scenario I.b, shown in Figure 3.10, the report was *"robot is certainly highly agile,"* and the target was tasked to periodically maneuver as follows:

$$mode1 \longrightarrow mode2$$

$$mode2 \longrightarrow mode1$$

Finally, in scenario I.c, shown is Figure 3.12, the report was *"robot is certainly extremely agile"* and the target was tasked to maneuver as follows:

$$mode1 \longrightarrow mode1$$

As shown in Figures 3.9 and 3.11, the proposed method improves tracking performance for both *medium* and *no agility*. In the case of an *extremely* agile target, as shown in Figure 3.12, regular MMPF cannot track the target. However, using soft data, the proposed method assigns higher weights to those particles that do not follow the maneuver characteristics defined by matrix $\Pi$, and assigns lower weights to the rest. The particles assigned higher weights will obtain even higher weights in the updating step, as they represent a target state more agreeable with the measurements. Consequently, in the resampling step, they survive and regenerate more; therefore, the estimation improves after some iterations.

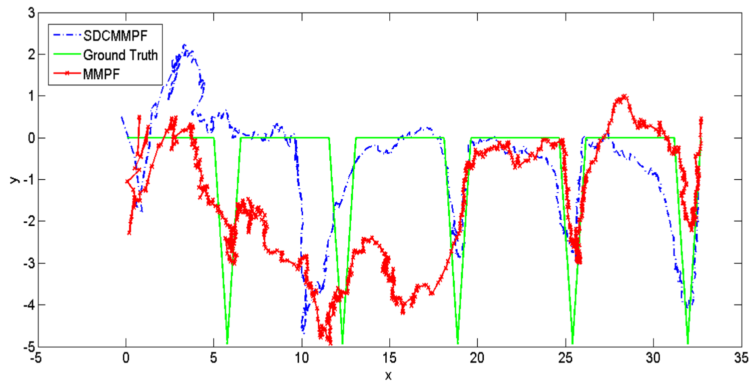


Figure 3.8: True & estimated target trajectory for scenario I.a

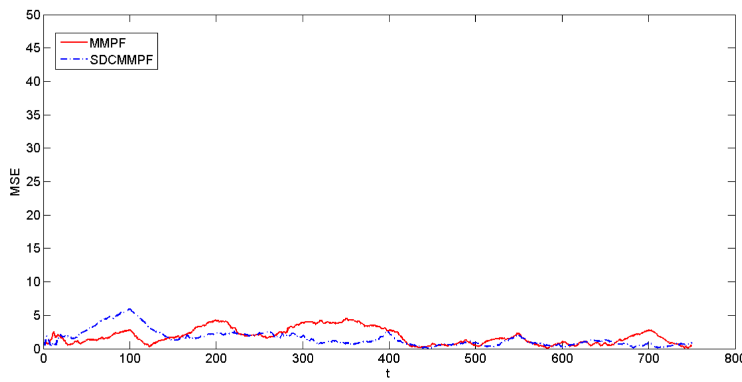


Figure 3.9: Performance comparison: low agile target & highly certain SD

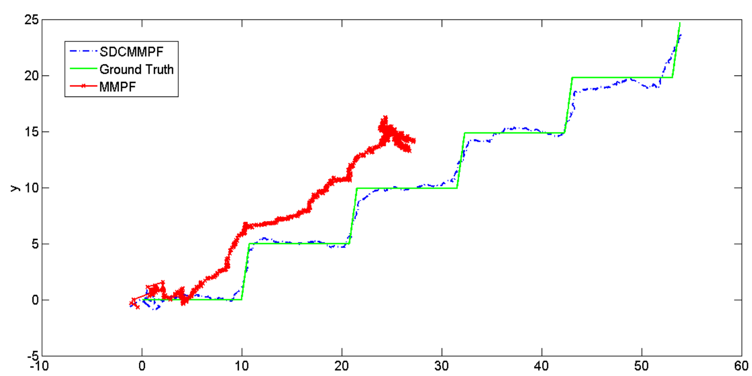


Figure 3.10: True & estimated target trajectory for scenario I.b

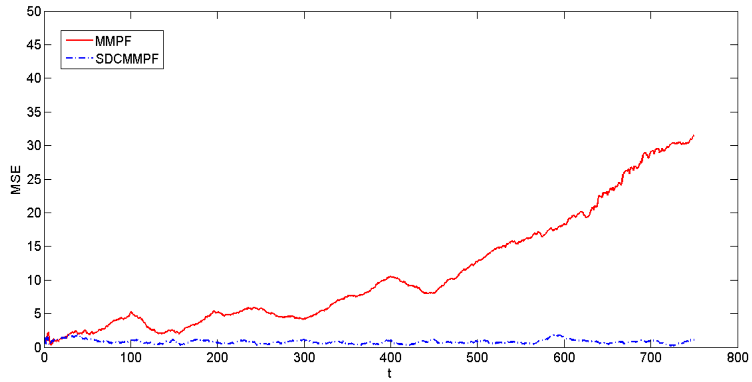


Figure 3.11: Performance comparison: agile target & highly certain SD

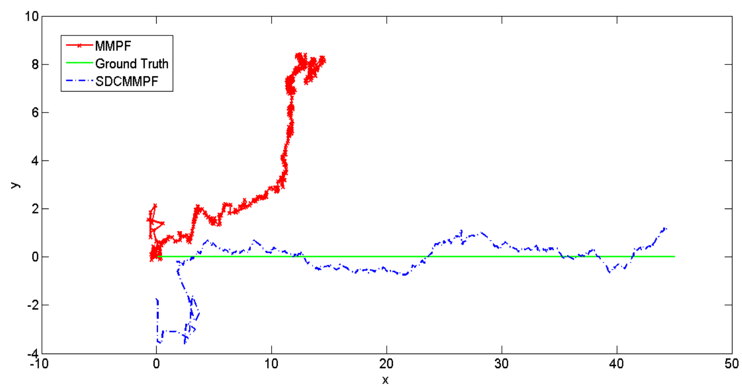


Figure 3.12: True & estimated target trajectory for scenario I.c

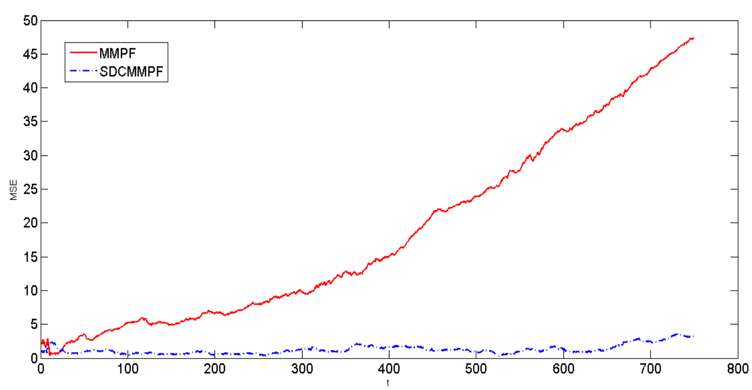


Figure 3.13: Performance comparison: highly agile target & highly certain SD

### 3.3.3 Category II: Impact of SD Certainty Level & Soft Data Certainty Level

In this set of experiments, the aim was to assess the effect of a human agent's level of uncertainty on estimation accuracy. Figures 3.14 to 3.17 show the results in which the human agent's uncertainty regarding the report was high. In the first scenario, the report was *"robot is perhaps highly agile,"* and in the second one, the report was *"robot is perhaps extremely agile."* As shown in Figures 3.14 to 3.17, when the certainty of the report given by the human agent was low; the estimation was not as accurate as the results obtained in the previous category, in which the user reported the agility level of the target with higher certainty.

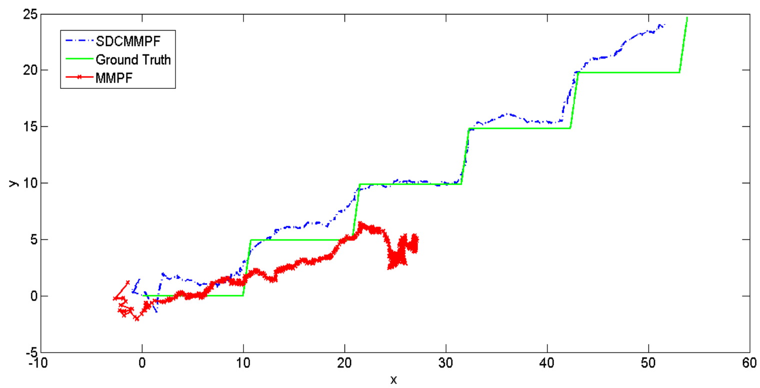


Figure 3.14: True & estimated target trajectory for scenario II.a

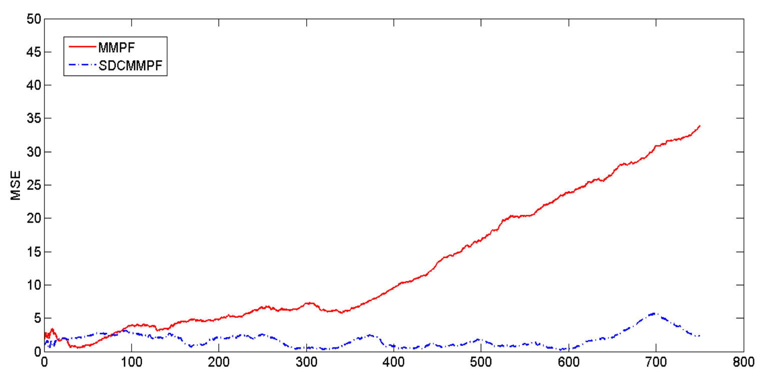


Figure 3.15: Performance comparison: agile target & uncertain SD

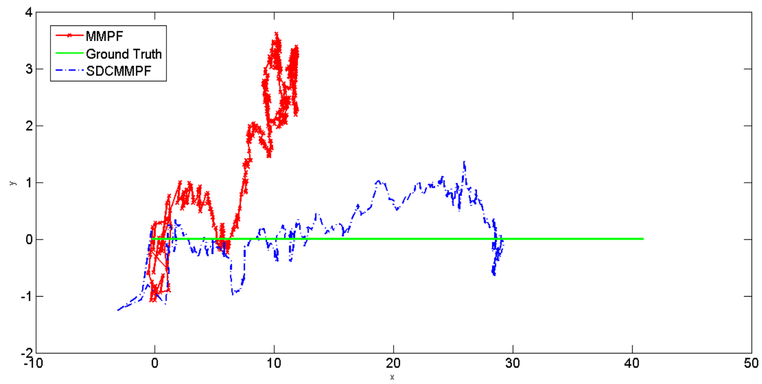


Figure 3.16: True & estimated target trajectory for scenario II.b

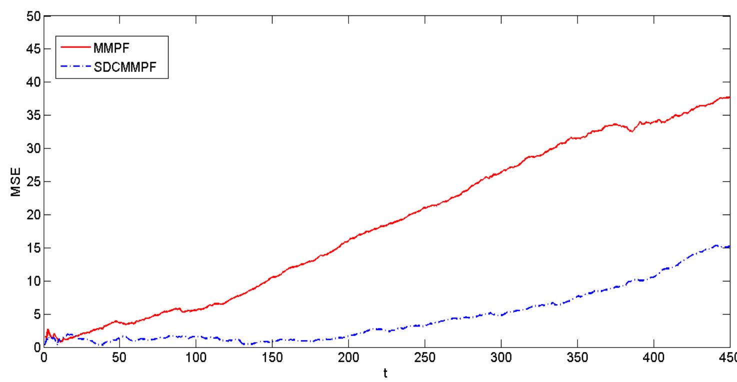


Figure 3.17: Performance comparison: highly agile target & uncertain SD

### 3.3.4 Category III: Impact of Varying Target Agility Level

In this experiment, the aim was to examine the robustness of the proposed method to varying agility levels, which is the most realistic case in practice. In this case, the target first maneuvers based on the characteristics defined by matrix $\Pi$, and after some time, it may start to deviate. Thus, the level of agility increases as time goes by. In this scenario, the report was given according to the situation of the target observed by the agent. The target was maneuvering without agility at the beginning, and after some iterations, it moved with a medium level of agility; it then continued with a high level of agility. Figure 3.18 shows the true and the estimated target trajectories based on MMPF and the proposed method. As shown, the proposed method can track a target while it is switching its behavior from not agile to highly agile; however, MMPF fails to track the target precisely and gets lost when the agility level becomes high. Figure 3.19 shows the performance comparison for the last scenario. The SD certainty level increases in this experiment, e.g. when target's behaviour is more agile the certainty level of SD increases. As shown in this figure in time $t = 900$ the target agility changes to be highly agile and at the same time the SD certainty level increases to be highly certain, consequently the MSE is lower in comparison with the case in which the target's agility level is medium and the SD certainty level is medium as well (t=450 to t=900).

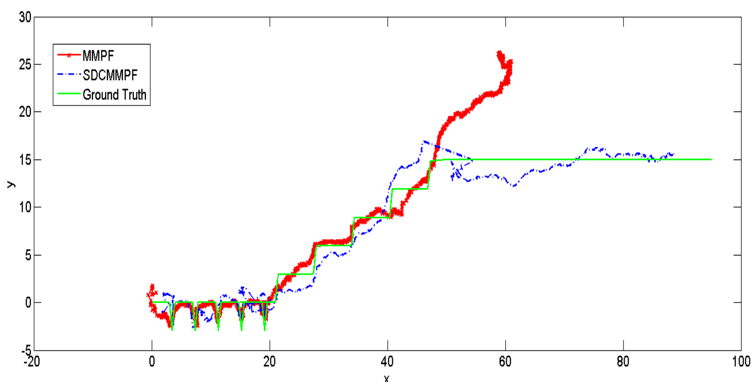


Figure 3.18: True & estimated target trajectory with varying agility levels

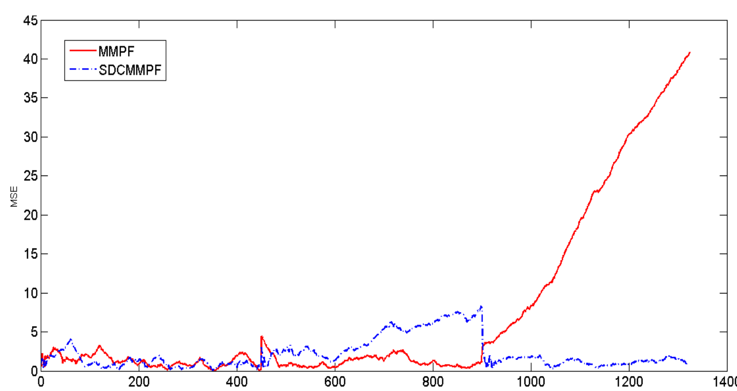


Figure 3.19: Performance comparison

## 3.4 Summary

This chapter highlighted the fact that soft data can provide important and necessary information for the estimation process. In order to use this high level information in the fusion process, different factors are studied and considered. Among the steps are how to gather this data, how to model it, and how to assign the uncertainty associated with it. The experimental results demonstrated the capability of the proposed method to significantly outperform the conventional MMPF when applied to various agile target-tracking scenarios. In particular, the conventional MMPF method is shown to diverge when applied to highly agile targets. In comparison, the proposed method is capable of tracking highly agile targets when provided with appropriate soft data. In the next chapter, the proposed distributed framework of the SDC MMPF is introduced and discussed.

# Chapter 4

# Distributed Soft-Data-Constrained Multi-Model Particle Filtering

## 4.1 Introduction

The problem of non-linear state estimation is an active field of research in sensor networks and data fusion research communities. The traditional centralized methods provide a flexible and powerful framework to solve this problem. However, they assume a fully connected network topology and require global communication among all sensor nodes, thus, suffer from issues such as a single point of failure, complexity and inflexibility of the routing data to a fusion center, high power consumption due to long-range communications, and inability to operate in partially connected networks. These known issues make them inapplicable to real-world applications involving sensor networks comprised of large number of sensor nodes. On the other hand, fully distributed solutions to this problem, which require each node to be merely aware of its neighbors and local communication among neighbouring nodes, have attracted researchers' attention in the past few years [1, 6, 7]. They provide an appealing alternative by improving upon the reliability, scalability, and ease of deployment of non-linear state estimation methods.

In Centralized Particle Filtering (CPF) approaches [65, 66], each node uses a local PF to estimate a local posterior using its measurement(s), and its estimate is then transmitted to a Fusion Center (FC). The FC then computes the global posterior and the global state estimate. FC-based DPFs are useful when the final estimate needs to be available only at a single central location; however, in the proposed method, the final estimate is available at every node at any time in the process. The communication requirements can be reduced by using approximate representations of the local posteriors, such as Gaussian representations [65] or histograms [119], transmitted from the sensor nodes to the FC. This way of communicating the parameters of the local posterior is also used among the neighbours in the proposed approach, in which the posterior at each node is the global posterior reflecting the current and the past measurements of all nodes.

Communicating particles among the nodes is costly; some works in the literature propose communicating certain selected particles over to the neighbouring nodes [120], such as the ones with the highest weights [121] in order to reduce the computational cost. However, the process of selecting the particles to transfer might be challenging and could reduce the ability to communicate sufficient information through the network. Alternatively, one can approximate the local particle cloud at each sensor node with a single or a mixture of Gaussians. This approach has the advantage of significantly reducing the communication cost involved in exchanging local estimates among nodes, which is required by distributed data aggregation schemes. As discussed by Gu *et al.* [122], the nodes sample their particles at each iteration from the aggregated global estimate.

A distributed non-linear estimation method based on Soft-Data-Constrained Multi-Model Particle Filtering (SDC MMPF) and applicable to a number of distributed state estimation problems is proposed. To make the MMPF work in a distributed manner, a Gaussian approximation of the particle cloud obtained at each sensor node and a Consensus Propagation (CP) based distributed data aggregation scheme are used to dynamically reweight the particles' weights. The constraints are enforced by adjusting particles' weights and assigning a higher mass to those closer to the global estimate represented by the nodes in the entire sensor network after each communication step. Each sensor node experiences gradual change; i.e., if a noise occurs in the system, the node, its neighbours, and consequently the overall network are less affected than with other approaches, and thus recover faster.

The main contribution of the proposed framework is to further develop our former algorithm to enable it to deal with distributed tracking scenarios, which has been motivated by recent trend in distributed data fusion schemes. In SDC MMPF the coefficients are computed based on only soft data, using a Fuzzy inference system; whereas, in the proposed distributed approach, these coefficients are influenced by both soft data and a likelihood function. The distance between the particle cloud at each sensor node and an aggregated global estimate are calculated and then a likelihood function is used to assign higher weights to the particles that are closer to the global Gaussian, and vice versa. The underlying distributed data aggregation scheme deployed is the consensus propagation algorithm [8]. Consensus prorogation is a solution for consensus problem in networks which is originally studied in control literature and has been applied to problem such as distributed coordination in multi-agent systems [123]. This procedure is repeated iteratively to allow information sharing among neighbouring nodes, and consequently the whole network, with the objective of diverting particle clouds of all sensor nodes towards the global aggregate.

## 4.2 Distributed Particle Filtering

The literature offers some designs that use distributed Bayes filters in sensor networks, either in a centralized or a distributed manner. In the first category, all the nodes send their information to a base station (center-based fusion) [65, 66] that performs all calculations and sends the final estimation back to all nodes in the network. This approach requires extensive communication and is very costly. Huge numbers of communications require a large amount of energy and offer a possible failure point at the central node. To overcome these issues, distributed strategies are an alternative providing a more general and robust solution, with fewer communications and the possibility of parallel processing [67, 68, 69, 70]. Decentralized Kalman filtering [124] has been proposed for a decentralized control problem, in which the network is fully connected. The same assumption is used in [125]. In [71], the Kalman filtering iterations are parallelized over a set of sensors; however, it still requires a fusion center to combine the estimates. Distributed particle filtering approaches are more effective for large-scale, nonlinear and non-Gaussian distributed estimation problems [72, 73].

Based on the type of data communicated between nodes, Distributed Particle Filtering (DPF) is classified into two types [74, 126]: statistical dissemination-based, in which processed data is exchanged between nodes [76, 127, 77], and measurement dissemination-based, in which raw measurements are exchanged [78], [79]. Different statistic dissemination-based methods exist, varying in their scheduling and communication topology. The proposed method lies in the category of statistics dissemination-based methods, and the communication among nodes is consensus-based, which means that all nodes in the network process the data simultaneously [80], [81].

The distributed state estimation methods mainly rely on distributed data aggregation schemes as their underlying enabling technology. The most common distributed data aggregation schemes are gossip-based consensus filters [84], message passing (belief propagation) algorithms [85], and data diffusion processes [86]. A distributed consensus filter is proposed in [81] where each sensor can communicate with the neighboring sensors, and only a small fraction of sensors need to measure the target information, with which the whole network can be controlled.

Each of these algorithmic categories has its own benefits and drawbacks. Additionally, a few hybrid approaches aim at providing a framework for developing algorithms based on existing methods and try to minimize their inherent restrictions as much as possible. Consensus algorithms are used for distributed computations [87, 80]. According to the context, consensus means a global agreement on some quantity that depends on the data of all nodes [74]. The underlying distributed data aggregation scheme deployed for the proposed method is the consensus propagation algorithm [8], which is a hybrid of Consensus Filtering (CF) and belief propagation methodologies.

CP is derived based on a simple yet elegant observation and is thus easy to expand and implement. Furthermore, it has been proven to converge, even when performed asynchronously, with the convergence time scaling gracefully with respect to a network's size [8]. The data (messages) sent to each of the neighbouring sensor nodes are specific to that node, despite CF in which messages are broadcast to all neighbours at each iteration. Moreover, using CP, the messages sent to neighbours contain the latest estimate of the desired parameter as well as the number of sensor nodes contributing to that estimation. On the other hand, similar to the CF approach, CP is a distributed protocol for averaging; i.e., it allows each node to obtain an estimate of the global average in a network while requiring information exchange among local (neighbouring) nodes.

CP is different from the diffusion strategies for distributed filtering [128] as the former is an iterative approach, i.e., it requires information exchange among neighbouring nodes until convergence, while the latter does not require more than one iteration to yield the global average every time. Although selecting the optimal weights for the diffusion approach can be challenging and requires solving an optimization problem in real-time, as shown in [129] and [130]. Recently, some diffusion protocols have been proposed that do not require one to select optimal diffusion weights [131, 132].

### 4.2.1 Gaussian Particle Filter

This section briefly describes the Gaussian filtering process that is deployed to fit a Gaussian or a mixture of Gaussians to a particle cloud. Some of the existing methods proposed for this step include using Expectation-Maximization (EM) [133], [134], [135] and K-means clustering. In the proposed method a Gaussian particle filtering technique described in [160] is adopted to fit a Gaussian to the local particle clouds as follows:

$$\mu_t = \frac{1}{N} \sum_{n=1}^{N} w_t^n x_t^n \tag{4.1}$$

Table 4.1: Algorithm 3: Distributed SDC MMPF

---

For sensor node s=1:S

**Step 1:** Perform SDC MMPF [104]

**Step 2:** Fit Gaussian to the local particle cloud

$E_t^{local(s)} = \frac{1}{N}\sum_{n=1}^{N} w_s^n x_s^n$

$\overline{\sum}_t^s = \frac{1}{N}\sum_{n=1}^{N} w_s^n (E_t^{local(s)} - x_s^n)(E_t^{local(s)} - x_s^n)^\top$

**Step 3:** Local message exchange between nodes $i$ and $j$

For r=1:R

$E_t^{global(j)} = \frac{E_t^{local(j)} + \sum_{i\in N(j)} M_t^{ij} K_t^{ij}}{1+\sum_{i\in N(j)} K_t^{ij}}$

$M_t^{ij} = \frac{E_t^{local(i)} + \sum_{l\in N(i)\backslash j} M_{t-1}^{li} K_{t-1}^{li}}{1+\sum_{l\in N(i)\backslash j} K_{t-1}^{li}}$

$K_t^{ij} = \frac{1+\sum_{l\in N(i)\backslash j} K_{t-1}^{li}}{1+\frac{1}{\gamma}(1+\sum_{l\in N(i)\backslash j} K_{t-1}^{li})}$

**Step 4:** For $n = 1 : N$ (reweight local particle cloud w.r.t. $E_t^{global(s)}$ )

$d_s^n = \left\| x_s^n - E_t^{global(s)} \right\|$

$\mathcal{L}(x_s^n) = p(x_s^n | E_t^{global(s)}) = e^{\{\frac{-d_s^n}{-d_s^n+\beta}\}}$

$\alpha_s^n = \mathcal{L}(x_s^n)$

$\{w_i^n = w_i^n \times \alpha_i^n\}_{n=1}^N$

Resampling

**Step 5:** Go to step 1

$$\Sigma_t = \frac{1}{N}\sum_{n=1}^{N} w_t^n(\mu_t - x_t^n)(\mu_t - x_t^n)^\top \tag{4.2}$$

in which $N$ is the number of particles and $\mu$ and $\Sigma$ are the mean and the variance of the Gaussian, respectively,. The above approach can be extended to enable fitting a mixture of Gaussians to a particle cloud as follows:

$$\widehat{\mu}_t = \sum_{i=1}^{G} w_t^i \mu_t^i \tag{4.3}$$

$$\widehat{\Sigma}_t = \sum_{i=1}^{G} w_t^i(\Sigma_t^i + (\widehat{\mu}_t - \mu_t^i)(\widehat{\mu}_t - \mu_t^i)^\top) \tag{4.4}$$

In (4.3) and (4.4), $G$ represents the number of mixands used. In (4.1) and (4.2), $w_t^n$ defines the weight of each particle $n$ at time $t$ and in (4.3) and (4.4), $w_t^i$ defines the weight of each mixand at time $t$. $\widehat{\mu}$ and $\widehat{\Sigma}$ are used to define the overall average estimation and its uncertainty (variance), respectively. Some of the major difficulties of this process are highlighted next. Choosing the number of Gaussians is challenging, especially if there is severe nonlinearilty in the model. Collapsing of the mixands can occur, and when it happens, the posterior distribution must be re-expressed as a Gaussian Mixture having small covariances. The covariance of the mixands grows especially when the covariance of the process noise is larger than the covariance of the mixands [137]. The parameters of the Gaussian Mixture Model are exchanged between the neighbour sensor nodes, in which for every node, $s = \{1, ..., S\}$, $E_t^{local(s)} = \mu_t$ , i.e. the local estimation of node $s$ and $\Sigma$ is the uncertainty of the estimation.

### 4.2.2 Distributed Aggregation Using Consensus Propagation

After the posterior of each node is approximated by a Gaussian, the aggregated global estimate is calculated using Consensus Propagation (CP) by sharing the Gaussian approximation parameters among the neighbouring nodes.

In this protocol, if a node communicates to a neighbouring node at time $t$, it transmits a message consisting of $M^{li}$ and $K^{li}$ which denote the values associated with the most recently transmitted message from $l$ to $i$ at or before time $t$ and the number of nodes

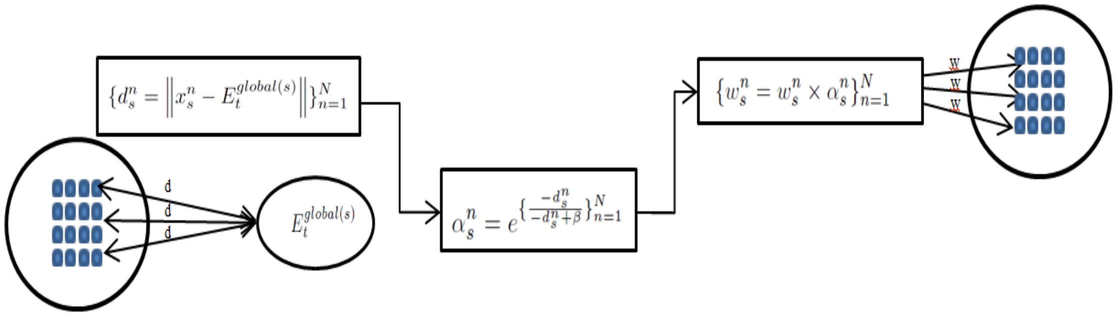


Figure 4.1: Updating weights based on likelihood function (Algorithm 3: Step4).

contributing to that estimation process respectively. At each time $t$, node $i$ has stored in memory the most recent message from each neighbor: $M^{li}, K^{li}, l \in N(i)$. If at time $t$, node $i$ chooses to communicate with a neighboring node $j$, it constructs a new message that is a function of the set of most recent messages $M^{li}, K^{li}, l \in N(i) \setminus j$ received from all neighbors other than $j$, $N(i)$ indicated all node's $i$ neighbouring nodes. Figure 4.2 illustrates the procedure of the message exchange between nodes $i$ and $j$.

$M^{ij}$ is a message from the nodes that are within a certain distance from node $i$, and $K^{ij}$ is the cardinality of this set, i.e., the number of nodes involved in this estimation. This information, along with the local estimate $E_t^{local(j)}$, are used to update the global estimate $E_t^{global(j)}$ of the desired parameter:

$$E_t^{global(j)} = \frac{E_t^{local(j)} + \sum_{i \in N(j)} M_t^{ij} K_t^{ij}}{1 + \sum_{i \in N(j)} K_t^{ij}} \tag{4.5}$$

in which, $M_t^{ij}$ and $K_t^{ij}$ are calculated as follows:

$$M_t^{ij} = \frac{E_t^{local(i)} + \sum_{l \in N(i) \setminus j} M_{t-1}^{li} K_{t-1}^{li}}{1 + \sum_{l \in N(i) \setminus j} K_{t-1}^{li}} \tag{4.6}$$

$$K_t^{ij} = \frac{1 + \sum_{l \in N(i) \setminus j} K_{t-1}^{li}}{1 + \frac{1}{\gamma}(1 + \sum_{l \in N(i) \setminus j} K_{t-1}^{li})} \tag{4.7}$$

In (4.7) $\gamma > 0$ is a constant used to control the attenuation level of the CP. The intuition

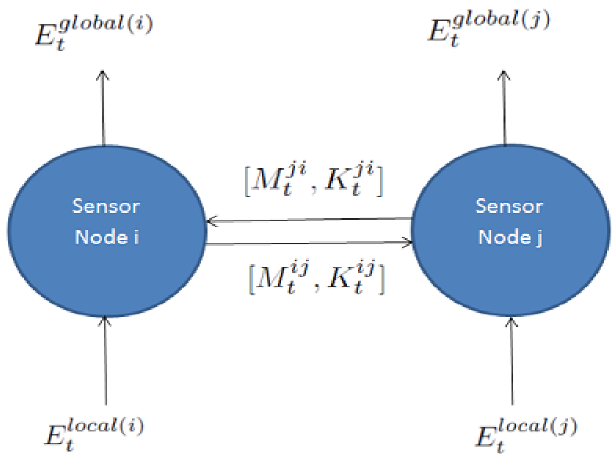


Figure 4.2: Distributed computation of global estimate using CP.

behind this attenuation process is to avoid the unbounded growth of $K$ in sensor networks with cycles. It is easy to see that the larger the $K$ and smaller the $\gamma$, the stronger the attenuation process would be. The convergence properties of this approach are proven and discussed in detail in [112]. At every node, the global Gaussian mean and its covariance are calculated by sharing the information as shown in equations (4.5) to (4.7). This global Gaussian is then used to infer the constraints, which are then enforced to update the particles' weights.

The CP algorithm is an iterative approach and in our experiments CP has been performed multiple times between measurement arrivals to reach consensus and obtain an estimate of the aggregated global estimate for all sensor nodes. However, the consensus among particle clouds of all sensor nodes is still achieved over time by adjusting the particle weights of each sensor node based on the likelihood function.

### 4.2.3 Enforcing Constraints Based on Global Aggregate

In most state-of-the-art DPF approaches, the aim is to approximate the aggregated global estimate to draw the samples directly from it for each sensor at each iteration. Therefore, after sharing the information, once the global posterior is achieved, each node draws new samples randomly from the global posterior. For instance, in some approaches, at each iteration, nodes sample their particles from the estimated Gaussian Mixtures (GM). In [134] an Expectation-Maximization (EM) algorithm is deployed to estimate the Gaussian Mixture Model (GMM). Most work in this category follows the same principle of fitting

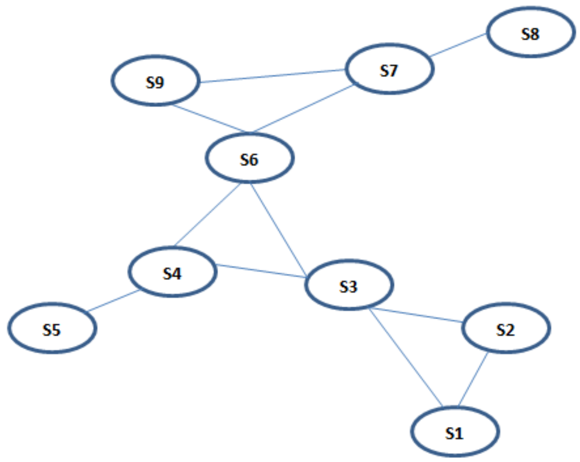

Figure 4.3: Sensor network topology used for the experiments.

a Gaussian to the particle cloud and then sampling from the aggregated global estimate distribution [138], [139], [140]; however, in the proposed approach, this global posterior is used to calculate the constraints and apply them to particle clouds at each sensor node.

In the proposed approach, the aim is to infer the constraints based on the distance of the local particles to the aggregated global estimate and to use this distance to re-weight the particles accordingly, instead of generating a new set of particles at each iteration.

A likelihood function (4.9) is used to assign higher weights to the particles that are closer to the global Gaussian, and vice versa. This procedure of updating weights occurs after the updating step of the PF algorithm, and before the resampling step. Therefore, after the resampling step, each node has a new population of particles that has the posterior distribution representing an estimate closer to that for the rest of the neighbouring nodes, and at the same time, closer to the global estimation. The distance is calculated as follows,

$$\{d_s^n = \left\| x_s^n - E_t^{global(s)} \right\| \}_{n=1}^N \tag{4.8}$$

and the likelihood function is computed as below:

$$\mathcal{L}(x_s^n) = p(x_s^n | E^{global(s)}) = e^{\{\frac{-d_s^n}{-d_s^n + \beta}\}_{n=1}^N} \tag{4.9}$$

$$\alpha_s^n = \mathcal{L}(x_s^n) \tag{4.10}$$

In the next step, the constraints are applied to the particles' weights:

$$\{w_s^n = w_s^n \times \alpha_s^n\}_{n=1}^N \tag{4.11}$$

In these equations, $N$ is the total number of particles in each particle cloud, $x_s^n$ is the estimation of the $n^{th}$ particle of sensor node $(s)$ regarding target location and $w_s^n$ is the weight of the $n^{th}$ particle. $E_t^{global(s)}$ is the parameter that is obtained using CP for node $s$, $\beta$ denotes the maximum possible distance between each particle and $E_t^{global(s)}$ (which depends on the environment and is calculated as $\beta = Max(d_s^n)_{n=1}^N$). In the next step, the constraints are applied to the particles' weights. Using the likelihood function shown in (4.9), the particles' weights are updated based on the global information of the network. After weight update in step 4 of Algorithm 3, the particle set may not be properly weighted, i.e., the summation of the weights might not be equal to one; however, a resampling step after this step makes sure this issue is resolved. The complete procedure of this step is presented in Figure 4.1. Algorithm 3 shows the complete procedure of the proposed method, in which $S$ is the total number of nodes in the network.

## 4.3 Single Target Tracking Experiments in Distributed SDC MMPF

### 4.3.1 Experiments

The network for the experiments consists of nine nodes, as shown in Figure 4.3 each node communicates only with its neighbouring nodes, as shown by the links. Three categories of experiments have been conducted. The first examines the effect of noisy measurements on tracking convergence properties. The second evaluates the tracking performance with respect to the number of particles used. The third studies the effect of incorrect soft data on tracking performance. The Root Mean Squared Error (RMSE) and the Average RMSE (ARMSE) are deployed as the performance metrics in the experiments.

Figure 4.6 has been deployed to show the Average RMSE and its variance for all the nodes in the network, as calculated in equations 4.13 and 4.14, respectively.

$$RMSE_t^s = \sqrt{\frac{\sum_{n=1}^N (x_t^{true} - \hat{x}_t^n)(x_t^{true} - \hat{x}_t^n)^\top}{N}} \tag{4.12}$$

$$ARMSE_t = \frac{\sum_{s=1}^{S} RMSE_t^s}{S} \tag{4.13}$$

$$\Sigma_{RMSE} = \sqrt{\frac{\sum_{s=1}^{S}(RMSE_t^s - ARMSE_t)^2}{S}} \tag{4.14}$$

The Root Mean Squared Error ($RMSE_t^s$) for each node $s$ at time $t$ is evaluated using equation 4.12, where $x_t^{true}$ is the true location of the target at time $t$ and $\hat{x}_t^s$ is the estimation of the $n^{th}$ particle. The average of the estimations over all of the nodes in the network at time $t$ ($ARMSE_t$) is calculated as shown in (4.13), in which $S$ indicates the total number of nodes. Equation 4.14 depicts the variance (uncertainty) of the overall estimation of the network and is evaluated by averaging the difference of all nodes' RMSE and the average error of the network.

Two methods, a distributed particle filter methodology, referred to as Baseline Distributed Particle Filtering (B-DPF), and Centralized Particle Filtering (CPF), are used as baselines. In the former, as discussed by Gu *et al.* [122], the nodes sample their particles at each iteration from the aggregated global estimate. In the latter, a central processing unit, referred to as a fusion center, is deployed. The local posterior estimated by each node is then sent to the fusion center to compute the global posterior.

The test scenarios in the first two categories are performed to examine the effect of the additive noise on a noisy node's estimation as well as that of its immediate neighbours.

### 4.3.2 Category I: Effect of the Additive Transient Noise on the Sensor Node

Figure 4.4 depicts the experimental results of a test scenario wherein a white Gaussian random noise is added to the observation measurements of one of the sensor nodes, i.e., S6. The additive noise is applied for a predefined time interval; i.e., in the total number of iterations, which is 250, the additive noise is only applied between iterations 150 to 200. As shown in Figure 4.4, once the noise is introduced, none of the examined methods can estimate the true target trajectory precisely. However, although all three methods gradually recover, the proposed method recovers faster, as shown in Figure 4.5. This fact can be attributed to the underlying sampling mechanism used by each method.

Using the B-DPF method, the data exchange is more explicit, as each sensor obtains its particle cloud by directly sampling from the aggregated global estimate. On the other hand, the proposed method (P-DPF) recovers faster because the aforementioned implicit data exchange mechanism restricts the influence of the noisy measurements on the neighbouring nodes, and S6's estimation improves over time, reaching consensus with its neighbouring nodes. Figure 4.6 depicts the overall performance of the network over time. As shown, the proposed approach achieves convergence to a progressively more accurate estimate over time, as demonstrated by the increasingly smaller RMSE mean (ARMSE) and variance over time. Moreover, as expected the overall performance is slightly deteriorated by introducing the transient noise, but recovers fairly rapidly as soon as noise is no longer present. Figures 4.6, 4.9 and 4.12 all represent the RMSE variance considering the estimate from all nodes in the entire network.

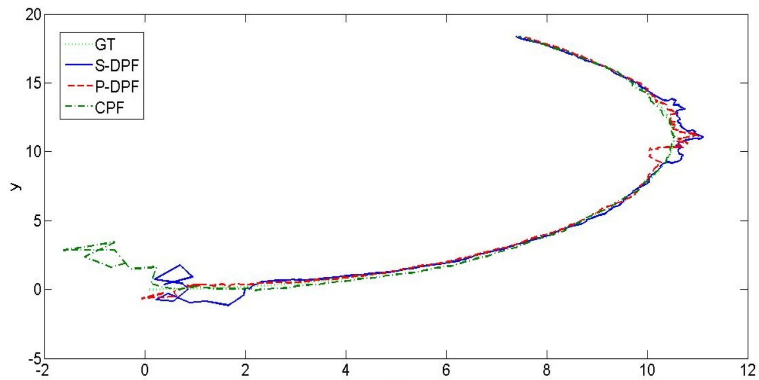


Figure 4.4: Experimental results obtained for the test scenario in which a white Gaussian random noise is added to the observation measurements of one of the sensor nodes for a predefined time interval: a) Ground truth (GT)

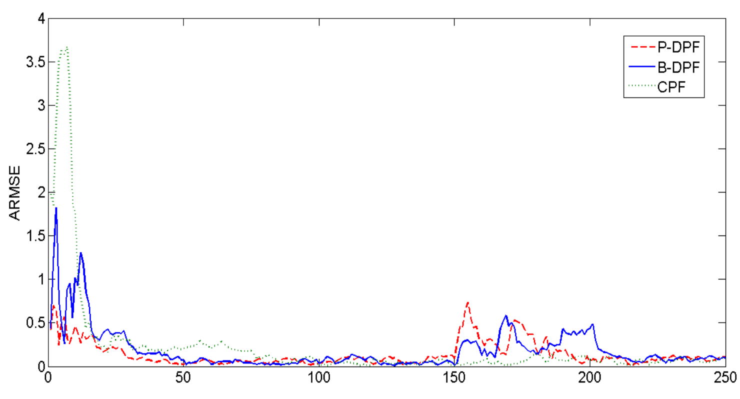


Figure 4.5: Performance comparison for the noisy node

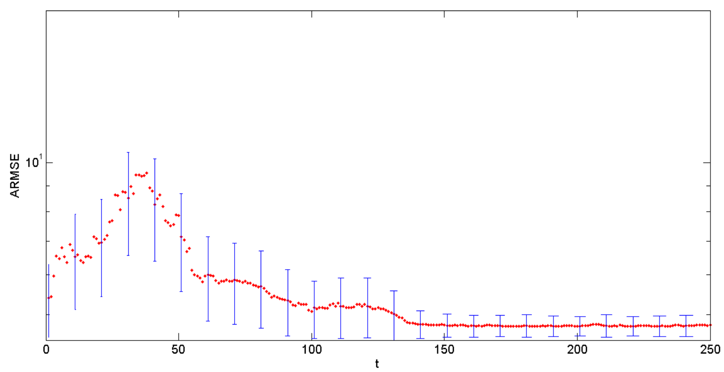


Figure 4.6: Overall network performance

### 4.3.3 Category II: Effect of the Noisy Neighbours on the Sensor Node

Figure 4.7 depicts the true and the estimated target trajectories for, S4, an immediate neighbour of the noisy sensor nodes (S3, S5, S6). A comparison of ARMSE obtained over time is shown in Figure 4.8. The results illustrate that compared to the baseline method B-DPF, using the proposed method, the immediate sensor node's estimation is less affected by the noisy measurements.

Using the proposed method, the particles' weights are affected by the implicit data exchange; i.e., each node keeps its particle population and just updates their weights based on the aggregated global estimate. Therefore, the effect of the noisy measurements on neighbouring sensor nodes is alleviated and overall network experiences a gradual change. In contrast, using the B-DPF, each neighbouring sensor node directly samples its particle population from the global aggregate, which is explicitly affected by the noisy measurement. Figure 4.9 depicts the overall performance of the network over time, which essentially represents the same trend as that of previous experiment shown in Figure 4.6. However, slightly larger RMSE mean and variance are obtained over time, which could be attributed to having a larger number of noisy sensor nodes.

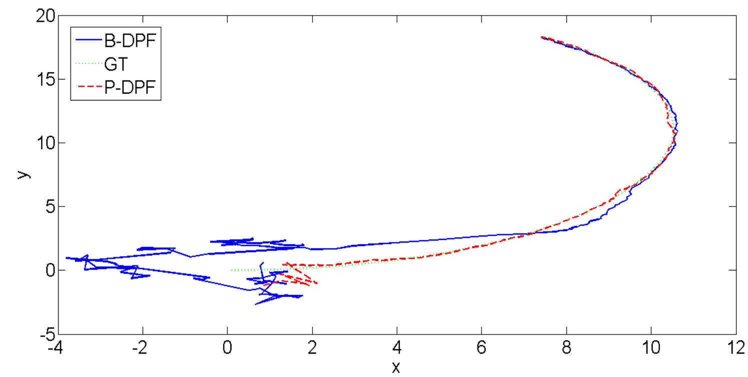


Figure 4.7: Experimental results to study the effect of noise on an immediate neighbour of the noisy sensor nodes: a) Ground truth

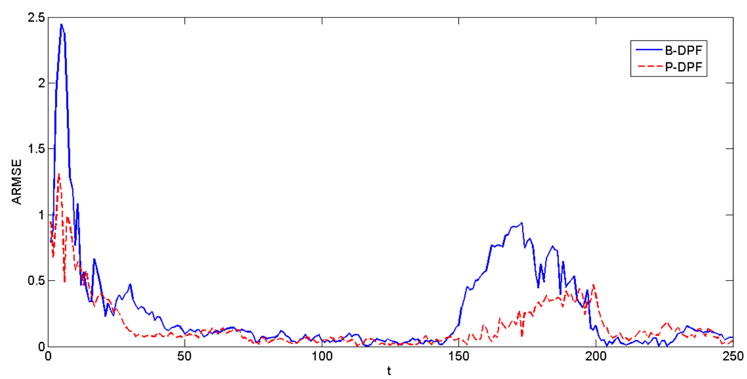


Figure 4.8: Performance Comparison for the neighbouring node

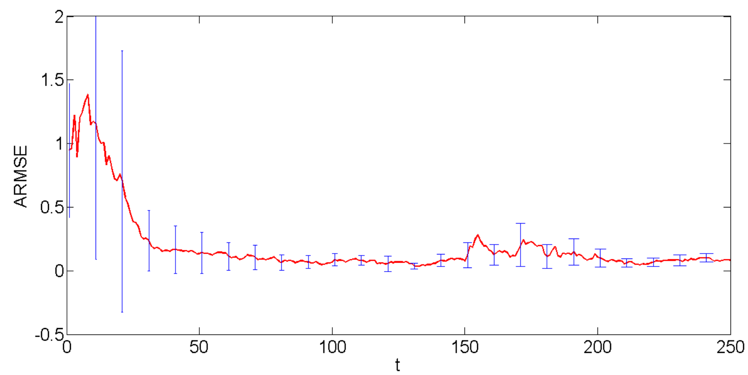


Figure 4.9: Overall network performance

### 4.3.4 Category III: Robustness to Incorrect Soft Data

This section, evaluates the proposed method's efficiency in dealing with erroneous soft data. In distributed case, the node receiving the incorrect soft data can enhance its estimation by communicating with its neighbouring nodes and taking advantage of their valid soft data to calculate constraints and applying them to its particle cloud. The target is highly agile in this case, and the soft data report for the faulty sensor node, i.e. S6, is "robot is certainly low agile", which is invalid regarding the aforementioned target dynamics. However, the report provided to the neighbouring sensor nodes is "robot is certainly highly agile", which truly describes the target dynamics.

In this case, as there is no other distributed approach based on particle filtering that incorporates soft data to refine the estimation process, the scenario wherein the faulty sensor node's neighbours are not supplied with any soft data report is considered as the baseline, i.e., only a single node (SN) is provided with incorrect soft data. Figure 4.10 depicts the true and the estimated target trajectories for this test scenario. A comparison of the obtained RMSE is shown in Figure 4.11. The results demonstrate the ability of the proposed method to yield fairly accurate tracking results, whereas in the baseline case, the filter diverges over time. In this case node S6 does not diverge but the overall performance of the network is affected by the continuous incorrect soft data received by node S6 as shown in Figure 4.12.

As mentioned, the proposed method operates by enforcing the constraints, which are in turn based on the global aggregate. In the baseline case, there is no correct soft data report regarding the existence of target agility, and thus, all particle clouds and hence their global aggregate will be corrupted. Accordingly, the filtering process diverges over time.

On the other hand, using the proposed method, the corrupted particle cloud of the faulty sensor node is aggregated with those of its neighbouring sensor nodes, which are accurate. Therefore, as shown by Figure 4.12 the proposed method is capable of avoiding divergence while providing acceptable level of tracking accuracy over time.

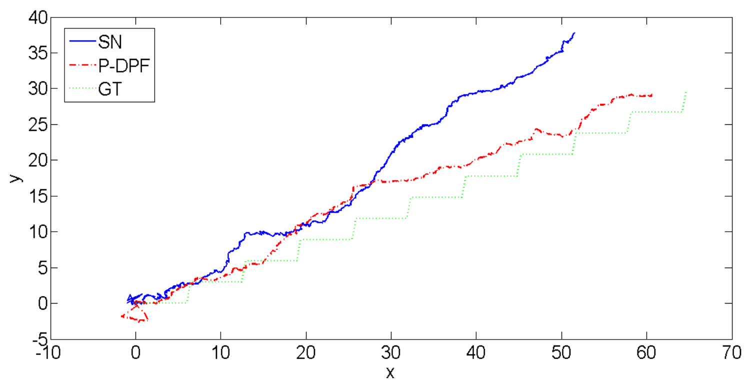


Figure 4.10: Experimental results for evaluating the efficiency of the proposed method in terms of dealing with erroneous soft data: a) Ground truth (GT) & estimated target trajectory

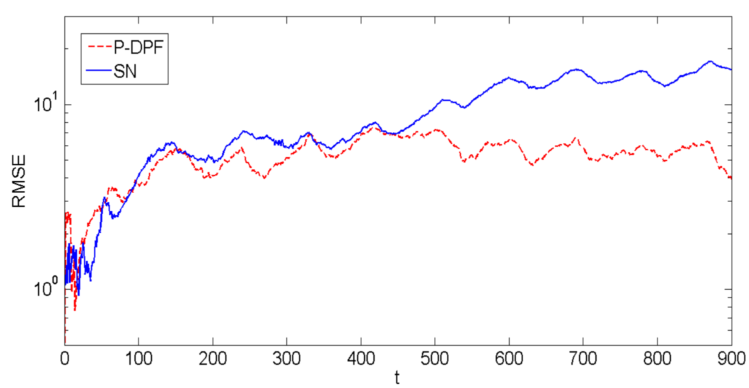


Figure 4.11: Performance Comparison for incorrect-soft-data test scenario

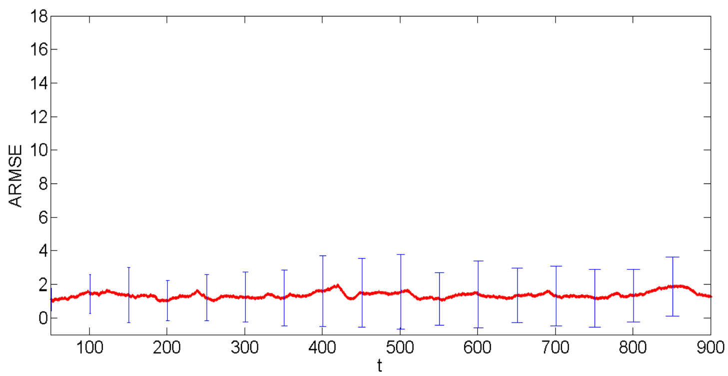


Figure 4.12: Overall network performance

### 4.3.5 Category IV: Performance Evaluation Based on Number of Particles

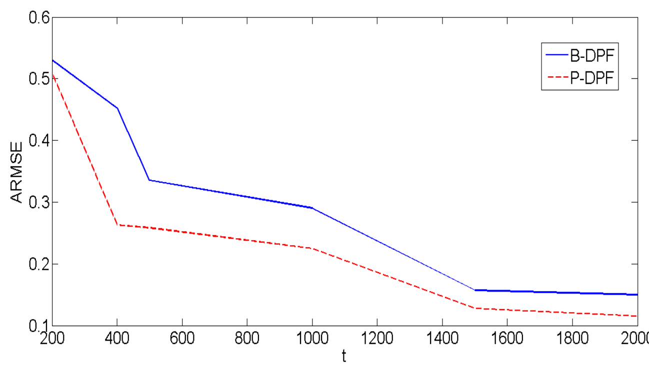


Figure 4.13: Performance vs. number of particles

The accuracy of the approximation is directly proportional to the size of the particle set $N$; increasing the total number of particles increases the accuracy of the approximation, but also increases the computational cost. In other words, the number of particles, $N$, is a trade-off between the accuracy and the computational resources. Therefore, being able to achieve the same performance using a smaller number of particles is highly advantageous when using particle-based approaches.

Figure 4.13 compares the performances of the proposed method and the baseline method B-DPF in terms of the number of particles used. Using the proposed method, all the nodes receive the report "robot is certainly low agile", which is a correct soft-data report regarding the agility level of the target. As shown, for both methods, a higher number of particles results in more-accurate tracking estimation. However, using the same number of particles, the proposed method consistently yields a lower ARMSE, over time. ARMSE in this case represents the average RMSE over the entire network and iterations. In particular, the performance gain is exponential before a certain threshold, which is about 400 particles in our experiments, but becomes less pronounced once the number of particles passes that threshold. More experiments can be done as an extension to evaluate the effective number of particles which is required and to evaluate the number of particles for which the performance of both approaches get closer.

## 4.4 Summary

In this chapter a distributed framework for the proposed method was discussed and experiments involving a single target tracking scenario in a distributed sensor network were carried out. Robustness to noisy sensor measurements was evaluated in two different categories: evaluating the effect of the additive transient noise on the sensor node and evaluating the effect of the noisy neighbours on the sensor's estimation. The third category of experiments presented the robustness of the proposed method to incorrect soft data. The proposed method is a sampling-based approach; therefore, the forth category of experiments demonstrated the performance evaluation of the proposed technique based on different numbers of particles. The proposed method can recover from failure situations and is robust to noise through communication with neighbouring nodes; keeping the same population of particles and updates their weight using the constraints inferred from the aggregated global estimate.

In a distributed framework, it is often considered that all of the sources are sharing their information with a same reliability; however, in the next chapter we discuss the importance of considering the sources' reliability in a distributed framework and a novel technique is proposed to tackle this issue.

# Chapter 5

# A Source-Reliability-Adaptive Distributed Soft-Data-Constrained Multi-Model Particle Filter

## 5.1 Introduction

The main body of the literature on information fusion concerns with building an appropriate uncertainty model without paying much attention to the related problem of reliability of these models and fusion results [141]. The majority of fusion operators assume that the models producing beliefs are equally reliable and play a symmetrical role in the fusion process. However, in reality different models may have different reliability and it is necessary to account for this fact in order to avoid decrease in the performance of fusion results and estimation process. Most reliability modeling schemas are derived from neighborhood information according to a distance metric [142, 143] or to likelihood functions [144, 145, 146]. Either an inappropriate choice of distance metric or a poor estimation of the likelihood function can lead to an inadequate belief model and, consequently unreliable beliefs being combined [141]. When combining information provided by many sources, the range and the limitations of the belief model used for each source should be taken into consideration. The most natural way to deal with this problem is to establish the reliability of the beliefs computed within the framework of the model selected. This may be achieved by using reliability coefficients, which introduce the second level of uncertainty and represent a measure of the adequacy of the model used and the state of the environment observed.

## 5.2 Accounting for Source Reliability in Information Fusion

At least two approaches are used for defining reliability as a higher-order uncertainty, i.e., uncertainty about uncertainty [147]. In one approach, reliability is understood as the relative stability of the first order uncertainty. In this case, reliability is often measured by the performance of each source, e.g., by the recognition or false alarm rates. The other approach is based on measuring the accuracy of predicted beliefs; in which, the reliability coefficients represent the adequacy of each belief model to the reality. The value of reliability coefficients may be provided by external sources, modeled by utilizing contextual information [148, 149], learned by using training data, as e.g., in a neural network [143, 150], or constructed as a function of agreement between different sources or sources and fusion results [151, 152].

One of the approaches to modeling source reliability is based on consensus among various sources or a degree of consensus among sources and fusion results [141]. One such method designed for target tracking [152] adaptively computes a deviation between measurements of each sensor and the fusion result, then uses this deviation measure to assess the degree of a source's reliability. A different consensus-based method utilizes the notion of inner trust introduced in [75]. Evaluation of inner trust is performed in two steps. First, a pairwise likeliness of the sources is computed, and then the inner trust is defined in such a way that a source is considered absolutely reliable if and only if there is no contradiction with other sources, while a source in absolute contradiction has very small reliability. The proposed approach evaluates the reliability of each source based on consensus among a node and its neighbouring nodes.

The problem of source reliability is related to the problem of conflict. Indeed, the existence of conflict indicates the existence of at least one unreliable source. On the other hand, unreliable sources might agree, and the absence of the conflict does not guarantee the reliability of sources. So, in the unlikely case where the majority of nodes fail, and there is also no conflict, detecting the unreliable sources becomes a very challenging problem. A potential solution to improve performance in such cases is to exploit any external information that becomes available [151].

### 5.2.1 Adaptive Distributed Soft-Data-Constrained Multi-Model Particle Filtering

Relying on distributed SDCMMPF method [153] and a weighted variant of the consensus propagation algorithm, a new approach for source-reliability-adaptive distributed estimation using soft and hard data is proposed and discussed in this section [154]. In Centralized Particle Filtering (CPF) approaches, each node uses a local PF to estimate a local posterior using its measurement(s), and its estimate is then transmitted to a Fusion Center (FC). The FC then computes the global posterior and the global state estimate. FC-based PFs are useful when the final estimate needs to be available only at a single central location; however, in the proposed method, the final estimate is available at every node at any time during the fusion process.

Communicating particles among neighboring nodes is costly; some works in the literature propose communicating certain selected particles over to the neighbouring nodes, such as the ones with the highest weights [121] in order to reduce the computational cost. However, the process of selecting the particles to transfer might be challenging and reduce the ability to communicate sufficient information through the network. Alternatively, one can approximate the local particle cloud at each sensor node with a single or a mixture of Gaussians. This approach has the advantage of significantly reducing the communication cost involved in exchanging local estimates among nodes, which is required by distributed data aggregation schemes.

Through iterative exchange of Gaussian representation of their local particle cloud, obtained using the SDCMMPF [104] method at each time, according to the weighted CP protocol, each node obtains an approximate of the global aggregated Gaussian. The aggregation weight assigned to each node during weighted CP is computed iteratively and reflects its estimated source reliability. Finally, the distance between the local particles and the aggregated global estimate is calculated, and is then used by a likelihood function to assign higher weights to the particles that are closer to the global Gaussian, and vice versa [153]. This re-weighting algorithm is followed by a resampling step and is intended to cause local particle clouds of sensor nodes to converge towards a global cloud. Algorithm 4 provides an overview of steps involved in the proposed approach, in which $S$ is the total number of nodes in the network. The following sections present a more detailed discussion of the aforementioned steps.

Table 5.1: Algorithm 4: Source-Reliability-Adaptive Distributed SDCMMPF

---

For sensor node s=1:S

**Step 1:** Perform SDCMMPF [104]

**Step 2:** Fit Gaussian to the local particle cloud [160], [137]

$E_t^{local(s)} = \frac{1}{N}\sum_{n=1}^{N} w_s^n x_s^n$

$\overline{\sum}_t^s = \frac{1}{N}\sum_{n=1}^{N} w_s^n (E_t^{local(s)} - x_s^n)(E_t^{local(s)} - x_s^n)^\top$

**Step 3:** For r=1:R (Apply weighted CP iteratively to obtain $E_t^{global(s)}$)

**Step 3.1:** Local message exchange between nodes $i$ and $j$

$E_r^{global(j)} = \frac{c_r(j)E_t^{local(j)} + \sum_{i\in N(j)} M_r^{ij} K_r^{ij}}{1+\sum_{i\in N(j)} K_r^{ij}}$

$M_r^{ij} = \frac{c_{r-1}(i)\times E_t^{local(i)} + \sum_{l\in N(i)\setminus j} M_{r-1}^{li} K_{r-1}^{li}}{1+\sum_{l\in N(i)\setminus j} K_{r-1}^{li}}$

$K_r^{ij} = \frac{c_{r-1}(i) + \sum_{l\in N(i)\setminus j} K_{r-1}^{li}}{1+\frac{1}{\gamma}(c_{r-1}(i) + \sum_{l\in N(i)\setminus j} K_{r-1}^{li})}$

**Step 3.2:** Compute source-reliability coefficient

$\delta_r(s) = E_r^{global(s)} - E_t^{local(s)}$

$c_r(s) = 1 - \frac{\delta_r(s)}{\delta_r max}$

**Step 4:** For $n = 1 : N$ (reweight local particle cloud w.r.t. $E_t^{global(s)}$ ) [153]

$d_s^n = \left\| x_s^n - E_t^{global(s)} \right\|$

$\mathcal{L}(x_s^n) = p(x_s^n | E_t^{global(s)}) = e^{\{\frac{-d_s^n}{-d_s^n+\beta}\}}$

$\alpha_s^n = \mathcal{L}(x_s^n)$

$\{w_i^n = w_i^n \times \alpha_i^n\}_{n=1}^N$

Resampling

**Step 5:** Go to step 1

### 5.2.2 Distributed Aggregation Using Weighted Consensus Propagation

After the global estimate of each node is approximated, using its local estimate and that of its neighbouring nodes, the difference between the node's local estimation and the global estimation of that node becomes the factors being used to infer its source reliability coefficient, i.e., aggregation weight. If this difference is high, the reliability of this node should be low, and vise versa. The weights are then deployed in the next weighted consensus prorogation iteration to account for the reliability of the nodes. In general, in CP protocol [8], if a node communicates to a neighbouring node at time $t$, it transmits a message consisting of $M^{li}$ and $K^{li}$, which denote the values associated with the most recently transmitted message from $l$ to $i$, at or before time $t$, and the number of nodes contributing to that estimation process, respectively.

At each time $t$, node $i$ has stored, in its memory, the most recent message from each neighbor: $< M^{li}, K^{li} >, l \in N(i)$. $N(i)$ denotes all neighbouring nodes of node $i$. If at time $t$, node $i$ chooses to communicate with a neighboring node $j$, it constructs a new message that is a function of the set of most recent messages $< M^{li}, K^{li} >, l \in N(i) \setminus j$ received from all neighbors other than $j$. From a distributed aggregation standpoint, $M^{ij}$ represents the average of the observations from the nodes that are within a certain distance from node $i$, and $K^{ij}$ denotes the cardinality of this set, i.e., the number of nodes involved in this estimation. The incoming neighboring node messages from previous round, along with the current local estimate $E_t^{local(j)}$, are used at each sensor node $j$ to iteratively update its global estimate $E_r^{global(j)}$ of the desired aggregated parameter at round $r$ as follows:

$$E_r^{global(j)} = \frac{c_r(j) E_t^{local(j)} + \sum_{i \in N(j)} M_r^{ij} K_r^{ij}}{1 + \sum_{i \in N(j) K_r^{ij}}} \tag{5.1}$$

where $c_r(j)$ denotes the source-reliability coefficient of node $j$ at round $r$, and $M_r^{ij}$ and $K_r^{ij}$ are calculated as follows:

$$M_r^{ij} = \frac{c_{r-1}(i) \times E_t^{local(i)} + \sum_{l \in N(i) \setminus j} M_{r-1}^{li} K_{r-1}^{li}}{1 + \sum_{l \in N(i) \setminus j} K_{r-1}^{li}} \tag{5.2}$$

$$K_r^{ij} = \frac{c_{r-1}(i) + \sum_{l \in N(i) \setminus j} K_{r-1}^{li}}{1 + \frac{1}{\gamma}(c_{r-1}(i) + \sum_{l \in N(i) \setminus j} K_{r-1}^{li})} \tag{5.3}$$

The parameter $\gamma \geq 0$ in 5.3 is a constant used to control the attenuation level of the CP. The intuition behind this attenuation process is to avoid the unbounded growth of $K$ in sensor networks with cycles. It is easy to see that the larger the $K$ and smaller the $\gamma$, the stronger the attenuation process would be. The convergence properties of this approach are proven and discussed in detail in [8]. In our experiments CP has been performed multiple times between measurement arrivals to reach consensus and obtain a global estimate for all sensor nodes.

At each round of CP, the distance between the local estimation of node $s$, i.e., $E_t^{local(s)}$, and its most recent global estimate $E_r^{global(s)}$, which is derived from sharing the estimations with its neighbouring nodes is used to calculate each node's source-reliability coefficient $c_r(s)$ as follows:

$$\delta_r(s) = E_r^{global(s)} - E_t^{local(s)} \tag{5.4}$$

$$c_r(s) = 1 - \frac{\delta_r(s)}{\delta_r(max)} \tag{5.5}$$

where $\delta_r(max)$ denotes the maximum possible difference between the local and global estimate of the nodes.

## 5.3 Experiments for Evaluating the Source Reliability in Information Fusion

In this section, a set of experiments are performed in order to evaluate the robustness of the proposed method with respect to permanent noise. The baseline case in which the sources are considered to be equally reliable, i.e., regular CP, is compared with the case in which one of the nodes in the sensor network is noisy, either due to a faulty hard sensor or a misleading soft data being reported, and the proposed adaptation scheme (weighted CP) is deployed. The performance metric is the mean squared error (MSE) of target trajectory obtained for the noisy sensor node in each case. Moreover, the average mean squared error (AMSE) of target trajectories obtained for all sensor nodes in the network along with its variance (uncertainty) are used to measure the overall convergence performance of the proposed source-reliability-adaptive distributed tracking approach.

### 5.3.1 Category I: Robustness to Noisy Hard Sensory Data

The purpose of this category of experiments is to examine the effect of the permanent noise added to the data provided by a hard sensor of one of the sensor nodes in the network. The target is tasked to be agile and all sensor nodes are being reported a valid soft data. All hard sensor data are initially noise free. However, starting from iteration 120 a white Gaussian noise is permanently added to the hard sensor of one of the sensor nodes in the network.

In Figure 5.1, the tracking performance obtained for the noisy sensor node is compared over time using both the baseline method, i.e., CP and the proposed method ,i.e., weighted CP (WCP). As shown, using WCP the effect of the noisy hard sensory data is alleviated, since the weights of its neighbouring sensor nodes are higher and its weight is lower, its estimation gets closer to that of its neighbouring nodes. Furthermore, comparing the network convergence performance results obtained using the CP and WCP are shown in Figures 5.1 and 5.2, respectively. Figure 5.1 shows the RMSE and uncertainty of CP and Weighte CP for the noisy sensor node and Figure 5.2 shows ARMSE and uncertainty of WCP for the entire sensor network. The WCP is yielding a fairly superior performance, in particular, in terms of lower variance (uncertainty) levels.

Using regular CP, all nodes share their information with their neighbouring nodes and consequently with all nodes in the networks with same weight and it yields the overall network performance to decrease as shown in Figure 5.1. However, when source reliability is accounted for, the weight of the noisy node and its affected neighbours is decreased for sharing information; therefore, the overall performance of the network is increased as shown in Figure 5.2.

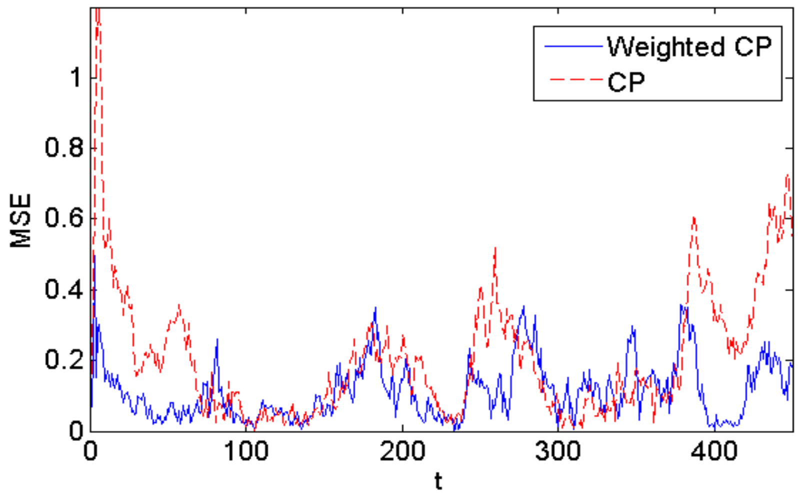


Figure 5.1: Target tracking performance comparison for sensor node with noisy hard sensory data

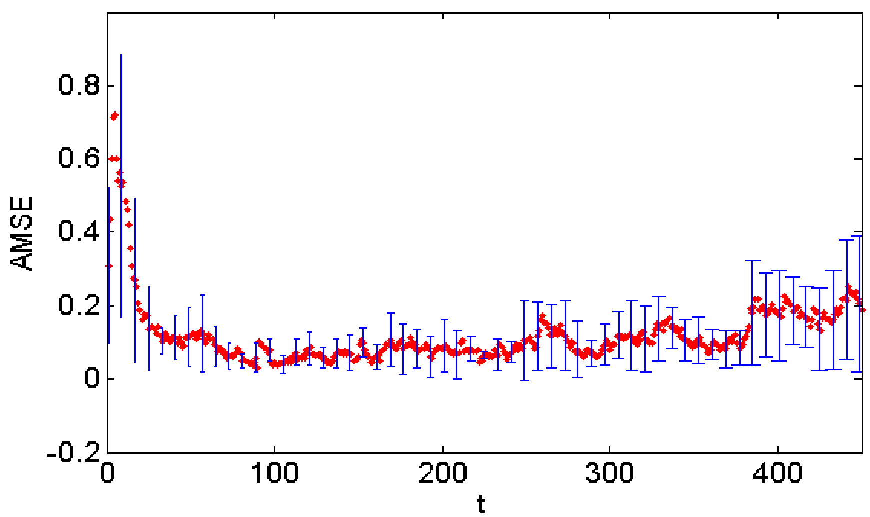


Figure 5.2: ARMSE and uncertainty of CP for entire sensor network with noisy hard sensory data

### 5.3.2 Category II: Robustness to Invalid Soft Data

Similar to the former category of experiments, the target is tasked to be agile and all sensor nodes are initially provided with valid soft and hard data. Starting from iteration 300 onwards, one of the sensor nodes in the network starts receiving an invalid soft data report.

As shown in Figure 5.3, using weighted CP the inflicted sensor node experiences an initial drop in performance, which then gradually improves and becomes stable eventually. The regular CP, however, is unable to deal with invalid soft data provided and ends up completely diverging after a period of time. As shown, using WCP method the effect of the misleading soft data is alleviated, since the weights of the neighbouring sensor nodes are higher and the weight of faulty sensor node is lower; therefore, its estimation gets closer to that of its neighbouring nodes. Moreover, from the entire network convergence perspective, WCP yields a more superior performance both in terms of overall accuracy of target trajectories and their corresponding uncertainties.

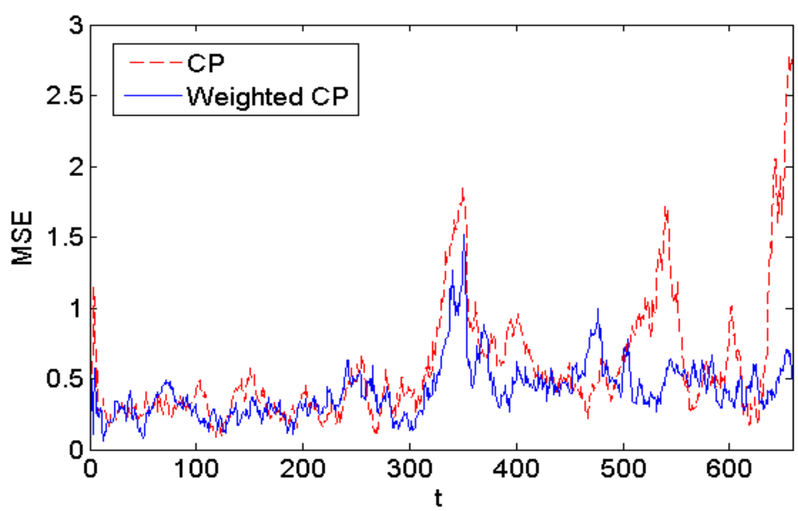


Figure 5.3: Target tracking performance comparison for sensor node with invalid soft sensory data

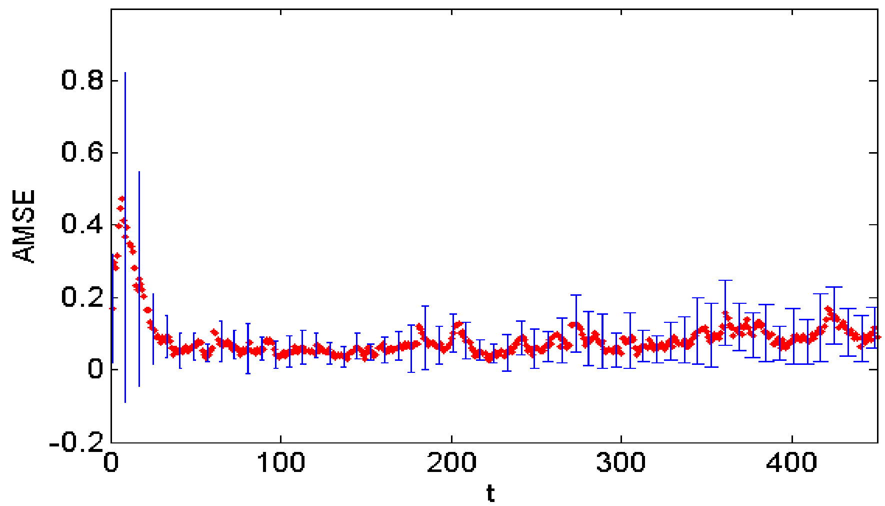


Figure 5.4: ARMSE and uncertainty of WCP for the entire sensor network with noisy hard sensory data

## 5.4 Summary

An adaptive source reliability technique was proposed in this chapter which considers each source's reliability in sharing the information, and it adaptively changes the reliability coefficients during the process. Using this approach as shown in the experimental results, if a sensor node is faulty, due to misleading soft data reported or a noisy hard data, its contribution in information sharing is reduced to alleviate its effect and it can also correct its estimation over time using the information from its neighbouring nodes. So far in this thesis, the problem of single target tracking was considered. In order to study the cases involving multi agile target tracking scenarios, a new approach has been proposed using the same concept of soft data integration into the Probability Hypothesis Density (PHD) filter.

# Chapter 6

# Soft-Data-Constrained Probability Hypothesis Density Filtering

## 6.1 Introduction

Numerous methods have been proposed to tackle the problem dealing with multiple targets. This thesis considers a problem in which the dynamics of maneuvering multiple-targets might deviate from the probabilistic characterization represented by a Transition Probability Matrix, i.e., multiple agile target tracking scenarios. In the following section, following an overview of the PHD filter in section 6.2, the SMC-based implementation of the interactive multiple model PHD filter is detailed and discussed in section 6.2.1, followed by the proposed SDC IMM-PHD filter presented in section 6.2.2.

## 6.2 PHD Filter

The Probability Hypothesis Density (PHD) filter proposed by Mahler [10] is a well-known multi-target tracking approach. Sequential Monte Carlo Implementation of the PHD Filter for Multi-target Tracking is proposed by Vo et al. [155] It relies on propagation of a first-order statistical moment of the multi-target posterior derived using the random set theory. The PHD filter can be implemented via the Gaussian Mixtures (GM) [63] or the Sequential Monte Carlo techniques [64]. SMC approaches have the advantage of computational tractability [156] and provable convergence properties [64], [157]. In addition, there is no need for the assumptions to be made on the form of the underlying probability density; therefore, they are applicable under the most general circumstances.

The SMC approximation of the IMM PHD filter is applicable to track multiple maneuvering targets with nonlinear, non-Gaussian dynamics. In particular, the SMC-PHD filter [158] has been extended using the interacting multiple-model principle (IMM SMC-PHD) to enable tracking of multiple maneuvering targets[159]. The IMM SMC-PHD is shown to diverge when applied to highly agile targets, whereas, the proposed soft-data-constrained variant is capable of tracking highly agile targets when provided with appropriate soft data.

### 6.2.1 IMM SMC-PHD Filter

Algorithm 5 shows the steps of the IMM SMC-PHD filter. As shown in the first step, there is an initialization of an augmented particle set $[\{x_t^n, w_t^n\}_{n=1}^N]$; in which, each particle consists of a state $x^n$, weight $w^n$ and a mode $r^n$, with $N$ being the total number of particles. After the particles' mode is predicted as shown in *step 2.1*; it is followed by a mode-dependant state prediction of the targets. For the target with state $x_{t-1}$ at time step $t-1$, the probability that it will survive at time $t$ is given by $e_{t|t-1}(x_{t-1})$. The prediction step is defined in *step 2.2*, where the Density $D_{t|t-1}(.)$ is similar to probability density except that it does not integrate to unity, $\delta(.)$ is the Dirac Delta function , and $w_{t-1}$ is the weight of the $n^{th}$ particle at time $t-1$. The function $f_{t|t-1}(.)$ in this equation characterizes the Markov target transition density.

Table 6.1: Algorithm 5: IMM SMC-PHD

$[\{x_t^n\}_{n=1}^N]$ =IMM SMC-PHD $[\{x_{t-1}^n\}_{n=1}^N, z_t]$

**Step 1:** Initialization: $\{x_t^n, r_t^n, w_t^n\}_{n=1}^N$

**Step 2:** Prediction

**Step 2.1:** Mode prediction

$p(r_t|z_{1:t-1})$

$= \sum_{m,m' \in \mathbb{N}} \sum_{n=1}^{N_t^P} h_{mm'}(x_{t-1}^n) w_{t-1}^n \delta(m - r_{t-1}^n)$

**Step 2.2:** Mode-dependant state prediction

$D_{t|t-1}(x_t, r_t|z_{1:t-1})$

$= \sum_{n=1}^{N_t^P} w_{t|t-1}^n \delta(x_t - x_{t|t-1}^n, r_t - r_{t|t-1}^n)$

$w_{t|t-1}^n = e_{t|t-1}(x_{t|t-1}^n) f_{t|t-1}(x_{t|t-1}^n | x_{t-1}^n, r_{t|t-1}^n)$

**Step 3:** Correction (Updating)

$w_t^n = (1 - P_D(x_{t|t-1}^n)) + \sum_{i=1}^{N_t^Z} \frac{P_D(x_{t|t-1}^n) f_{t|t}(z_t^i | x_{t|t-1}^n, r_{t|t-1}^n)}{\lambda_t c_t(z_t^i) + \psi_t(z_t^i)}$

with the likelihood function:

$\psi_t(z_t^i) = \sum_{n=1}^{N_t^P} P_D(x_{t|t-1}^n) f_{t|t}(z_t^i | x_{t|t-1}^n, r_{t|t-1}^n) w_{t|t-1}^n$

**Step 4:** Evaluate number of targets

$\hat{T}_t = \sum_{n=1}^{N_t^P} w_t^n$

**Step 5:** Grouping & clustering estimations

**Step 6:** Go to step 2

The predicted PHD can be corrected with the availability of measurements $z_{1:t}$ at time step $t$ to get the updated PHD. We assume that the number of false alarms is Poisson distributed with the average rate of $\lambda_t$, and that the probability density of the spatial distribution of false alarms is $c_t(z_t)$. Let the detection probability of a target with state $x_t$ at time step $t$ be $P_D(x_t)$, updating or correction step based on measurement data is defined in *step 3*, where $N_t^Z$ indicates the number of measurements at time $t$. The single-target/single-sensor measurement likelihood function is defined by $f_{t|t}(.)$ in this equation.

In contrast to the particle filter, in PHD filter the summation of the particles' weights is not equal to one, rather it is equal to the total number of targets at that moment. In other words, the expected number of targets at time step $t$ is the summation of the weights of all the particles at that moment. In *step 4*, the total number of targets is estimated, where $\hat{T}_t$ and $N_t^p$ indicate the number of estimated targets and the number of particles at time $t$, respectively. In the next step, particles are clustered to provide final targets' estimations.

### 6.2.2 Soft-Data-Constrained IMM PHD Filter

This section presents the proposed soft-data-constrained IMM SMC-PHD filter used to tackle the problem of agile multi-target tracking. The filter uses a Fuzzy logic approach [102, 29] to model and incorporate the soft data.

#### Soft Data Modeling Using Fuzzy Logic

In the mode prediction step, early in SDC IMM-SMC-PHD method, the same number of particles ($\frac{N}{M}$) is transferred to each mode, where $N$ is the total number of particles and $M$ is the total number of modes (step 4.1 in Algorithm 6). For particles with their current mode defined by $m$, the next mode ($m'$) is predicted using the TPM. That is, respective value is extracted from the transition matrix, $\pi_{mm'}$. The value of $\pi_{mm'}$ is also assigned to a variable called the Stochastic Agility Discount (SAD). For each particle, the next mode is also predicted using the IMM SMC-PHD filter. Then the prediction step is performed using the predicted mode for each case, and the difference of the resulting clouds is evaluated using the Optimal Sub-Pattern Assignment (OSPA) [161], in order to compare the distance of these two density clouds (Figure 6.1). The OSPA is defined as follows:

$$\overline{d}_p^{(c)}(X,Y) = (\frac{1}{\beta}(min_{\pi\in\prod_\beta}\sum_{i=1}^{\alpha} d^{(c)}(x_i, y_{\pi(i)}))^p + c^p(\beta-\alpha))^{\frac{1}{p}} \tag{6.1}$$

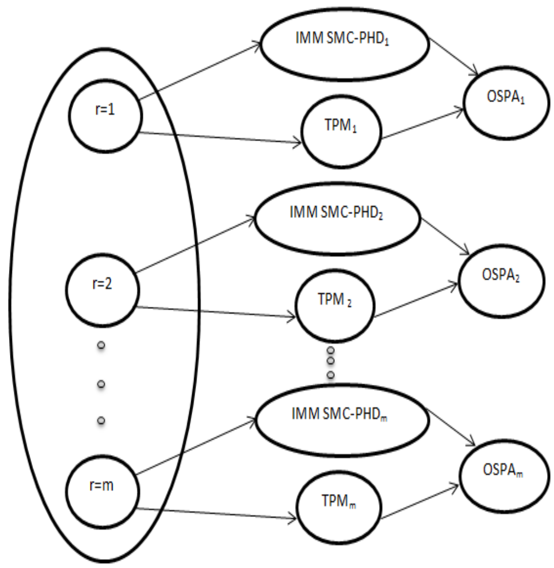


Figure 6.1: OSPA distance calculation procedure including predicting clusters using both IMM SMC-PHD & TPM followed by computing the distance between two predicted densities

where $X = \{x_1, ..., x_\alpha\}$ and $Y = \{y_1, ...y_\beta\}$ are finite subsets, $\alpha$ and $\beta \in N_o = \{0, 1, 2, ...\}$, $1 \leq p < \infty$ and $c > 0$; in our simulations, $p = 1$ and $c = 50$. The output of this step is called the Expected Cluster Weight (ECW), and shows the estimated target's agility with respect to the maneuvering characteristics defined by the TPM (step 4.2 in Algorithm 6).

After choosing the FIS, the value $ECW$, which represents the divergency of the target behaviour with respect to the TPM, along with the $SAD$ are the inputs to the FIS. The output of the FIS is the set of constraints $[\{C_t\}_{1=n}^{N}]$ used to re-weight the particles (step 5 in Algorithm 6), in order to incorporate the external knowledge in the estimation process (step 6 in Algorithm 6).

**Incorporating Soft Data as Dynamic Constraints**

As discussed in the previous section, the FIS is modeled based on the inputs RAL and RCL. After that, divergency of the target behaviour with respect to the TPM is evaluated ($ECW$) and is the first input to the FIS. For each particle, based on its previous mode ($m$) and its predicted mode ($m'$), the respective value of TPM is selected ($SAD$) and is the other input to the FIS. Then, constraints are calculated based on the Fuzzy rules and are incorporated into the particles' weights.

As discussed earlier in Chapter 3, Figures 3.3 to 3.6 show the effect of the inputs on the Fuzzy inference output for two different Fuzzy models selected based on the soft data report. These two figures are deployed to show how variation in the soft data report affects the constraints produced by FIS. To make the figures clearer and to briefly explain how the Fuzzy rules are modeled to infer the constraints based on the report, some of the cases are explained as follows:

Let us consider the case in which the target is agile; therefore, the $ECW$ is high, since the distance between the density predicted by IMM SMC-PHD filter and TPM is large. If the report is "target is certainly extremely agile", the FIS shown in Figure 3.3 is selected for the inference process based on the RAL="extremely" and RCL="certainly". If the $SAD$ is low for the $n^{th}$ particle, then the constraint coefficient applied to the weight of the respective particle should be very high, Table I and Figure 3.2 depict the same situation, and vice versa. That is, for particles with a high SAD value, indicating that the particle is behaving based on the behaviour characterized by TPM, the constraint should be low to decrease the weight of the respective particle, since in reality the target is agile and its trajectory is not based on the behaviour characterized by TPM. Based on the similarity of these two estimations calculated by TPM and IMM SMC-PHD, and the agility reported, the constraints are evaluated and are then applied to the respective particles. If the report indicates the existence of agility and the target is agile, i.e., it does not behave in a similar fashion to the TPM, the particles in dominant mode should get lower weights and the rest of the particles should be assigned higher weights in order to survive and to re-generate more.

After incorporating the constraints into particles' weights, a resampling step is performed. If the target's agility level, which is input by the user, is high and the target is not agile, then the particles that follow the behavior defined by TPM should get low weights to gradually disappear. On the other hand, the rest of the particles should get higher weights in order to survive.

The soft-data-inspired dynamic constraints affect the particles' weights before the resampling step; therefore, the weighting of the particles is updated as follows:

$$w_t^n = w_{t-1}^n p(z_t | x_t, r = m') C_t^n \tag{6.2}$$

Table 6.2: Algorithm 6: SDC IMM-SMC-PHD

$[\{x_t^n, C_t^n\}_{n=1}^N]$ =SDC IMM-SMC-PHD $[\{x_{t-1}\}_{n=1}^N, z_t, SD]$

**Step 1:** Define a set of FIS

**Step 1.1:** Define rules based on RAL

**Step 1.2:** Define membership functions based on RCL

**Step 2:** Interpret SD: {RCL & RAL}

Selecet FIS based on given RCL & RAL

**Step 3:** Particle Initialization

**Step 4:** Mode Prediction

**Step 4.1:** Cluster Particle cloud into M particle clouds

$PC_m : m = 1, ...M$

**Step 4.2:** For $PC_m : 1, ..., M$ (Figure 6.1)

Predict next mode (m') using TPM
Predict m' using generic IMM PHD

ECW $\alpha$ Distance of the two clouds using OSPA
SAD $\alpha$ The respective element of TPM based on m and m'

**Step 5:** Compute constraints:

For n=1:N

$ECW\ \alpha\ OSPA$ & $SAD_t^n\ \alpha\ \pi_{mm'}$

$C_t^n = FIS(SAD_t^n, ECW_t)$

**Step 6:** Apply constraints to particles' weights

For n=1:N

$w_t^n = w_t^n \times C_t^n$

**Step 7:** Resampling

**Step 8:** Mode-dependant state prediction

**Step 9:** Correction (Updating)

**Step 10:** Evaluating number of targets

**Step 11:** Grouping & clustering the estimates

**Step 12:** Go to Step 4

$$C_t^n = FIS(SAD_t^n, ECW_t) \quad n = 1, ..., N \tag{6.3}$$

in which the constraints ($C_t^n$) are calculated in *step 5* of Algorithm 6. Algorithm 6 has many steps similar to those in Algorithm 5, and it also adds a number of additional steps to accomplish the mode prediction.

## 6.3 Multi-Target Tracking Experiments

### 6.3.1 Experiments

A two dimensional tracking example is used to compare the impact of the soft data in the case of agility in target dynamics. There are five targets that can appear and disappear successively, with initial positions of $(3 \times 10^2, 4 \times 10^2)m, (4 \times 10^2, 3 \times 10^2)m, (6 \times 10^2, 8 \times 10^2)m, (6 \times 10^2, 10 \times 10^2)m$, and $(7 \times 10^2, 5 \times 10^2)m$. Figures 6.2 and 6.3 show target trajectories with no agility and high agility, respectively. There are three modes: a constant velocity model and two coordinated turn models. The Markovian transition probability matrix indicating the transition probability between different modes is shown below:

$$[h_{mm'}] = \begin{bmatrix} 0.1 & 0.45 & 0.45 \\ 0.7 & 0.1 & 0.2 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \tag{6.4}$$

The TPM represents the state transition probability from the $m^{th}$ mode to the $m'^{th}$ mode. Constant velocity and coordinated turn models are described as follows, respectively:

$$x_t = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} + x_{t-1} + \sigma_t \tag{6.5}$$

$$x_t = \begin{bmatrix} 1 & sin(\frac{\Omega_{t-1}T}{\Omega_{t-1}}) & 0 & -\frac{1-cos(\Omega_{t-1}T)}{\Omega_{t-1}} \\ 0 & cos(\Omega_{t-1}T) & 0 & 1-sin(\Omega_{t-1}T) \\ 0 & \frac{1-cos(\Omega_{t-1}T)}{\Omega_{t-1}} & 1 & \frac{sin(\Omega_{t-1}T)}{\Omega_{t-1}} \\ 0 & sin(\Omega_{t-1}T) & 0 & cos(\Omega_{t-1}T) \end{bmatrix} + x_{t-1} + \sigma_t \tag{6.6}$$

where $\Omega_t$ is the turning rate at time step $t$, and $T$, which is the sample time, is equal to one. $\sigma$ is an i.i.d sequence of zero-mean Gaussian vectors with a covariance $Q$.

$$Q = \begin{bmatrix} \frac{T^4}{4} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} q \tag{6.7}$$

The level of the power spectral density of the corresponding continuous process noise ($q$) is equal to $1 \times 10^{-3}$. Performance evaluation of multi-target tracking algorithms is of great practical importance in the design and comparison of tracking systems. In order to evaluate the performance of the proposed method, a consistent metric, recently proposed, called OSPA, is used as defined in the Section 6.2.2.

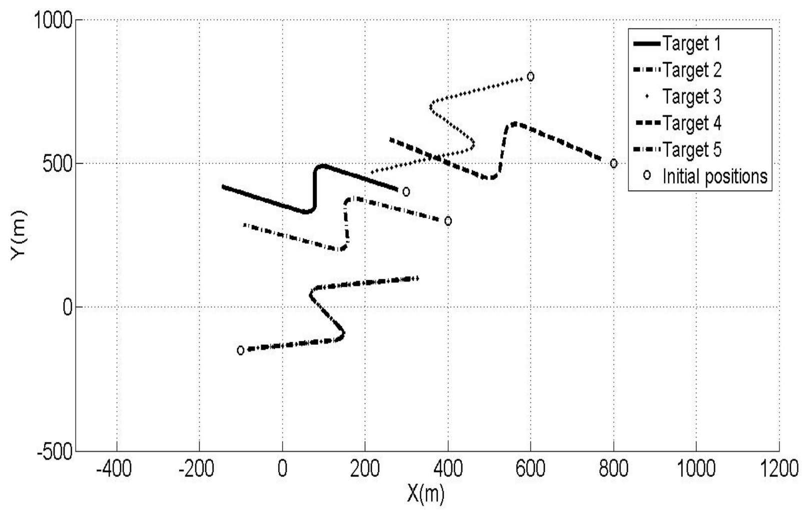


Figure 6.2: Targets trajectory without agility

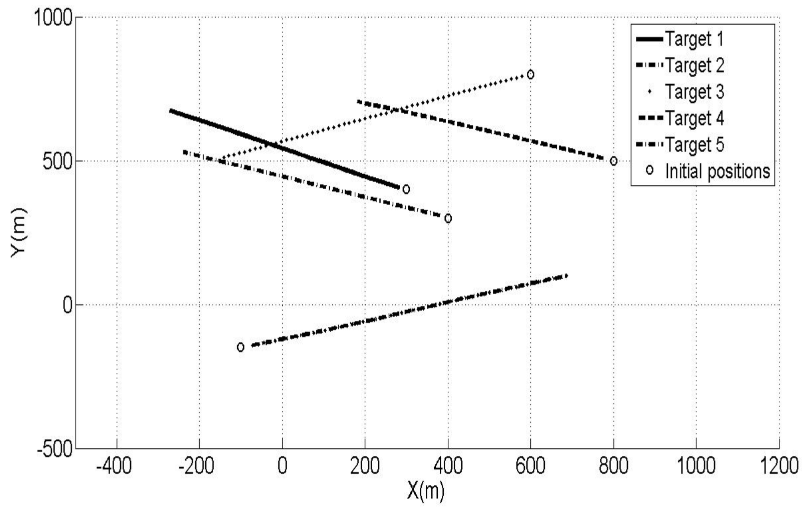

Figure 6.3: Targets trajectory with agility

### 6.3.2 Category I: Impact of Incorporating Soft Data

In this scenario, targets are highly agile, i.e, targets are expected to make turns based on TMP; however, they travel only in a straight line during the simulation. Figure 6.4 demonstrates a comparison of the OSPA for the case with no soft data provided, i.e., the generic IMM SMC-PHD filter, and the proposed SDC IMM-SMC-PHD filter with the soft-data report "target is certainly extremely agile".

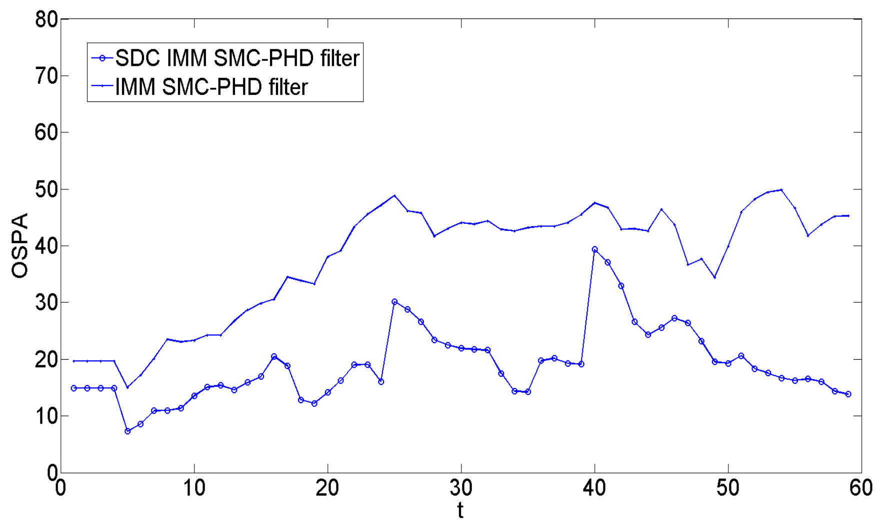


Figure 6.4: Soft-data effect in case of agility

As shown in Figure 6.4, the proposed method, SDC IMM-SMC-PHD filter, has less OSPA distance during the simulation time, which shows more accurate tracking performance. It is clear that when the targets do not switch their modes, there are no obvious differences between them; however, when maneuvers occur, the OSPA distances increase, i.e. at simulation times $t = 25$ and $t = 40$. This result occurs because when the conditional model probabilities and switching rates have small values, there may be very few particles for one or more models in the IMMSMC-PHD filter, especially if there is agility in the target dynamics. Then the empirical density spanned by all particles with such a mode does not perform an accurate approximation of the corresponding exact conditional density. Such problems have been solved by the proposed algorithm, since the exact conditional density is approximated by incorporating the external knowledge.

### 6.3.3 Category II: Impact of Soft Data Certainty Level

In this set of experiments, the effect of the soft data's certainty level of the soft data is examined and compared (Figure 6.5), for a case in which the target is agile and the reports are "target is certainly highly agile" and "target is perhaps highly agile". In both cases, the reported soft data provides a correct information regarding the targets' agility level; however, the certainty levels of the reports are different. The effect of the constraints on the particles' weights and therefore the filters' performance can be observed in this figure.

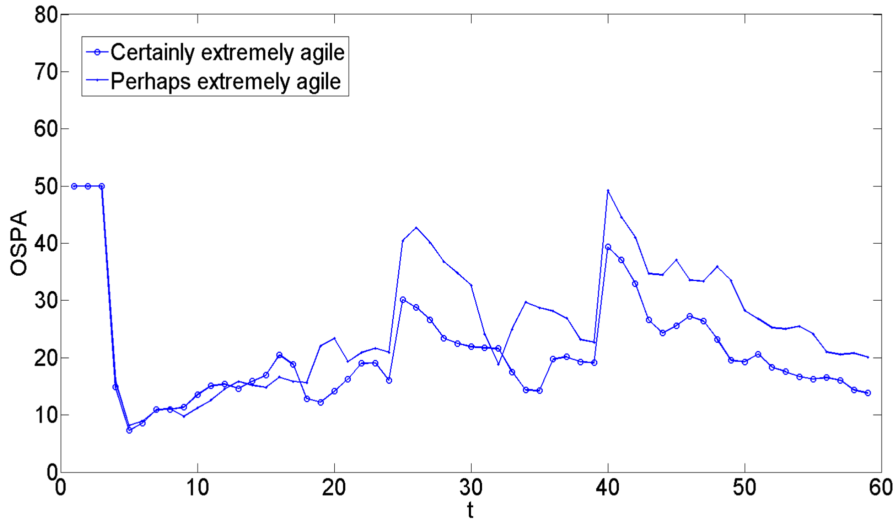


Figure 6.5: Impact of the soft data report's certainty level

As shown, when correct soft data is reported regarding the agility level of the target, the report with the higher certainty provides better approximation, i.e., a lower OSPA distances are observed. When the certainty level decreases, since the constraints are not that effective anymore, they have less effect on the particles' weights and therefore the approximation is not as accurate.

### 6.3.4 Category III: Impact of Number of Particles

Different number of particles are used in order to evaluate the resulting effects. As shown in Table 6.3, in SMC-based methods, the number of particles used is very important. The accuracy of the approximation is directly proportional to the size of the particle set ($N$); increasing the total number of particles increases the accuracy of the approximation, but also increases the computational cost. In other words, choosing the number of particles is a trade-off between the accuracy and the computational resources.

Table 6.3: Impact of number of particles

| Number of Particles | Average simulation time(s) | Average OSPA distance(m) |
|---|---|---|
| 100 | 10 | 40.01 |
| 500 | 70 | 30.84 |
| 1000 | 180 | 23.59 |

## 6.4 Summary

This chapter has presented PHD filter, which is used for tracking multiple targets, followed by a discussion of an interactive multiple model variant of it. The technique adopted for modeling soft data and a way to incorporate it into the the filtering process have been presented earlier in section 3.2 of this thesis in details, and the procedure used to model soft data using Fuzzy logic is demonstrated. The traditional methods cannot handle tracking when there is an agility in target dynamics; however, the proposed method has shown to be robust in the experimental results. The tests evaluate the effect of incorporating soft data on the estimation performance as well as the effect of the soft data report's certainty level. In the next chapter, concluding remarks along with some possible future directions for research as the extension of this work are presented.

# Chapter 7

# Concluding Remarks and Future Work

## 7.1 Concluding Remarks

The proposed framework can be used for implementing any intelligent system which aims at automated fusion of high level information into its estimation process. Soft data which is provided via pictures, videos, text, Podcasts, and other media provide a unique opportunity for the data fusion community. Some of the potential applications of incorporating soft data include crisis management, understanding and addressing natural disasters, and other situation assessments [1].

This thesis has proposed a Soft-Data-Constrained Multi-Model Particle Filtering (SDC MMPF) method, in which inherently vague soft data provided by human agents are properly modeled using a Fuzzy inference system. These data are then transformed into a set of constraints and imposed on the MMPF method, enabling it to deal with tracking situations involving potentially highly agile targets. The experiments conducted for the task of single agile target tracking demonstrate the proposed approach's efficiency in enhancing the MMPF method's ability to deal with target agility. The method meets this goal by incorporating the agility level reported as soft data into the tracking process as dynamic constraints. In particular, the conventional MMPF method is shown to perform poorly; that is, it diverges when applied to highly agile targets. However, the proposed method is capable of tracking highly agile targets when provided with appropriate soft data.

Furthermore distributed framework for the proposed method is proposed. As demonstrated by the experimental results, the proposed method has the ability to recover from failure situations and is robust to noise, since it adjusts the same population of particles, instead of regenerating them. It uses the aggregated global estimate to infer the constraints, which are then applied to the particles' weights in order to adjust them appropriately. In contrast, in most of the existing methodologies, each node directly samples its particle population from an aggregated global estimate. In the proposed method, if a noise occurs in the system, it has less effect on the noisy node and its neighbouring nodes and consequently on the overall network, unlike in the other approaches, and can recover faster. As a result, the nodes do not corrupt their estimations upon a sudden failure in the system and experience gradual change in case of a failure. Moreover, the proposed method is more computationally efficient, since using the same number of particles it can yield lower ARMSE. Lastly, when provided with incorrect soft data, the proposed method is able to avoid divergence and maintain fairly acceptable performance.

In order to account for sources' reliability in sharing their information, a Weighted CP (WCP) is proposed; in which, during each round of CP, the distance between the local estimation of a node and its most recent global estimate, which is derived from sharing the estimations with its neighbouring nodes, is used to calculate each node's reliability coefficient. This step enables Consensus propagation method to incorporate the sources' reliability as a set of weights into the consensus process. The case of maneuvering multi-target tracking is also considered, wherein multiple target maneuvers may deviate from their stochastic characterization represented by the transition probability matrix.

## 7.2 Future Research Directions

In the following, several interesting directions for future work are presented:

- The soft data ontology could be extended to enable human agents to supply more reports regarding the target dynamics such as the appearance/disappearance of a target, as well as merging, and spawning, also reports can provide information on other aspects of the targets.
- More experiments can be performed with larger number of sensor nodes to further evaluate the efficiency of the proposed distributed data aggregation scheme in terms of its scalability.
- The current experiments are performed in simulation environment only. In order to observe the efficiency of the system, it should be implemented in a real world testbed environment.

- The Random Set theory has been shown to be capable of representing first order, second order and even composite rules [162]. It can be used to model human data provided in form of rule-based logical statements as well as data available on the Web and incorporate them into the fusion process.

- While performing high-level information fusion, it is usually required to have access to some high level information. There are some challenges involved in use of this high-level information, for example, learning how to treat humans and search engines as sensors and task them, especially in crisis situations [1]. This research area provides rich opportunities for both theoretical development and practical demonstrations of these new resources.

# Bibliography


[1] David L. Hall , Michael Mcneese , James Lunas , Tracy Mullen, "A Framework for Dynamic Hard/Soft Fusion," *In Proceedings of the 11th International Conference on Information Fusion (FUSION)*, 2008.

[2] J. C. Rimland, D. Coughlin, D. L. Hall, J. L. Graham, "Advances in data representation for hard/soft information fusion," *Multisensor, Multisource Information Fusion*, 2012.

[3] Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy and M. B. Srivastava, "Participatory Sensing," *Proceedings of WSW'06 at SenSys '06*, Oct 31, 2006.

[4] Edith Law, Luis von Ahn, "Human Computation," *SYNTHESIS LECTURES ON ARTIF ICIAL INTELLIGENCE AND MACHINE LEARNING*, 2011.

[5] B. Ristic, S. Arulampalam and N. Gordon, "Beyond the Kalman filter: Particle filters for tracking applications," *Artech House Publishers*, 2004.

[6] B. Khaleghi, F. Karray, "Distributed random set theoretic soft-hard data fusion: Target tracking application," *Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 455–460, 2011.

[7] B. Khaleghi, A. Khamis, F. Karray, "Random finite set theoretic based soft/hard data fusion with application for target tracking," *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 50–55, 2010.

[8] C. C. Moallemi, and B. Van Roy, "Consensus propagation," *IEEE Trans. Information Theory*, vol. 52, no. 11, pp.4753–4766, 2006.

[9] R. Mahler, "Statistical Multisource-Multi-target Information Fusion," *Artech House*, 2007.

[10] R. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[11] I. Bloch, "Information combination operators for data fusion: a comparative review with classiffication," *IEEE Trans. on SMC: Part A*, 26 (1) (1996) 52-67.

[12] D.L. Hall, J. Llinas, "An introduction to multisensor fusion," *Proc. of the IEEE*, 85 (1) (1997) 6-23.

[13] D. Smith, S. Singh, "Approaches to multisensor data fusion in target tracking: a survey," *IEEE Trans. on Knowledge and Data Engineering*, 18 (12) (2006) 1696-1710.

[14] A.A. Goshtasby, S. Nikolov, "Image fusion: advances in the state of the art," *Information Fusion*, 8 (2) (2007) 114-118.

[15] I. Corona, G. Giacinto, C. Mazzariello, F. Roli, C. Sansone, "Information fusion for computer security: state of the art and open issues," *Information Fusion*, 10 (4) (2009) 274-284.

[16] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Huubner, "Ontology-based integration of information - a survey of existing approaches," *in: Proc. of the IJCAI - Workshop on Ontologies and Information Sharing*, 2001, pp. 108-117.

[17] D. L. Hall and J. M. Jordan, "Human-Centered Information Fusion," *Boston, MA, USA: Artech House*, 2010.

[18] B. Dasarathy, "Decision Fusion Strategies in Multisensor Environments ", *IEEE Transactions on Systems, Man, and Cybernetics*, pp 1140-1154, vol. 21, 1991.

[19] M. Bedworth and J. Obrien, "The Omnibus Model: A New model of data fusion," *AES Magazine*, April 2000.

[20] I. Kadar, "Perceptual Reasoning in Adaptive Fusion Processing," *SPIE*, 2002.

[21] H. Bostroom, S.F. Andler, M. Brohede, R. Johansson, A. Karlsson, J. van Laere, L. Niklasson, M. Nilsson, A. Persson, T. Ziemke, "On the definition of information fusion as a field of research, Informatics Research Centre," *University of Skoovde, Tech. Rep. HS-IKI-TR-07-006*, 2007.

[22] E.L. Walts, “Data fusion for C3I: a tutorial,” *in Command, Control, Communications Intelligence, (C3I) Handbook*, EW Communications Inc., Palo Alto, CA, 1986, pp. 217-226.

[23] M. Kumar, D.P. Garg, R.A. Zachery, “A generalized approach for inconsistency detection in data fusion from multiple sensors,” *in: Proc. of the American Control Conference*, 2006, pp. 2078-2083.

[24] R.P.S. Mahler, “statistics 101 for multisensor, multitarget data fusion,” *IEEE Aerospace and Electronic Systems Magazine*, 19 (1) (2004) 53-64.

[25] Y. Zhu, E. Song, J. Zhou, Z. You, “Optimal dimensionality reduction of sensor data in multisensor estimation fusion,” *IEEE Trans. on Signal Processing*, 53 (5) (2005) 1631-1639.

[26] B.L. Milenova, M.M. Campos, “Mining high-dimensional data for information fusion: a database-centric approach,” *in Proc. of International Conference on Information Fusion*, 2005, pp. 638-645.

[27] H.F. Durrant-Whyte, T.C. Henderson, “Multisensor Data Fusion,” *in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics, Springer*, 2008, pp. 585-610.

[28] F.K.J. Sheridan, “A survey of techniques for inference under uncertainty,” *Artificial Intelligence Review*, 5 (1-2) (1991) 89-119.

[29] L.A. Zadeh, “Fuzzy sets,” *Information and Control*, 8 (3) (1965) 338-353.

[30] L.A. Zadeh, “Fuzzy sets as a basis for a theory of possibility,” *Fuzzy Sets and Systems*, 1 (1) (1978) 3-28.

[31] Z. Pawlak, “Rough Sets: Theoretical Aspects of Reasoning about Data,” *Kluwer Academic Publishers*, Norwell, MA, 1992.

[32] G. Shafer, “A Mathematical Theory of Evidence,” *Princeton University Press*, 1976.

[33] B. Khaleghi, A. Khamis, F. O. Karray, S. N. Razavi, “Multisensor data fusion: A review of the state-of-the-art,” *Information Fusion*, 2011.

[34] Hugh Durrant-Whyte, “Multisensor Data Fusion,” *Australian Centre for Field Robotics, The University of Sydney*, 2001.

[35] S.J. Julier, J.K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," *in: International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997, pp. 182-193.

[36] G. Welch, G. Bishop, "An introduction to the Kalman filter," *Department of Computer Science, University of North Carolina, Tech. Rep. TR-95-041*, 1995.

[37] A. Doucet, N. de Freitas, N. Gordon, "Sequential Monte Carlo Methods in Practice," *Statistics for Engineering and Information Science, Springer*, New York, 2001.

[38] B.A. Berg, "Markov Chain Monte Carlo Simulations and Their Statistical Analysis," *World Scientific*, Singapore, 2004.

[39] D. Crisan, A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing*, pp. 736–746, 50 (3) (2002).

[40] A. Doucet, N. de Freitas, K. Murphy, S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000.

[41] G. Kitagawa and W. Gersch, "Smoothness Priors Analysis of Time Series," *Lecture Notes in Statistics, Springer*, vol.116, 1996.

[42] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Journal of Statistics and Computing*, vol. 10, Issue 3, pp. 197–208, 2000.

[43] S. Thurn, W. Burgard, and D. Fox, "Probabilistic Robotics," *MIT Press*, 2005.

[44] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking. part V. multiple-model methods," *In Proc. SPIE Conf. on Signal and Data Processing of Small Targets*, pp. 559–581. 2003.

[45] X. R. Li, "Engineer's guide to variable-structure multiple-model estimation for tracking," *In Y. Bar-Shalom and W. D. Blair (Eds.), Multitarget-Multisensor Tracking: Applications and Advances, Vol. III, Boston, MA: Artech House*, ch. 10, pp. 499–567, 2000.

[46] D. G. Lainiotis, "Optimal adaptive estimation: Structure and parameter adaptation," *IEEE Transactions on Automatic Control*, 16(2), pp. 160–170, 1971.

[47] Y. Bar-Shalom, K. C. Chang, H. A. P. and Blom, "Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, 25(2), pp. 296–300.

[48] X. R. Li, and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic Control*, 41(4), pp. 478–493, 1996.

[49] J. Wang and Y. Yagi, "Adaptive Mean-Shift Tracking With Auxiliary Particles," *IEEE Trans. on Systems, Man, and Cybernetics: Cybernetics*, vol. 39, no. 6, pp. 1578–1589, 2009.

[50] Z. Hou, L. Cheng, and M. Tan, "Decentralized Robust Adaptive Control for the Multiagent System Consensus Problem Using Neural Networks," *IEEE Trans. on Systems, Man, and Cybernetics: Cybernetics*, vol. 39, no. 3, pp. 636–647, 2009.

[51] O. Straka, J. Dunk, M. Limandl, "Truncation nonlinear filters for state estimation with nonlinear inequality constraints," *Journal of Automatica*, vol. 48, no. 2, pp. 273–286, 2012.

[52] S. M.cGinnity and G. W. Irwin, "Multiple Model Bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerospace and Electronic systems*, vol. 36, Issue 3, pp. 1006–1012, 2000.

[53] D. S. Angelova, T. A. Semerdjiev, V. P. Jilkov and E. A. Semerdjiev, "Application of Monte Carlo Method for tracking maneuvering target in clutter," *The Second IMACS Seminar on Monte Carlo Methods: Mathematics and Computers in Simulation*, vol. 55, Issue 13, pp. 15-23, 2001.

[54] Fun-Bin Duh and Chin-Teng Lin, "Tracking a Maneuvering Target Using Neural Fuzzy Network," *IEEE Trans. Systems, Man, and Cybernetics: Cybernetics* vol. 34, no. 1, pp. 16–33, 2004.

[55] V. Kreinovich, "Random Sets Unify, Explain, and Aid Known Uncertainty Methods in Expert Systems," *in: J. Goutsias, R.P.S. Mahler, H.T. Nguyen (Eds.), Random Sets: Theory and Applications, Springer-Verlag*, 1997, pp. 321-345.

[56] S. Mori, "Random Sets in Data Fusion. Multi-Object State-Estimation as a Foundation of Data Fusion Theory," *in: J. Goutsias, R.P.S. Mahler, H.T. Nguyen (Eds.), Random Sets: Theory and Applications, Springer-Verlag*, 1997, pp. 185-207.

[57] I.R. Goodman, R.P.S. Mahler, H.T. Nguyen, "Mathematics of Data Fusion," *Kluwer Academic Publishers*, Norwell, MA, 1997.

[58] R.P.S. Mahler, "Random sets: unification and computation for information fusion - a retrospective assessment," *in Proc. of the International Conference on Information Fusion*, 2004, pp. 1-20.

[59] R.P.S. Mahler, "Statistical Multisource-Multitarget Information Fusion," *Artech House*, Boston, MA, 2007.

[60] R.P.S. Mahler, "PHD filters of higher order in target number," *IEEE Trans. on Aerospace and Electronic Systems*, 43 (4) (2007) 1523-1543.

[61] B.T. Vo, B.N. Vo, A. Cantoni, "The cardinalized probability hypothesis density filter for linear gaussian multi-target models," *in: Proc. of the Annual Conference on Information Sciences and Systems*, 2006, pp. 681-686.

[62] B.T. Vo, B.N. Vo, A. Cantoni, "Analytic implemenations of the cardinalized probability hypothesis density filter," *IEEE Trans. on Signal Processing*, 55 (7) (2007) 3553-3567.

[63] B.N. Vo, W.K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Trans. on Signal Processing*, 54 (11) (2006) 4091-4104.

[64] B. Vo, S. Singh, A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224-1245, 2005.

[65] L. Zuo, K. Mehrotra, P. K. Varshney, and C. K. Mohan, "Bandwidth-efficient target tracking in distributed sensor networks using particle filters," *in proc. IEEE Int. Conf. on Information Fusion*, pp. 1–4, 2006.

[66] M. Vemula, M. F. Bugallo, and P. M. Djuric, "Target Tracking in a Two-Tiered Hierarchical Sensor Network," *in proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[67] M. S. Mahmoud, M. Sabih, "An assessment of distributed state estimation," *Int. Journal of Systems, Control and Communications (IJSCC)*, vol. 5, no. 2, pp. 93–107, 2013.

[68] M. S. Mahmoud, "Distributed Control and Filtering for Industrial Systems," *IET Press*, UK, December 2012.

[69] M. S. Mahmoud, H. M. Khalid and M. Sabih, "Improved Distributed Estimation Method for Environmental Physical variables in Static Sensor Networks," *in proc. IEEE Wireless Sensor Systems, IET*, vol. 3, no. 3, pp. 216–232, 2013.

[70] M. S. Mahmoud and M. Sabih, "Experimental Investigations for Distributed Networked Control Systems," *IEEE Systems Journal*, doi: 10.1109/JSYST.2012.2228122, 2013.

[71] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Trans. Automatic Control*, vol. 33, no. 1, pp. 88-94, January 1988.

[72] L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. DurrantWhyte, "Consistent methods for decentralised data fusion using particle filters," *in proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 85-91, 2006.

[73] L. Ong, T. Bailey, H. Durrant-Whyte, and B. Upcroft, "Decentralised particle filtering for multiple target tracking in wireless sensor networks," *in proc. IEEE Int. Conf. on Information Fusion*, pp. 1–8, 2008.

[74] O. Hlinka, F. Hlawatsch, and P. M. Djuri'c "Distributed Particle Filtering in Agent Networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, 2013.

[75] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric. and M. Rupp. "Distributed Gaussian particle filtering using likelihood consensus ," *in proc. IEEE Int. Conf. on Speech and Signal Processing (ICASSP)*, pp. 3756–3759, 2011.

[76] F. Xaver, G. Matz, P. Gerstoft, and C. Mecklenbruker, N. Goertz, "Localization of acoustic sources utilizing a decentralized particle filter," *in proc. IEEE Conf on Signals, Systems and Computers (ASILOMAR)*, pp.276–280, 2012.

[77] S. S. Dias and M. G. S. Bruno, "Cooperative particle filtering for emitter tracking with unknown noise variance," *in proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2629–2632, 2012.

[78] P. M. Djuric, J. Beaudeau, and M. F. Bugallo, "Non-centralized target tracking with mobile agents," *in proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5928-5931, 2011.

[79] P. M. Djuric and L. Geng, "Non-centralized target tracking in networks of directional sensors," *in proc. IEEE Conf. on Aerospace*, pp. 85-88, 2011.

[80] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems", *Proceedings of the IEEE*, vol.95, no.1, pp. 215–233, 2007.

[81] W. Yu, G. Chen, Z. Wang, and W. Yang, "Distributed Consensus Filtering in Sensor Networks," *IEEE Trans. on Systems, Man, and Cybernetics: Cybernetics*, vol. 39, no. 6, pp. 1568–1577, 2009.

[82] M. Malek-Zavarei and M. Jamshidi, "Time-Delay Systems: Analysis, Optimization and Applications", *Elsevier Science Inc.*, 1987.

[83] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents With Switching Topology and Time-Delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[84] A. G. Dimakis, M. Kar. M. MF Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[85] D. Bickson, S. Ori, and D. Dolev, "Distributed Kalman filter via Gaussian belief propagation," *in proc. 46th Annual Allerton Conf. on Communication, Control and Computing*, pp. 628–635, 2008.

[86] F. Cattivelli, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Automatic Control*, vol.55, no.9, pp. 2069–2084, 2010.

[87] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *In Proc. IEEE IPSN*, pp. 63–70, Apr. 2005.

[88] A. O'Hagan, C. Buck, A. Daneshkhah, R. Eiser, P. Garthwaite, D. Jenkinson, Oakley and T. Rakow, "Uncertain Judgements: Eliciting Experts' Probabilities," *Wiley*, NY, 2006.

[89] M. D. McNeese, "How Video Informs Cognitive Systems Engineering: Making Experience Count," *International Journal ofCognition, Technology and Work*, 6(3), 186-196, 2004.

[90] M. D. McNeese, H. S. Bautsch and S. Narayanan, "A Framework for Cognitive Field Research," *International Journal ofCognitive Ergonomics*, 3 (4), 307-332, 1999.

[91] M. D. McNeese, B. S. Zaff, M. Citera, C. E. Brown and R. Whitaker, "AKADAM: Eliciting User Knowledge to Support Participatory Ergonomics," *The International Journal of Industria Ergonomics*, 15 (5),345-363,1995.

[92] B. S. Zaff, M. D. McNeese and D. E. Synder, "Capturing Multiple Perspectives: A User-Centered Knowledge Acquisition," *Knowledge Acquisition*, 5 (1), 79-116, 1993.

[93] M. A. Pravia, R. K. Prasanth, P. O. Arambel, C. Sidner and C. Y. Chong, "Generation of a fundamental data set for hard/soft information fusion", *Proceedings of the International Conference on Information Fusion*, pp. 1–8, 2008.

[94] K. Premaratne, M. N. Murthi, J. Zhang, M. Scheutz and P. H. Bauer, "A Dempster-Shafer theoretic conditional approach to evidence updating for fusion of hard and soft data", *Proceedings of the International Conference on Information Fusion*, pp. 2122–2129, 2009.

[95] D. L. Hall, M. D. McNeese, D. B. Hellar, B. J. Panulla and W. Shumaker, "A cyber infrastructure for evaluating the performance of human centered fusion", *Proceedings of the International Conference on Information Fusion*, pp. 1257–1264, 2009.

[96] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE transactions on Aerospace and Electronic Systems*, 39(4), pp. 1333–1364, 2003.

[97] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking-Part II: Ballistic target models," *In Proc. of SPIE Conf. on Signal and Data Processing of Small Targets*, pp. 559–581, 2001.

[98] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking- Part III: Measurement Models," *SPIE Conf. on Signal and Data Processing of Small Targets*, 2001.

[99] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking- Part IV: Decision-Based Methods," *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets,*, Orlando, FL, USA, Paper 4728-60, April 2002.

[100] D. Simon, Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," Control Theory & Applications, IET 4(8), pp. 1303–1318, 2010.

[101] E.A. Bier, S.K. Card, and J.W. Bodnar, "Entity-Based Collaboration Tools for Intelligence Analysis," *IEEE Symposium on Visual Analytics Science and Technology*, Oct., 2008.

[102] F. Karray, C. de Silva , "Soft Computing and Intelligent Systems Design: Theory, Tools and Applications," *Addison-Wesley*, 2004.

[103] H. Li, X. Zhang and S. Li, "A Three-Dimensional Fuzzy Control Methodology for a Class of Distributed Parameter Systems," *IEEE Trans. Fuzzy Systems*, vol. 15, no. 3, pp. 470–481, 2007.

[104] S. Seifzadeh, B. Khaleghi F. Karray, "Soft-Data-Constrained Multi-Model Particle Filter for Agile Target Tracking,", *16th Int. Conf. on Information Fusion* 2013.

[105] S. Seifzadeh, B. Khaleghi, and F. Karray, "A Fuzzy-Logic-Based Approach for Soft Data Constrained Multiple-Model PHD Filter," *in proc. IEEE Int. Conf. on Fuzzy Systems*, 2014.

[106] M. Tahk and J. Speyer, "Target tracking problems subject to kinematic constraints," *IEEE Trans. on Automatic Control*, vol.35, no.3, pp. 324–326, 1990.

[107] B. O. S. Eixeira, J. Chandrasekar, L. A. Aguirre, and D. S. Bernstein, "State estimation for equality-constrained linear systems," *in proc. IEEE Conf. on Decision and Control*, pp. 6220–6225, 2007.

[108] L. Lang, W.S. Chen, B. R. Bakshi, P. K. Goel and S. Ungarala, "Bayesian estimation via sequential Monte Carlo sampling-constrained dynamic systems," *Automatica*, vol. 43, Issue 9, pp. 1615–1622, 2007.

[109] J. S. Julier and J. LaViola,"On Kalman filtering with nonlinear equality constraints," *Trans. Signal Processing*, vol. 55, Issue 6, pp. 2774–2784, 2007.

[110] C. V Rao and J. B. Rawlings,"Constrained process monitoring: Moving-horizon approach," *AIChE journal*, vol. 48, Isuue 1, pp. 97–109, 2002.

[111] S. C. Patwardhan, S. Narasimhanb, P. Jagadeesanc, B. Gopalunid and S. L. Shahe, "Nonlinear Bayesian state estimation: A review of recent developments," *in proc. 4th Symposium on Advanced Control of Industrial Processes (ADCONIP)*, pp. 933-953, 2012.

[112] Z. Zhao, B. Huang and F. Liu, "A particle filter based on a constrained sampling method for state estimation," *in pro. 15th IEEE Int. Conf. on Information Fusion*, pp. 816–823, 2012.

[113] F. Papi, M. Podt, Y. Boers, G. Battistello and M. Ulmke, "On Constraints Exploitation for Particle Filtering Based Target Tracking," *in proc. 15th Int. Conf. on Information Fusion*, pp. 455–462, 2012.

[114] X. Sheng and Y. H. Hu, "Distributed particle filters for wireless sensor network target tracking," *In proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 845–848, 2005.

[115] A. Cano, A. R. Masegosa, and S. Moral, "A Method for Integrating Expert Knowledge When Learning Bayesian Networks From Data," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 41, no. 5, pp. 1382–1394, 2011.

[116] R. Chou, Y. Boers, M. Podt and M. Geist, "Performance Evaluation for Particle Filters," *Proceedings of the 14th International Conf. on Information Fusion*, pp.1–7, 2011.

[117] C. Liu and H.Y. Shum, "Kullback-Leibler boosting," *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2003.

[118] T. M. Cover and J.A. Thomas, "Elements of information theory," *Wiley*, 1991.

[119] Q. Cheng and P. K. Varshney, "Joint state monitoring and fault detection using distributed particle filtering," *in proc. 41th Asilomar Conf. Signals, Systems and Computers*, pp. 715–719, 2007.

[120] S. Thrun, D. Fox, W. Burgard, "Monte Carlo Localization With Mixture Proposal Distribution", *in proc. AAAI National Conf. on Artificial Intelligence*, 2000.

[121] A. T. Ihler, J. W. Fisher, III, and A. S. Willsky, "Using sample-based representations under communications constraints," *Massachusetts Institute of Technology, LIDS Technical Report*, no. 2601, 2004.

[122] D. Gu, "Distributed Particle Filter for Target Tracking," *in proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3856–3861, 2007.

[123] H. Zhang, J. Zhang, G. Yang, and Y. Luo, "Leader-based Optimal Coordination Control for the Consensus Problem of Multi-agent Differential Games via Fuzzy Adaptive Dynamic Programming," *IEEE Trans. on Fuzzy Systems*, doi: 10.1109/TFUZZ.2014.2310238, 2014.

[124] J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Automatic Control*, vol. 24, no. 2, pp. 266-269, 1979.

[125] B. S. Rao and H. F. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," *IEEE Control Theory and Applications*, vol. 138, no. 5, pp. 413-420, 1991.

[126] S. A. Pasha, B. N. Vo, H. D. Tuan, W. K. Ma, "A Gaussian mixture PHD filter for jump Markov system models," *IEEE Transactions on Aero- space and Electronic Systems*, vol. 45 no. 3, pp. 919-936, 2009.

[127] A. Mohammadi and A. Asif, "Distributed state estimation for large-scale nonlinear systems: A reduced order particle filter implementation," *IEEE Workshop on Statistical Signal Processing (SSP*, pp. 249-252, 2012.

[128] F. Cattivelli, C. G. Lopes, and A H. Sayed, "Diffusion strategies for distributed Kalman filtering: formulation and performance analysis," *in proc.of IAPR Workshop on Cognitive Information Processing*, pp. 36–41, 2008.

[129] F. Cattivelli, and A. H. Sayed, "Diffusion distributed Kalman filtering with adaptive weights," *in proc. of 43th Asilomar Conf. on Signals, Systems and Computers*, pp. 908–912, 2009.

[130] L. Xiaoa, S. Boydb, S. Kimb , "Distributed average consensus with least-mean-square deviation," *Journal of parallel and distributed computing*, vol. 67, pp. 33-46 2007.

[131] S. S. Dias, and M. G. S. Bruno, "Distributed Emitter Tracking using Random Exchange Diffusion Particle Filters," *in proc. IEEE Int. Conf. on Information Fusion*, pp. 1251–1257, 2013.

[132] S. S. Dias and M. G. S, Bruno, "A Rao-Blackwellized Random Exchange Diffusion Particle Filter for Distributed Emitter Tracking," *in proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp.348–351, 2013.

[133] C. M. Bishop, "Mixture models and EM," *Pattern Recognition and Machine Learning, Springer*, pp. 423–459, 2006.

[134] D. Gu, "Distributed EM algorithm for Gaussian mixtures in sensor networks," *IEEE Trans. on Neural Networks*, vol. 19, Issue 7, pp. 11541166, 2008.

[135] O. Hlinka, P. M. Djuric, and F. Hlawatsch, "Time-space-sequential distributed particle filtering with low-rate communications" *in proc. 41th Asilomar Conf. Signals, Systems and Computers*, pp. 196–200, 2009.

[136] J. H. Kotecha and P. M. Djuric, "Gaussian Particle Filtering", *IEEE Trans. Singla Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.

[137] J. H. Kotecha and P. M. Djuric, "Gaussian Sum Particle Filtering", *IEEE Trans. Singla Processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[138] D. Gu, J. Sun, Z. Hu, and H. Li, "Consensus based distributed particle filter in sensor networks," *in proc. IEEE Int. Conf. on Information and Automation, (ICIA)*, pp. 302–307, 2008.

[139] A. Simonetto, T. Keviczky, and R. Babuska, "Distributed nonlinear estimation for robot localization using weighted consensus," *in proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3026-3031, 2010.

[140] A. Mohammadi and A. Asif, "Consensus-based distributed unscented particle filter," *in proc. IEEE Workshop on Statistical Signal Processing (SSP)*, pp. 237-240, 2011.

[141] G. L. Rogova, V. Nimier, "Reliability In Information Fusion: Literature Survey," *in proc. 7th IEEE Conference on Information Fusion*, 2004.

[142] Thierry Denoeux, "A k-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Trans on Systems, Man, and Cybernetics*, 25 (5), 804-813, 1995.

[143] Galina Rogova, "Adaptive decision fusion by reinforcement learning neural network, in Distributed Reinforcement Learning For Decision-making In Cooperative Multi-agent Systems," *Part 1, CUBRC Technical report prepared for AFRL,* 2003.

[144] Jean-Francois Grandin and Miguel Marques, "Robust Data Fusion," *in: Proc. of the Third Int. Conf. on Information Fusion*, Paris, France, 2000.

[145] Francois Delmotte, Laurent Dubois, "Multi-models and belief functions for decision support systems, In: IEEE Int. Conf. on Systems, Man, and Cybernetics, 1, 181 186, 1996.

[146] Moshe Kam, Chris.Rorres, Wei. Chang, and Xiaoxun.Zu, "Performance and geometrical interpretation for decision fusion with memory," *IEEE Trans. On SMC Part A*, 29(1). 52-62, 1999.

[147] Pei Wang, "Confidence as higher level of uncertainty," *in Proc. of the Int. Symp. on Imprecise Probabilities and Their Applications*, Ithaca, NY, 352–361, 2001.

[148] Vincent Nimier, "Supervised multisensor tracking algorithm by context analysis," *in Proc. of the First Int. Conf. On Information Fusion*, pp. 149-156, 1998.

[149] V. Nimier, "Introducing Contextual Information in Multisensor Tracking Algorithms," *in Proc. of the 5th International Conference on Management of Uncertainty in Knowledge-Based Systems*, pp. 595–604, 1994.

[150] G. Rogova, J. Kasturi, "Reinforcement Learning Neural Network For Distributed Decision Making," *in Proc. of the Forth Conf. on Information Fusion*, 2001.

[151] Francois Delmotte, and Pierre Borne, "Context-dependent trust in data fusion within the possibility theory," *in Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics,* vol. 1, pp. 538–543, 1996.

[152] Ramon Parra-Loera, Wiley Thompson, Ajit Salvi, "Adaptive selection of sensors based on individual performance in a multisensor environment," *in Proceedings of the Meeting on Data structures and target classification*, pp. 30–36, 1991.

[153] S. Seifzadeh, B. Khaleghi, F. Karray, "Distributed Soft-Data-Constrained Multi-Model Particle Filter," *IEEE Trans. on Systems, Man, and Cybernetics, DOI:10.1109/TCYB.2014.2326549*, 2014.

[154] S. Seifzadeh, B. Khaleghi, and F. Karray, "A Source-Reliability-Adaptive Distributed Soft-Data-Constrained Multi-Model Particle Filter, *in proc. IEEE Int. Conf. on System, Man, Cybernetics*, 2014.

[155] B. Vo, S. Singh, A. Doucet, "Sequential Monte Carlo Implementation of the PHD Filter for Multi-target Tracking," *in proc. Int. Conf. Information Fusion*, pp. 792–799, 2003.

[156] S. Musick, J. Greenwald, C. Kreucher, and K. Kastella, "Comparison of particle method and finite difference nonlinear filters for low SNR target tracking," *Proc. of the fourth annual Conf. on Information Fusion*, Montreal, Canada, 2001.

[157] Johansen, A., Singh, S., Doucet, A., and Vo, B. "Convergence of the sequential implementation of the PHD filter," *Technical Report*, 2005.

[158] J. Yoon, D. Kim, K. Yoon, "Efficient importance sampling function design for sequential Monte Carlo PHD filter," *IEEE Trans. on Signal Processing*, vol. 92, no.9, pp. 2315-2321, 2012.

[159] K. Punithakumar, T. Kirubarajan, A. Sinha, "Multiple-model prob- ability hypothesis density filter for tracking maneuvering targets," *IEEE Trans. Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 87-98, 2008.

[160] J. H. Kotecha and P. M. Djuric, "Gaussian Particle Filtering", *IEEE Trans. Singla Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.

[161] D. Schuhmacher, B. T. Vo, B. N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3447-3457, 2008.

[162] M. Maehlisch, R. Schweiger, W. Ritter, K. Dietmayer, "Multisensor vehicle tracking with the probability hypothesis density filter," *in Proc. of the IEEE Int. Conf. on Information Fusion*, pp. 1–8, 2006.