# PARTICLE SWARM OPTIMIZATION BASED PARTICLE FILTER TECHNIQUES FOR TARGET TRACKING IN MULTISTATIC UWB RADAR SENSOR NETWORK

BY

**MUHAMMAD MUJAHID AMIN**

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

**MASTER OF SCIENCE**

**In**

**ELECTRICAL ENGINEERING**

DECEMBER 2014

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

## DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MUHAMMAD MUJAHID AMIN** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING DEPARTMENT**.

**Thesis Committee**

Dr. Samir Nasser Alghadhban
(Adviser)

(Co-adviser)

Dr. Abdelmalek Chikh Zidouri
(Member)

Dr. Saad Mohammad Alahmadi
(Member)

(Member)

Dr. Ali Ahmad Al-Shaikhi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

2/2/15
Date

©Muhammad Mujahid Amin
2014

*Dedicated to my parents, their prayers are best companion for worst and best times.*

# ACKNOWLEDGMENTS

*"In the name of Allah, the Most Gracious, the Most Merciful.*

*I am thankful to Allah for His blessings upon me to complete this work . I am grateful to KFUPM which provide me opportunity for this thesis work. My sincere thanks to my thesis advisor " Dr. Samir Nasser Alghadhban" for his meticulous attention, useful guidance, efforts and patience during whole consoling period. I appreciate my thesis committee members Dr. Abdelmalek Chikh Zidouri and Dr. Saad Mohammad Alahmadi for their valuable constructive suggestions. I am thankful to Dr. Ali Muqaibel and Dr. Adnan Landolsi for enlightening me concepts of stochastic processes and statistical signal processing which became basis of my thesis work. I thank to my family, colleagues and friends for their emotional and technical support.*

# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS** iii

**TABLE OF CONTENTS** iv

**LIST OF TABLES** vii

**LIST OF FIGURES** viii

**LIST OF ABBREVIATIONS** x

**ABSTRACT (ENGLISH)** xii

**ABSTRACT (ARABIC)** xiv

**CHAPTER 1 INTRODUCTION** 1

- 1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  - 1.1.1 Radar Sensor Network . . . . . . . . . . . . . . . . . . . . 2
  - 1.1.2 Particle Filter . . . . . . . . . . . . . . . . . . . . . . . 6
  - 1.1.3 Particle Swarm Optimization . . . . . . . . . . . . . . . . 7
- 1.2 Thesis Contributions . . . . . . . . . . . . . . . . . . . . . . . 8
- 1.3 Thesis Layout . . . . . . . . . . . . . . . . . . . . . . . . . . 9

**CHAPTER 2 LITERATURE REVIEW** 11

**CHAPTER 3 SYSTEM DESCRIPTION AND PARTICLE FILTER, PARTICLE SWARM OPTIMIZATION ALGORITHMS** 19

- 3.1 SYSTEM DESCRIPTION . . . . . . . . . . . . . . . . . . . . . . . 20

3.1.1 Target movement model . . . . . . . . . . . . . . . . . . . . 20
3.1.2 UWB Multistatic Radar Setup (Measurement Model) . . . 22
3.1.3 Maximum likelihood Position estimation based on Time of Arrival . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
3.2 Particle filter Algorithm . . . . . . . . . . . . . . . . . . . . . . . . 27
3.2.1 Prediction Step . . . . . . . . . . . . . . . . . . . . . . . . 30
3.2.2 Correction Step . . . . . . . . . . . . . . . . . . . . . . . . 31
3.2.3 Degeneracy problem . . . . . . . . . . . . . . . . . . . . . 33
3.2.4 Resampling . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.2.5 Resampling impoverishment problem . . . . . . . . . . . . 34
3.3 PARTICLE SWARM OPTIMIZATION . . . . . . . . . . . . . . 36
3.3.1 Functionality of PSO algorithm . . . . . . . . . . . . . . . 36
3.3.2 PSO bias and slow convergence issues . . . . . . . . . . . . 38

**CHAPTER 4 PROPOSED ALGORITHMS 40**
4.1 AW-PSO-Particle Filter . . . . . . . . . . . . . . . . . . . . . . . 40
4.1.1 Initialization . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.1.2 Time update . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.1.3 Measurement update . . . . . . . . . . . . . . . . . . . . . 46
4.1.4 Adaptive inertia Weight Particle Swarm Optimization . . . 46
4.1.5 Particle Filter Processing . . . . . . . . . . . . . . . . . . . 47
4.1.6 Particle Filter estimation Stage . . . . . . . . . . . . . . . 48
4.2 Distributed Particle swarm optimization Particle Filter . . . . . . 48
4.2.1 Distributed Particle Swarm Optimization . . . . . . . . . 50
4.2.2 Modified Particle Filter processing . . . . . . . . . . . . . 52

**CHAPTER 5 EXPERIMENTAL SETUP AND RESULTS 54**
5.1 Test setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
5.1.1 Radar sensor network parameters . . . . . . . . . . . . . . 55
5.1.2 Target model . . . . . . . . . . . . . . . . . . . . . . . . . 56
5.1.3 Testing trajectory and Target velocities . . . . . . . . . . 57

5.1.4 Algorithm parameters . . . . . . . . . . . . . . . . . . . . . 59

5.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

5.2.1 Algorithm tracking Error . . . . . . . . . . . . . . . . . . . . 60

5.2.2 Algorithm tracking Robustness . . . . . . . . . . . . . . . . 65

5.2.3 Computation Complexity Comparison of Algorithm . . . . 66

5.2.4 Tracked and untracked trajectories . . . . . . . . . . . . . 67

**CHAPTER 6 CONCLUSION AND FUTURE WORK 73**

**REFERENCES 76**

**VITAE 88**

# LIST OF TABLES

4.1 List of AW-PSO-PF parameters . . . . . . . . . . . . . . . . . . . . . 44

5.1 Tracking Error Comparisons . . . . . . . . . . . . . . . . . . . . . . . 65

5.2 Tracking robustness comparisons . . . . . . . . . . . . . . . . . . . . 65

5.3 Computational Complexity of Algorithms . . . . . . . . . . . . . . . 66

5.4 Numeric Computational Complexity of algorithms . . . . . . . . . 67

# LIST OF FIGURES

2.1 Trilateration Technique . . . . . . . . . . . . . . . . . . . . . . . . 13
2.2 Triangulation Technique . . . . . . . . . . . . . . . . . . . . . . . . 14

3.1 Hidden Markov Model . . . . . . . . . . . . . . . . . . . . . . . . 29
3.2 PF algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.3 PSO Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

4.1 AWPSOPF Block diagram . . . . . . . . . . . . . . . . . . . . . . 43
4.2 AWPSOPF Algorithm . . . . . . . . . . . . . . . . . . . . . . . . 49
4.3 DPSOPF Block diagram . . . . . . . . . . . . . . . . . . . . . . . 51
4.4 DPSOPF Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 53

5.1 Radar Sensor Network Setup . . . . . . . . . . . . . . . . . . . . 55
5.2 First derivative Gaussian monocycle pulse with reflection time wait and its spectrum . . . . . . . . . . . . . . . . . . . . . . . . . . 56
5.3 Straight line trajectory . . . . . . . . . . . . . . . . . . . . . . . . 57
5.4 Circle trajectory . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.5 Straight line - Ellipse - Straight line trajectory . . . . . . . . . . . 58
5.6 Zigzag trajectory . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
5.7 Best and worst SNR where SNR due to noise is -20dB . . . . . . . 60
5.8 RMS tracking error on straight line trajectory . . . . . . . . . . . 62
5.9 RMS tracking error on circular trajectory . . . . . . . . . . . . . . 63
5.10 RMS tracking error on a straight line, ellipse & straight line trajectory 64
5.11 RMS tracking error on Zigzag trajectory . . . . . . . . . . . . . . 64

5.12 Target tracking on a straight line trajectory SNR 5 dB by PF . . 67
5.13 Target tracking lost on Straight line trajectory SNR -10 dB by PF 68
5.14 Target tracking on circular trajectory SNR 5dB by DPSOPF . . . 69
5.15 Target Tracking lost on Circular trajectory SNR 0dB by PF . . . 69
5.16 Target tracking on straight line elliptical trajectory SNR 5dB by AWPSOPF . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
5.17 Target Tracking lost by PSO on Straight line elliptical trajectory and SNR -15dB . . . . . . . . . . . . . . . . . . . . . . . . . 71
5.18 Target Tracking by PSO on zigzag trajectory and SNR 5dB . . . 71
5.19 Target Tracking lost by PF on Zigzag trajectory and SNR -20dB 72

6.1 Particle distribution in squares . . . . . . . . . . . . . . . . . . . . . 75

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AOA | Angle Of Arrival |
| AWPSO-PF | Adaptive Weight Particle Swarm Optimization Particle Filter |
| CRLB | Cramer Rao Lower Bound |
| CW | Continuous Wave |
| DPSO-PF | Distributed Particle Swarm Optimization Particle Filter |
| EKF | Extended Kalman Filter |
| EM | Electromagnetic |
| HDM | Hidden Markov Model |
| IR | Impulse Radio |
| KF | Kalman Filter |
| LDS | Linear Dynamic system |
| MLE | Maximum Likelihood Estimation |
| MVUE | Minimum variance Unbiased Estimator |

| | |
|---|---|
| PDF | Probability Distribution Function |
| PF | Particle filter |
| PRF | Pulse Repetition Frequency |
| PRT | Pulse Repetition Time |
| PSO | Particle swarm optimization |
| RF | Radio Frequency |
| RF ID | Radio Frequency Identification |
| RMSE | Root mean Square Error |
| RSN | Radar Sensor Network |
| RSS | Received Signal Strength |
| SIMO | Single Input Multiple Output |
| SMC | Sequential Monto Carlo |
| SNR | Signal-to-Noise Ratio |
| TDOA | Time Difference of Arrival |
| TOA | Time Of Arrival |
| UWB | Ultra Wide Band |
| WSN | Wireless Sensor Network |

# THESIS ABSTRACT

| | |
|---|---|
| **NAME:** | Muhammad Mujahid Amin |
| **TITLE OF STUDY:** | Particle swarm optimization based Particle filter techniques for Target Tracking in multistatic UWB radar sensor network |
| **MAJOR FIELD:** | Electrical Engineering Department |
| **DATE OF DEGREE:** | December, 2014 |

*Sensor networks are mainly used for applications such as emergence detection, target of interest monitoring, non-cooperative human object detection such as an intruder (target) for border surveillance, intrusion unauthorized movement around critical facilities. In this thesis, two new algorithms are proposed for target (human intruder) tracking in an Ultra WideBand (UWB) multistatic Radar Sensor Network (RSN) consisting of one transmitter and multiple receivers. These algorithms are based on Particle Filter (PF) with embedded variants of Particle Swarm Optimization (PSO) techniques to provide a solution of tracking problem in dynamic and noisy environments. First algorithm; Adaptive inertia Weight Particle*

*Swarm Optimization Particle Filter (AWPSOPF), is a PF embedded with fitness based adaptive inertia weight PSO that improves the convergence of PSO, tackle PSO bias issue, solves PF sample impoverishment problem and improve tracking accuracy. The second algorithm; Distributed Particle Swarm Optimization Particle Filter (DPSOPF), is an enhanced version of AWPSOPF where distributed PSO is embedded in PF and PSO particles are divided into further small groups based on minimum distance which provides a robust solution for target tracking problem.*

## ملخص الرسالة

| | |
|---|---|
| **الاسم** | محمد مجاهد أمين |
| **عنوان الرسالة** | تقنيات سرب الجسيمات المتحسن (PSO) على اساس مرشح الجسيمات (PF) لتتبع الهدف في شبكة اجهزة اسشعار الرادار ذات النطاق الفائق العرض (UWB) المتعدد |
| **القسم** | الهندسة الكهربائية |
| **التاريخ** | ديسمبر، 2014 |

شبكات الاستشعار تستخدم بشكل رئيسي في التطبيقات المختلفة كالكشف عن الأشياء و مراقبة الأجسام المستهدفة أو الكشف عن الأشخاص غير المتعاونين كالمتسللين عبر الحدود، أو الحركات غير المصرح بها حول المنشآت الحساسة.

في هذه الرسالة، تم اقتراح خوارزميتين لتتبع مسار الهدف كالمتسلل البشري في نطاق عريض من شبكات الاستشعار الرادارية (UWB) المتضمنة لمرسل واحد وأجهزة متعددة الاستقبال.

هذه الخوارزميات مبنية على مرشح الجسيمات المضمنة مع خوارزمية أسراب الجسيمات لتقديم الحلول المناسبة لمشكلة تتبع المسار في البيئات الديناميكية و المشوشة.

الخوارزمية الأولى المعتمدة على الحل الأمثل باستخدام مرشح الجسيمات المبنية على أسراب الجسيمات المتكيفة والمعتمدة على القصور الوزني (AWPSOPF) والذي بدوره هو المرشح للجسيمات والمضمن مع كفاءة والمعتمد على القصور الوزني التكيفي لخوارزمية أسراب الجسيمات والتي تسهم في تحسين الحل التقاربي للخوارزمية، بالاضافة الى معالجة قضية التحيز للخوارزمية، بالاضافة الى مساهمتها في حل مشكلة قلة العينات وتطوير دقة التتبع.

أما الخوارزمية الثانية فهي مرشح الجسيمات المعتمدة على خوارزمية أسراب الجسيمات الموزعة (DPSOPF) والتي هي النسخة الأكثر تطورا من (AWPSOPF) من حيث أنها مضمنة مع مرشح الجسيمات وأن جسيمات (PSO) مقسمة الى مجموعات أصغر نسبيا من الأولى والمبنية على المسافة الأقل بين الجسيمات والتي تعطي بدورها حلولا أقوى لمشاكل تتبع الأهداف.

# CHAPTER 1

# INTRODUCTION

Sensor networks, for indoor surveillance, have caught the attention of researchers in the last decade. Sensor networks are mainly used for applications such as emergence detection, target of interest monitoring, non-cooperative human object detection such as an intruder (target) for border surveillance, intrusion unauthorized movement around critical facilities and to precisely estimate the target location at consecutive time instants (target tracking) [1]. Target tracking is the estimation of an object kinematic state, this may be its current state or its future state based on current state [2]. The ability to track an object is a valuable feature for any sensor network as so more focus will be given to the area of interest [3].

The radar sensor network (RSN) is an optimum choice for sensor networks [4]. Radar acts as all condition all weather sensor and performs in diverse environments [3]. Radar is an active sensor as it transmits an electromagnetic signal (EM), it senses larger search space compared to other passive sensors [5]. Therefore radar sensor network requires less number of sensors to cover the same space compared to other sensor networks [4].

Indoor target tracking with radar sensor networks is a challenging task due to unpredictable motion of the target and the presence of multipath [6]. In this work, two new algorithms are proposed for target (human intruder) tracking in an ultra wideband (UWB) multistatic radar sensor network (RSN) consisting of one transmitter and multiple receivers. These algorithms are based on Particle Filter (PF) with embedded variants of Particle Swarm Optimization (PSO) techniques to provide a solution of tracking problem in dynamic and noisy environments. First algorithm; Adaptive inertia Weight Particle Swarm Optimization Particle Filter (AWPSOPF), is a PF embedded with fitness based adaptive inertia weight PSO that improves the convergence of PSO, tackle PSO bias issue, solves PF sample impoverishment problem and improve tracking accuracy. The second algorithm; Distributed Particle Swarm Optimization Particle Filter (DPSOPF), is an enhanced version of AWPSOPF where distributed PSO is embedded in PF and PSO particles are divided into further small groups based on minimum distance which provides a robust solution for target tracking problem.

## 1.1 Background

### 1.1.1 Radar Sensor Network

The “radar” is abbreviated from RAdio Detection And Ranging. In general, radars use electromagnetic (EM) waveforms and directional antennas transmit

EM energy into a search volume in space to detect targets [3]. Targets within a search volume reflect back portions of this EM energy (radar echoes/returns) towards the radar. Then the radar receiver processes these echoes to obtain the target information such as location, angular position, velocity, and other target characteristics [7]. Radar systems can be classified into many categories based on the specific characteristics, such as location, mission, operating frequency band, antenna type and location, and waveforms utilized [3].

Radars are classified based on their operating location as ground, airborne or ship based radar systems. Another classification is based on the mission and the functionality of the radar such as weather, search, tracking, early warning radars etc [7]. Radars are also classified by their waveform types such as continuous wave (CW) or pulsed radars [7]. CW radars have separate non-collocated transmitting and receiving antennas and accurately measure target radial velocity (based on Doppler shift) and target angle [3]. CW radars require some form of modulation to measure target range information. Pulsed radars emit pulsed waveforms and can be classified on the basis of the pulse repetition frequency (PRF) such as high PRF (>1000 pulses per second (pps)), medium PRF (>350 pps and<1000 pps) and low PRF (<350 pps) [3]. Pulsed radar measures target range accurately, high PRF radar also measures radar target radial velocity. Radars may be classified based on the operating frequency band. Low frequency radars mainly used in ships and on the ground where high frequency radars mainly used in airborne applications.

Radars are also classified based on intra Pulse modulation such as pulse code modulation, linear frequency modulation and phase modulation [3]. Pulse bandwidth is also used to classify radar as narrowband, wideband and ultra-wideband (UWB). Radar systems can also be classified based on antenna locations such as i) mono-static where transmit and receive antennas are positioned on same location or single antenna is multiplexed for transmitting and receiving signals, ii) bi-static where transmit and receive antennas are positioned on different locations for transmission and reception of EM signals, and iii) multi-static radar, which consists of at least three or more non collocated transmitting and receiving antennas [8].

Radars are also classified as an active and passive based on EM signal transmission mechanism. Active radar transmits known EM signal that receiver knows in advance and the positions of the transmitter and receiver are known, passive radar does not know about transmitted signal and transmitter location in advance, it tries to estimate transmitted signal and location of transmitter and treats unknown transmitter as a target [3].

Multi-static radar is a natural choice for Radar Sensor Network (RSN). Due to spatial diversity in multi-static radar, more information is available and wide coverage area is gained [8]. Impulse Radio (IR) Ultra-wideband (UWB) technology is an ideal candidate for radar sensor network to be used for indoor surveillance due to the obstacle penetration ability of UWB signals and more immune to multipath effects [1]. UWB radar sensors are not dependent upon environmental exposure and also not affected by environment exposure compare to other sensors, for example, visual sensor (which depend on light) and acoustic senor (which are affected by voice disturbance) [4]. UWB radar sensors can be concealed in any object like trees, rocks etc [4]. UWB radar sensors can co-exist with large number of devices operating in their surroundings [1]. UWB sensors use low power and short pulses for transmission, these pulses are considered as the background noise by other sensing devices and this makes UWB sensor networks prone to jamming [1]. These short pulses offer an extraordinary resolution and localization precision even in indoor multi-path environment [4, 9]. Impulse radio technology also reduces system complexity, there is no need of up convertors and down convertors [10].

There are many approaches for target tracking using radar sensors. The most effective ones are based on estimating parameters such as time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength (RSS) [6, 11].

Radar sensor networks collect measurements from all the receivers and combine them. The processing unit estimates the present state of the target based on the received measurements and previous target states. This new update is reported to system users. These measurements are nonlinear functions of target state and mainly consist of position and velocity [12].

### 1.1.2 Particle Filter

There are many electronic signal processing systems that rely on estimation theory to extract information as Radar, Robotics, Seismology, Image analysis, Communications and Control etc [13]. Estimators are classified as i) optimal estimators that achieve Cramer Rao lower bound (CRLB) with finite observations as minimum variance unbiased estimators (MVUE), and ii) suboptimal estimators that achieve CRLB if infinite observations are available [13]. Estimators are also classified based on estimation parameter of intersect. If the estimated parameter is deterministic and unknown constant, then these estimators are classical estimators and if the estimated parameter is a random variable and estimators are estimating their particular realization then these estimators are Bayesian estimators [13].

Particle filter (PF) belongs to Bayesian estimators suboptimal filter family. PF is a sequential Monte Carlo (SMC) method as particles (point masses) in PF represents posterior probability of a state at a given time for SMC based estimation [14]. PF is a solution for Hidden Markov Model (HMM) in which system have observable variables (observation process) and hidden variables (state process). These variables are related by some known function and the probabilistic model of dynamical system that relates change of state variables is known [15]. PF based estimation is a desired solution for nonlinear, non-Gaussian and multimodal probability distribution cases [16].

PF was first proposed by the name of poor man's Monte Carlo by Hammersley in 50's [17]. PF was reinvented by Gordon when he published his work "A novel

Approach to nonlinear/non-Gaussian Bayesian State estimation” in 1993 [18] as his algorithm (PF) does not require any prior assumptions about noise and system state.

### 1.1.3 Particle Swarm Optimization

Particle swarm optimization (PSO), introduced by Russell Eberhart and Kendey [19, 20], is a versatile population based optimization based on the intelligence and movement of swarms for nonlinear functions. PSO derives its inspiration from the aggregate motion of a flock of birds, herd of animals and similar natural group movements of others spices. Swarm of birds has many contrasts as it consists of discrete birds, but swarm’s overall motion is synchronized like fluid motion. This is because the bird’s location looks randomly distributed, but their motion is synchronized by some intentional and centralized control. Swarm motion is aggregate result of the actions of individuals but there is some type of information sharing between swarm individuals. Based on these observations Craig W. Reynolds coined term of Particle swarm for computer graphic design on basis of collision avoidance, velocity matching and swarm centering similar to bird swarms [21].

PSO was originally used to solve non-linear continuous optimization problems, but more recently it has been used in many practical, real-life application problems. PSO draws inspiration from the sociological behavior associated with bird flocking. It is a natural observation that birds can fly in large groups with no collision for extended long distances, making use of their effort to maintain an optimum distance between themselves and their neighbors.

PSO works like a cooperative system, where each particle is able to take the decision based on its individual current and past observations, this decision is modified based on observations from other particles of the swarm.

## 1.2 Thesis Contributions

This thesis proposes two new algorithms to improve target tracking in radar sensor networks and provide a robust solution of tracking problem. The first proposed algorithm is an Adaptive inertia Weight Particle Swarm Optimization based on Particle Filter (AWPSO-PF) which uses the diversity of particles of PSO to improve re-sampling of particle filter (PF). This improves particle filter performance and overcomes computation overhead in addition. Adaptive weights are used for fast convergence of particle swarm optimization. The second proposed algorithm distributes particle swarm into small swarms. In the new proposed algorithm, particles can more diversely explore the search space. The probability of particles trapping in local optimized points is reduced. These distributed swarms of particles are used for resampling of a particle filter. This mechanism will improve overall performance and makes the algorithm robust. This proposed algorithm is named as Distributed Particle Swarm Optimization based Particle Filter (DPSO-PF).

These algorithms mainly address the PF sample impoverishment issue, PSO bias issue, PSO slow convergence issue and PSO tendency to trap local optimal solution issue for target tracking in RSN. The PF faces degeneracy problem, after some iterations only few particles will remain with significant weights compared to the remaining particles [16]. The solution of this degeneracy problem is to resample particles where low weight particles are replaced with high weight particles. During PF re-sampling step, there is a probability that large weight particles will be picked multiple times compared to small weight particles so diversity of particles tend to decrease after re-sampling and this phenomena is known as sample impoverishment [16].

As PSO directs all particles towards optimum solutions, all particles memorize their best position in swarm. This increases the significant weights of all particles and overcome PF sample impoverishment issue, PSO has a bias issue as PSO particles have tendency to move toward coordinate axis of search space [22–24]. Where PSO convergence is improved by using adaptive weights and distributed swarms are used to minimize probability of PSO to trap in local optimal solutions.

## 1.3 Thesis Layout

This thesis is divided into six chapters. First chapter is introduction to target tracking problem, Radar related terms, PSO and PF algorithms and thesis contributions. Second chapter is a literature review related to estimation and tracking problem in wireless and radar sensor network. Third chapter mainly describes the system model that will be used for algorithm, includes the derivation of maximum likelihood estimator for tracking problem and functionality of PF, PSO algorithms. Fourth chapter is a description of proposed algorithms and how these algorithms are functioning on linear dynamic system. Fifth chapter describes the simulated data models and results by applying the new proposed algorithm compare to conventional algorithms. Sixth chapter is concluded remarks and proposed future work.

# CHAPTER 2

# LITERATURE REVIEW

UWB based target position estimation and tracking can be divided in three categories. First one is based on mapping or finger printing where the target search area is well scanned and a database maintains previous known estimated values of modeled and simulated target. In real scenarios, target estimate is based on this database. Mainly mapping algorithm uses RSS values and compare to recorded data which is called radio fingerprinting to estimate presence and track target [1].

Probabilistic method, k nearest neighbor and neural networks are the main three techniques that are used for position estimation based on radio finger printing. Youssef et. al. proposed a probabilistic based clustering technique by utilizing probability distribution of signal strength with clustering of locations to estimate the position of the target [25]. Bahl et. al. proposed k nearest neighbor based position estimation technique by utilizing radio finger printing combined with the signal processing model [26]. Battiti et. al. proposed neural network based technique where neural networks are used to compensate nonlinearity between radio fingerprinting and RSS for position estimation [27]. Headley et. al. proposed a position estimation algorithm by change in multipath due to presence of target for UWB RSN [28]. Svecova et. al. proposed a position estimation based on the least square difference and Taylor series approximation of current received signal with radio frequency RF map [29]. All above methods require a record of radio fingerprinting, however for many scenarios, this is not possible and not practical [1]. Also fingerprinting methods require a lot of computation power as these algorithms are based on comparing and matching [1].

A second category of position estimation is based on geometrical techniques to estimate target position from direct measurements so these techniques do not require a database. Trilateration is a geometric technique that computes the position based on the TOA estimates between the target and radar sensors. Distances between target and radar sensor calculated based on these TOA. These distances are used to estimate target position by calculating the point of intersection of circles drawn by these distances as shown in figure 2.1. At least three radar sensors at different locations required to compute the position of target [30]. $d_1, d_2$ and $d_3$ are distances calculated based on TOA from the target node located at $(x_t, y_t)$ where sensor nodes are located at $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$ respectively. Target position can be calculated by following equations [30].

$$x_t = \frac{(y_2 - y_1)(x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2) + (y_2 - y_3)(x_1^2 - x_1^2 + y_1^2 - y_2^2 + d_2^2 - d_1^2)}{2[(x_2 - x_3)(y_2 - y_1) + (x_1 - x_2)(y_2 - y_3)]} \tag{2.1}$$

$$y_t = \frac{(x_2 - x_1)(x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2) + (x_2 - x_3)(x_1^2 - x_1^2 + y_1^2 - y_2^2 + d_2^2 - d_1^2)}{2[(x_2 - x_1)(y_2 - y_3) + (x_2 - x_3)(y_1 - y_2)]} \tag{2.2}$$

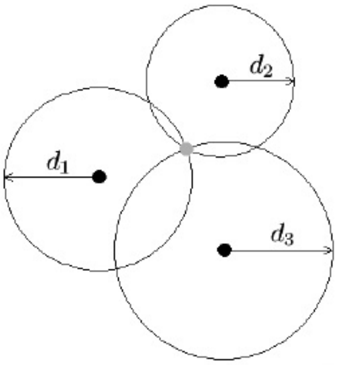

Figure 2.1: Trilateration Technique

Triangulation is another geometric technique that computes the position based on the angle of arrival (AOA) estimation between the target and radar sensors. Target position is estimated based on the intersection of two lines as shown in figure 2.2. This technique requires at least two radar sensors [31]. $\psi_1$ and $\psi_2$ indicates AOA at sensor nodes from the target node located at $(x_t, y_t)$, these nodes are located at $(x_1, y_1)$ and $(x_2, y_2)$ respectively. Then target position can be calculated by following equations [31]

$$x_t = \frac{x_2\ tan\psi_2 - x_1\ tan\psi_1 + y_1 - y_2}{tan\psi_2 -\ tan\psi_1} \tag{2.3}$$

$$y_t = \frac{y_1\ tan\psi_2 - y_2\ tan\psi_1 + (x_2 - x_1)tan\psi_1 tan\psi_2}{tan\psi_2 -\ tan\psi_1} \tag{2.4}$$

Geometric techniques are straight forward if measurements are error free which is not possible in real world scenarios. These schemes do not provide exact one point intersection when measurements are noisy [1]. Due to noise in the measurements, the solution of geometric technique provides multiple intersection points

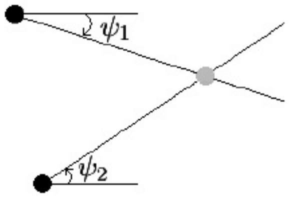

Figure 2.2: Triangulation Technique

which means that estimated target is present at multiple positions that is not possible and this is called position ambiguity [1]. Geometric approaches faces real challenges due to position ambiguity problem as these techniques require error free measurements which is not possible in multipath, multi scatter and cluttered environment [1]. In most of cases, these schemes are not practical and require an additional stochastic algorithm to compensate their shortcomings [1]. To improve performance of geometric techniques, a number of methods are proposed. Groginsky suggested a recursive position estimator by using trilateration and maximum likelihood [32]. Manolakis proposed an algorithm for the position estimates based on uncertainty in the range measurements and special distribution of reference points [33]. Thomas and Ros derived formulae for position estimate also based on trilateration by using CayleyMenger determinants instead of centric coordinates to estimate the position error and this estimated error is used to compensate the target position estimate [34]. Decarli et. al. proposed a joint position estimation technique with radio frequency identification (RF ID) and RSN based on variation at RF ID receiver antenna and trilateration [35].

Due to limitation of mapping and geometric techniques, the most suitable choice for estimation is statistical techniques. The main focus of these estimators is Bayesian approaches as movement of the target is also modeled as a random process [36, 37]. The most popular Bayesian statistical estimation technique is Kalman Filter (KF) for tracking applications due to its recursive nature and low computation complexity. It offers an optimal solution under the assumption of linearity and Gaussian measurement noise distribution [38]. A lot of Kalman filter approaches are recently proposed for wireless sensor networks (WSNs) to tackle the effects of uncertainty and measurement noise in target tracking applications [39–43]. KF implemented in wireless sensor networks by two strategies: one is based on one common central processing unit which is called fusion center and the other is decentralized approach where sensor nodes near proximity of target take measurements and communicate this data to other nodes. Thus KF is implemented locally node to node. Since KF assumes that noise probability distribution function (PDF) is Gaussain at every step, posterior probability density is characterized by only its mean and covariance and system is described by only linear equations. However, many real world systems are nonlinear and noise is not necessarily Guassain in many cases [16, 18, 44]. RF prorogation mainly modeled by Rayleigh or Rice distribution and target movement is described by Newton's law of motions that results in nonlinear equations [45, 46]. Thus, KF is not the best choice for target tracking in radar sensor network. A solution of linear and Gaussain assumption problem is to use extended Kalman filter (EKF) which utilizes the first term in a Taylor expansion where probability density function (PDF) is approximated as Gaussian [47]. EKF reduces linearization errors in range by using the measurement components of a detection recursively in a certain order, but this approach is not very effective at shorter ranges [48]. Di and Joo presented a comprehensive study of implementation of KF and EKF which is based on one or two measurements available from TOA, DTOA, RSS and angle of arrival [49]. Hongyang et. al. proposed EKF for position estimation in WSN based on TDOA measurements [50]. Caceres et. al. proposed adaptive EKF for position estimation in WSN based on RSS measurements [51]. Ribeiro et. al. proposed many variants of EKF based on TOA measurements for central and distributed computing configurations of wireless sensor networks [52]. EKF is stable and estimation converges if parameters carefully tuned according to specific conditions and the system model is well approximated. Otherwise, if these approximations are unreliable, EKF is prone to diverge or converge slowly [53].

There are many new approaches for recursive nonlinear estimation that appeared in literature in the last decade. Particle filter (PF) is a statistical estimation technique based on recursive Monte Carlo statistical signal processing that provides a desired solution for the estimation problem in non linear, non Gaussain cases. PF performs sequential Monte Carlo estimation by utilizing independent random samples of probability densities and these samples called particles and weight associated with particles represent point mass representations of the posterior probability density function of the state. PF basic framework is introduced in 50's, but after the introduction of resampling phase in 90's, large scale and low cost availability of computation devices, drastically increased PF based estimation in many fields [17,18]. A resampling step to replace low weight particles with high weight particles until the number of particles satisfies threshold. Gustafsson et. al. described a general frame work for particle filter based position estimation and tracking [15, 54]. Earliest usage of PF by Arulampalam et al. for ground radar when they proposed PF based algorithm for ground moving target indication [55]. Housfater et al. proposed PF based algorithm to estimate missing measurements in multistatic radar network [56]. Ing proposed a distributed version of PF for object tracking in sensor networks [57].

PF without resampling faces the degeneracy problem where particle weights decrease drastically so after few iterations only few particles are contributing towards estimation. A suitable metric for degeneracy is introduced by Liu and Rong [58]. Resampling eradicate this problem but cause another phenomena called sample impoverishment. As in resampling there is more probability that large weight particles will be picked multiple times compare to small weight particles. Therefore, diversity of particles tends to decrease and in extreme case, all particles might converge into a single particle. This will degrade particle filter estimation. A solution to this problem is resampling by particle swarm optimization.

Particle swarm optimization is a modern optimization algorithm, which belongs to the category of swarm intelligence methods. Particle swarm optimization was originally used to solve nonlinear continuous optimization problems, but more recently it has been used in many practical, real life application problems. The particle swarm is an ad hoc system where each particle is on its own and acts upon its local information, but as a group of particles, the swarm of particles is capable of self organizing to perform complex tasks. Due to intercommunication among the particles, the formation of complex structures is possible at swarm level, which helps in solving complex optimization problems.

Schwaab et al and He et al used PSO for nonlinear parameter estimation [59, 60]. Low and Guo proposed an optimal location assignment by using PSO [61]. Kulkarni et al., Gopakumar and Jacob proposed wireless sensor position estimation scheme by using PSO [62] [63]. PSO has a bias issue as PSO particles have tendency to move toward coordinate axis of search space [22–24].

# CHAPTER 3

# SYSTEM DESCRIPTION AND PARTICLE FILTER, PARTICLE SWARM OPTIMIZATION ALGORITHMS

A moving target represents a dynamic system and dynamic system requires at least two models. The first model is a motion model that describes the change in the target state with respect to time, it is also called system model. The second model is a measurement model which relates the noisy measurements to the state.

## 3.1 SYSTEM DESCRIPTION

### 3.1.1 Target movement model

Target motion is modeled as a piecewise linear dynamic system (LDS). General form of LDS for stochastic process is given in the following equation [64]

$$s(t) = F\dot{s}(t) + G(t)u(t) + n(t) \tag{3.1}$$

where $t$ indicates time, $s(t)$ and $\dot{s}(t)$ are current and past states, $F$ is a system transition, $G(t)$ is input gain, $u(t)$ is an input vector at that instant and $n(t)$ is the process noise [64]. In target tracking problem, input is unknown and the continuous motion model is difficult to realize. Therefore, we are using the following discrete model [65].

$$\mathbf{s}(k) = F\mathbf{s}(k-1) + G(k)a_k \tag{3.2}$$

where $a_k$ is the target acceleration with zero mean and $\sigma_a^2$ is the variance of target acceleration. In target tracking problem, input vector is unknown, we do not know exactly how much force is behind variation of position, velocity and acceleration of the target. The input vector for target tracking is also random as target position, velocity an acceleration can vary at any time instant by unknown applied force. $G(k)$ applies the effect of unknown random acceleration to state vector. The transition matrix $F$ will be explained later.

For simplicity of derivation, one dimensioned motion is assumed then this is extended to two dimensions. $\mathbf{s}$ is a state vector that represents the target current position and velocity in x axis. $\mathbf{s} = [x \quad v_x\ ]^T$ where $x$ is target position and $v_x$ is target velocity. Measurements are updated after $\Delta t$ time as $k$ indicates the measurement update number. From constant linear motion equation [65]

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{3.3}$$

$$G = \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} \tag{3.4}$$

then new model is [16]

$$\mathbf{s}(k) = F\mathbf{s}(k-1) + \nu_{\mathbf{k}} \tag{3.5}$$

where $\nu_{\mathbf{k}}$ is a multivariate Gaussian noise with zero mean as $\nu_{\mathbf{k}} \sim \mathcal{N}(0, Q)$, $Q$ is a covariance matrix based on acceleration variance $\sigma_a^2$ and process noise $\nu_{\mathbf{k}}$, that is [65],

$$Q = \sigma_a^2 G G^T = \sigma_a^2 \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix} \tag{3.6}$$

Now this model is extended to two dimensions where $\mathbf{s}$ is a state vector that represents the target current position and velocity in x and y axis. $\mathbf{s} = [x \;\; v_x \;\; y \;\; v_y]^T$ where $x$, $y$ are target positions and $v_x$ and $v_y$ are target velocities in x and y axis directions accordingly. Measurements are processed after a time interval $T_f$ that is called frame time. The state vector is extended to two dimensions and the transition matrix is [48]

$$F = \begin{bmatrix} 1 & T_f & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

The covariance matrix for two dimensions by assumption of independent noise in every dimension is [48]

$$Q = \sigma_a^2 \begin{bmatrix} T_f^4/4 & T_f^3/2 & 0 & 0 \\ T_f^3/2 & T_f^2 & 0 & 0 \\ 0 & 0 & T_f^4/4 & T_f^3/2 \\ 0 & 0 & T_f^3/2 & T_f^2 \end{bmatrix} \tag{3.8}$$

### 3.1.2 UWB Multistatic Radar Setup (Measurement Model)

A UWB multistatic single transmitter Tx and multiple receivers $\mathrm{Rx}_1$, $\mathrm{Rx}_2$,..., $\mathrm{Rx}_\mathrm{N}$ are considered for single input multiple output (SIMO) nodes radar setup where the transmitter and receiver nodes are placed at different locations. One receiver antenna can be multiplexed with transmitter on the same location where transmitter antenna is located. A central node collects processed data from all nodes for position estimation and target tracking.

An unmodulated first derivative Gaussian mono-cycle pulse is selected for transmission from transmitter as it facilitates the design of amplifiers and antennas. Transmitted pulse $p(t)$ is generated according to the following equation at time $t$ where pulse width is controlled by $\tau_p$ which is called pulse width parameter [66].

$$p(t) = At\ e^{-\frac{t^2}{2\tau_p^2}} \tag{3.9}$$

where $A$ is pulse amplitude and $e$ is an exponential constant.

The transmitter node transmits Gaussian monocycle pulses repeatedly after a time interval that is called pulse repetition time ($PRT$). A frame consists of $N_{prt}$ number of pulses that are processed jointly where $T_f$ is frame time for one frame and $T_f = (N_{prt}) \times (PRT)$ . The system is designed in such way that channel response to transmitted pulses does not change in a frame duration and movement of the target is not affecting channel response during a single frame. Hereafter, terms frame and scan are used alternatively.

Received pulse $r_j$ at $jth$ receiver node without any type of interference is a delayed version of transmitted pulse $p(t - \tau_j)$ and time delay $\tau_j$ can be calculated as [67–69]

$$\tau_j = \frac{d(Tx, Tgt) + d(Rx_j, Tgt)}{c} \tag{3.10}$$

where $d(Tx, Tgt)$ is the distance between the transmitter antenna and the target, $d(Rx_j, Tgt)$ is the distance between $jth$ receiver antenna and target and $c = 3 \times 10^8\ meter/second$ is the speed of light in free space. So this equation can be written as [67–69]

$$\tau_j = \frac{\sqrt{(x_{Tx} - x_{Tgt})^2 + (y_{Tx} - y_{Tgt})^2} + \sqrt{(x_{Rx_j} - x_{Tgt})^2 + (y_{Rx_j} - y_{Tgt})^2}}{c} \tag{3.11}$$

where $x_{Tx}$, $y_{Tx}$ are x axis and y axis position of the transmitter node antenna, $x_{Rx_j}$, $y_{Rx_j}$ are x axis and y axis positions of $jth$ receiver node antennas and $x_{Tgt}$,

$y_{Tgt}$ are x axis and y axis position of the target.

### 3.1.3 Maximum likelihood Position estimation based on Time of Arrival

The following estimation will be used in particle filter and particle swarm optimization algorithms. First, we derive it for a single transmitter and receiver pair. Let $r_j^i$ is received signal at $jth$ receiver where $i$ indicates the pulse number within a frame. For simplicity of derivation, it is assumed that clutter is removed and transmitted pulse is received from direct path as given in the following equation.

$$r_j(t) = a_j p(t - \tau_j) + w_j(t) \tag{3.12}$$

In this equation, $r_j(t)$ is the received pulse at $jth$ receiver, $a_j$ is a channel gain that is established between transmitter to target and target to $jth$ receiver and which can be negative or positive and for simplicity of derivation, it is assumed that channel gain remains constant during one frame. Delayed transmitted pulse scattered by the target is $p(t - \tau_j)$ and $w_j(t)$ is additive white Gaussian noise estimated variance. After sampling of received signal, there are $N$ sampled pulses in each frame of $jth$ receiver and $\mathbf{r^j} = [r_1^j, r_2^j, ..., r_N^j]^T$ represent received pulses in a frame at $jth$ receiver where $\mathbf{p}_{Tx} = [p_1, p_2, ..., p_N]^T$ are cross-ponding transmitted pulses and one received pulse is

$$r_i^j = a_j p_{i-k_j} + w_i^j, \qquad i = 1, 2, 3, ..., N \tag{3.13}$$

$k_j$ is a sampled version of delay as $k_j = \lceil \tau_j / T_s \rceil$ is $Ts$ is sampling interval and $\lceil \; \rceil$

shows rounding to nearest integer.

If target position is known then delay due to target is calculated by equation (3.11). If $p_{i-k_j}$ is known then $a_j$ can be estimated by some estimation technique. Considering the channel gain is estimated by some means, and estimated channel gain is $\hat{a}_j$, each $r_i^j$ has a Gaussian distribution according to equation (3.13) so $r_i^j \sim \mathcal{N}(\hat{a}p_{i-k_j}, \sigma^2)$ where $\sigma^2$ is noise variance. If $p_{i-k_j}$ are known, then the likelihood of received signal given target state $p(r_i^j|\mathbf{s})$ can be expressed as

$$p(r_i^j|\mathbf{s}) \propto e^{-\dfrac{r_i^j - \hat{a}_j p_{i-k_j}}{2\sigma^2}} \tag{3.14}$$

It is seen from equation (3.11) and (3.12) that $k_j$ is a function of state $\mathbf{s}$.

As noise is white, likelihood of received vector $\mathbf{r^j}$ can be expressed as

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{-\dfrac{\Sigma_{i=1}^{N}(r_i^j - \hat{a}_j p_{i-k_j})^2}{2\sigma^2}} \tag{3.15}$$

By maximum likelihood estimation (MLE) technique, the amplitude of channel gain can be estimated by [13]

$$ln(p(\mathbf{r^j}|\mathbf{s})) \propto -\frac{\Sigma_{i=1}^{N}(r_i^j - \hat{a}_j p_{i-k_j})^2}{2\sigma^2} \tag{3.16}$$

Now for MLE, left side is equal to zero and taking the derivative of right side where the proportionality constant can be changed into equality as first term (which is not appearing due to proportionality) derivative is zero with respect to $\hat{a}_j$

$$0 = \frac{\partial}{\partial \hat{a}_j}\left(-\frac{\Sigma_{i=1}^{N}(r_i^j - \hat{a}_j p_{i-k_j})^2}{2\sigma^2}\right) \tag{3.17}$$

$$0 = -\frac{\Sigma_{i=1}^{N}(r_i^j - \hat{a}_j p_{i-k_j})(-p_{i-k_j})}{\sigma^2} \tag{3.18}$$

$$0 = \Sigma_{i=1}^{N}(r_i^j p_{i-k_j} - \hat{a}_j p_{i-k_j}^2) \tag{3.19}$$

$$\Sigma_{i=1}^{N}\hat{a}_j p_{i-k_j}^2 = \Sigma_{i=1}^{N}(r_i^j p_{i-k_j}) \tag{3.20}$$

$$\hat{a}_j = \frac{\Sigma_{i=1}^{N}(r_i^j p_{i-k_j})}{\Sigma_{i=1}^{N} p_{i-k_j}^2} \tag{3.21}$$

As receiver channel gain $\hat{a}_j$ is estimated by MLE approach, expanding expression (3.15)

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{-\Sigma_{i=1}^{N}(r_i^{j2}) - \Sigma_{i=1}^{N}(\hat{a}_j{}^2 p_{i-k_j}^2) + 2\Sigma_{i=1}^{N}(r_i^j \hat{a}_j p_{i-k_j})}{2\sigma^2}} \tag{3.22}$$

As $\Sigma_{i=1}^{N} x = N\bar{x}$ where $\bar{x}$ is mean of $x$ so the expression in (3.22) can be written as

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{-N(\overline{r_i^{j2}}) - \hat{a}_j{}^2 N(\overline{p_{i-k_j}^2}) + 2N(\overline{r_i^j p_{i-k_j}})}{2\sigma^2}} \tag{3.23}$$

Similarly, value of $\hat{a}_j$ from the equation (3.21) is

$$\hat{a}_j = \frac{\overline{(r_i^j p_{i-k_j})}}{(\overline{p_{i-k_j}^2})} \tag{3.24}$$

Now putting value of $\hat{a}_j$ in expression (3.23)

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{-N(\overline{r_i^{j2}}) - N\left(\dfrac{\overline{(r_i^j p_{i-k_j})}}{(\overline{p_{i-k_j}^2})}\right)^2 (\overline{p_{i-k_j}^2}) + 2N(\overline{r_i^j p_{i-k_j}})}{2\sigma^2}} \tag{3.25}$$

Multiplying and dividing expression (3.25) by $(\overline{p_{i-k_j}^2})$

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{-N(\overline{r_i^{j2}})\ (\overline{p_{i-k_j}^2}) - N(\overline{r_i^j p_{i-k_j}})^2 + 2N(\overline{r_i^j p_{i-k_j}})^2}{2\sigma^2(\overline{p_{i-k_j}^2})}} \tag{3.26}$$

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{-N(\overline{r_i^{j2}})}{2\sigma^2} + \dfrac{N(\overline{r_i^j p_{i-k_j}})^2}{2\sigma^2(\overline{p_{i-k_j}^2})}} \tag{3.27}$$

$\dfrac{(r_i^{j2})}{\sigma^2}$ is assumed constant during one frame time so first term can be neglected and expanding second term in expression (3.27)

$$p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{(\Sigma_{i=1}^{N}\ r_i^j p_{i-k_j}\ )^2}{2\sigma^2(\Sigma_{i=1}^{N} p_{i-k_j}^2)}} \tag{3.28}$$

But $p_{i-k_j}^2$ is energy of transmitted pulses in one frame where $\Sigma_{i=1}^{N}\ r_i^j p_{i-k_j}$ is simply obtained by cross correlation of transmitted pulses of one frame by received pulses of the same frame.

Now the expression (3.28) can be generalized for multiple receiver nodes as these receiver nodes are independent and $N_R$ is the number of received antennas so

$$p(\mathbf{r^1}, \mathbf{r^2}, ..., \mathbf{r^{N_R}}|\mathbf{s}) = \prod_{j=1}^{N_R} p(\mathbf{r^j}|\mathbf{s}) \propto e^{\dfrac{\Sigma_{j=1}^{N_R}(\Sigma_{i=1}^{N}\ r_i^j p_{i-k_j}\ )^2}{2\sigma^2(\Sigma_{i=1}^{N} p_{i-k_j}^2)}} \tag{3.29}$$

This relation will be used for weights in particle filter and also for calculation of fitness value in particle swarm optimization.

## 3.2 Particle filter Algorithm

Particle filters are used in applications where a discrete-time state must be estimated that evolves over time and system state is not known directly [44,54,58,70]. Measurements are known at each time instant. These noisy measurements, along with the prior state information can be utilized to determine the hidden system states. Since the measurements are obtained sequentially, particle filters use recursive computation to incorporate all measurements through the current time instant.

Common applications of particle filters include target tracking, wireless channel estimation, time series modeling and many other problems. Important problem areas looked at by researchers are:

- Positioning, where one's own position is to be estimated. This is a filtering problem when the position is not fixed, but changes as a function of time [54].

- Target Tracking, where another object's position is to be estimated based on measurements obtained by sensors [14].

- Blind Detection, where the transmitted signals in flat-fading channels are estimated without the use of pilot signals [71].

- Human Movement Tracking, where full body human motion can be tracked recognized and quantified over a large range of motion [72].

Implementation of a particle filter requires both a time update operation and a measurement update operation. The time update addresses the evolution of the hidden system states over time by predicting the system states from each time instant to the next. The measurement update incorporates the noisy measurements into the estimate of the system state. These two operations compute the conditional distribution of the system state to be estimated. This distribution completely describes the estimated system state.

A hidden Markov model showed in Figure 3.1 which shows observable and unobservable states of a linear dynamic system.

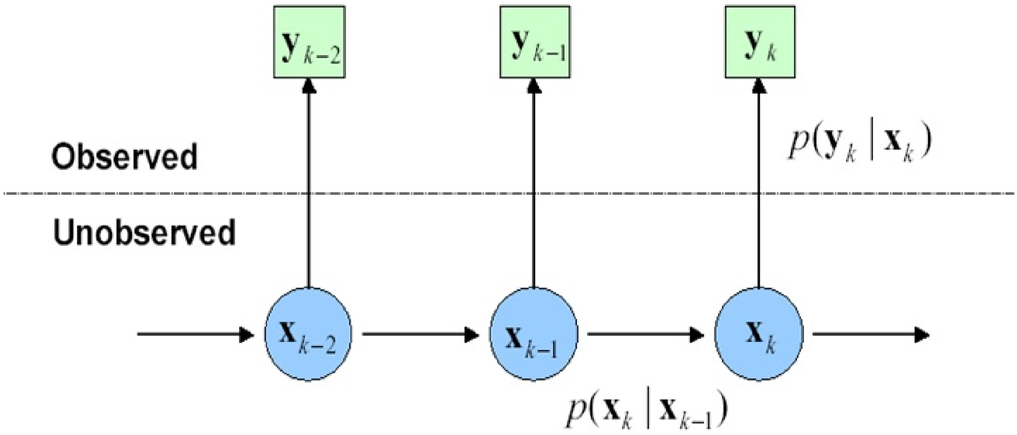

Figure 3.1: Hidden Markov Model

The subscript $k$ denotes the time index, $x^i$ is a $n$ dimensional random state vector; $\mathbf{x_0},\ \mathbf{x_1}, ...,\ \mathbf{x_{k-1}},\ \mathbf{x_k}$ is a Markov sequence of hidden state variables that must be estimated and $\mathbf{y_1},\ \mathbf{y_2}, ...,\ \mathbf{y_{k-1}},\ \mathbf{y_k}$ are the observations related to the corresponding state vector. Let $\mathbf{x_{o:k}}$ denote the sequence of state variables from 0 to $k$ (i.e. $\mathbf{x_0},\ \mathbf{x_1}, ...,\ \mathbf{x_k}$). Similarly, let $\mathbf{y_{1:k}}$ denote the sequence of measurements from 1 to $k$; (i.e. $\mathbf{y_1},\ \mathbf{y_2}, ...,\ \mathbf{y_k}$). The objective of filtering is to compute $\mathbf{p(x_{0:k}|y_{1:k})}$. This distribution is known as the filtering distribution or the posterior distribution and its determination is the primary objective of the Bayesian Filtering algorithm.

We model the time evolution of the state variable by the equation [16]:

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}, \mathbf{w_{k-1}}) \tag{3.30}$$

For simplicity of derivation, $\mathbf{w_{k-1}}$ is a Gaussian process noise $\mathbf{w_k} \sim \mathcal{N}(0, Q)$ where $Q$ is a process noise covariance matrix. The system state at time k is only dependent upon the system state at time $k-1$ and not on any previous system states. This system is therefore a first order Markov process [16]:

$$p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{x_{k-2}}, ..., \mathbf{x_0}) = p(\mathbf{x_k}|\mathbf{x_{k-1}}) \tag{3.31}$$

The measurement model is

$$\mathbf{y_k} = f(\mathbf{x_k}, \nu_\mathbf{k}) \tag{3.32}$$

where $\mathbf{y_k}$ is the measurement obtained at time $k$ and $\nu_\mathbf{k-1}$ is a Gaussian measurement noise $\nu_\mathbf{k} \sim \mathcal{N}(0, R)$ where $R$ is a measurement noise covariance matrix. It is clear from the measurement model that the measurement at time $k$ is only dependent upon the system state at time $k$ and not on any other measurement or state [16]:

$$p(\mathbf{y_k}|\mathbf{x_k}, \mathbf{x_{0:k-1}}, \mathbf{y_{0:k-1}}) = p(\mathbf{y_k}|\mathbf{x_k}) \tag{3.33}$$

Computing the filtering distribution $p(\mathbf{x_{0:k}}|\mathbf{y_{1:k}})$ is not a trivial problem, since it depends on all the states from 0 to $k$. However, if only the current state $\mathbf{x_k}$ is estimated, then to compute the marginal of the filtering distribution $p(\mathbf{x_k}|\mathbf{y_{1:k}})$ one does not need to keep track of all the states. Computation of $p(\mathbf{x_k}|\mathbf{y_{1:k}})$ is done recursively using a two step process of prediction and correction. In the prediction step, given $p(\mathbf{x_{k-1}}|\mathbf{y_{1:k-1}})$, we compute $p(\mathbf{x_k}|\mathbf{y_{1:k-1}})$. In the correction step, we update the predicted density of $\mathbf{x_k}$ with the measurement $\mathbf{y_k}$ to compute $p(\mathbf{x_k}|\mathbf{y_{1:k}})$. This two step procedure is repeated recursively for all the time steps.

### 3.2.1 Prediction Step

The prediction step computes $p(\mathbf{x_k}|\mathbf{y_{1:k-1}})$ from $p(\mathbf{x_{k-1}}|\mathbf{y_{1:k-1}})$. To do this, we need to know Bayes' multiplication theorem [73]:

$$P(A, B|C) = P(B|C).P(A|B, C) \tag{3.34}$$

We compute $p(\mathbf{x_k}|\mathbf{y_{1:k-1}})$ as follows [16]:

$$p(\mathbf{x_k}|\mathbf{y_{1:k-1}}) = \int p(\mathbf{x_k},\ \mathbf{x_{k-1}}|\mathbf{y_{1:k-1}})\, \mathrm{d}\mathbf{x_{k-1}}$$

$$p(\mathbf{x_k}|\mathbf{y_{1:k-1}}) = \int p(\mathbf{x_{k-1}}|\mathbf{y_{1:k-1}})p(\mathbf{x_k}\ |\mathbf{x_{k-1}}, \mathbf{y_{1:k-1}})\, \mathrm{d}\mathbf{x_{k-1}} \tag{3.35}$$

$$p(\mathbf{x_k}|\mathbf{y_{1:k-1}}) = \int p(\mathbf{x_k}|\mathbf{x_{k-1}})p(\mathbf{x_{k-1}}|\mathbf{y_{1:k-1}})\, \mathrm{d}\mathbf{x_{k-1}}$$

Equation (3.35) is known as the Chapman-Kolmogrov Equation [16] and is used to implement the prediction step. The next section will derive the correction step in which $\mathbf{y_k}$ is incorporated into $p(\mathbf{x_k}|\mathbf{y_{1:k-1}})$.

### 3.2.2 Correction Step

The correction step computes $p(\mathbf{x_k}|\mathbf{y_{1:k}})$. This requires the use of Bayes' Theorem [73]:

$$P(A|B,C) = \frac{P(A|C).P(B|A,C)}{P(B|C)} \tag{3.36}$$

Therefore,

$$p(\mathbf{x_k}|\mathbf{y_{1:k}}) = p(\mathbf{x_k}|\mathbf{y_{1:k-1}}, \mathbf{y_k}) \tag{3.37}$$

By using Bayes' theorm in equation (3.37)

$$p(\mathbf{x_k}|\mathbf{y_{1:k-1}}, \mathbf{y_k}) = \frac{p(\mathbf{x_k}|\mathbf{y_{1:k-1}}).p(\mathbf{y_k}|\mathbf{x_k}, \mathbf{y_{1:k-1}})}{p(\mathbf{y_k}|\mathbf{y_{1:k-1}})} \tag{3.38}$$

Now by using Bayes' multiplication theorem in equation (3.38)

$$p(\mathbf{x_k}|\mathbf{y_{1:k-1}}, \mathbf{y_k}) = \frac{p(\mathbf{x_k}|\mathbf{y_{1:k-1}}).p(\mathbf{y_k}|\mathbf{x_k})}{p(\mathbf{y_k}|\mathbf{y_{1:k-1}})} \tag{3.39}$$

In equation (3.39) $p(\mathbf{y_k}|\mathbf{x_k})$ is called the likelihood function, $p(\mathbf{x_k}|\mathbf{y_{1:k-1}})$ is called the prior or the prediction density and $p(\mathbf{y_k}|\mathbf{y_{1:k-1}})$ is called the evidence function.

But the recursive propagation of (3.35) and (3.39) is only conceptual solution because it cannot be determined analytically as it require infinite dimensional vector to store entire probability distribution function (PDF) that is not necessarily Gaussian. Therefore, different approximation methods are used.

A Particle Filter (PF) is an approximation method to solve prediction and correction steps of a Bayesian estimator by Stochastic Monte Carlo numerical integration. As it is out of scope to give the complete derivation of PF filter, so a short mathematical formulation is presented. A detailed and complete mathematical derivation of particle filter can be found in following literature [14,16,44]. The basic idea of particle filter is to use random samples directly from the state space to approximate the posterior distribution $p(\mathbf{x_{0:k}}|\mathbf{y_{1:k}})$ and these random samples are known as particles. Particle Filter considers full posterior distribution $p(\mathbf{x_{0:k-1}}|\mathbf{y_{1:k-1}})$ at time $k-1$ rather than the filtering distribution $p(\mathbf{x_{k-1}}|\mathbf{y_{1:k-1}})$ which is just the marginal of the full posterior distribution with respect to $\mathbf{x}_{k-1}$ then approximate the posterior distribution $p(\mathbf{x_{0:k-1}}|\mathbf{y_{1:k-1}})$ at time $k-1$, with a weighted set of samples $\{\mathbf{x}_{0:k-1}^{i},\ w_{k-1}^{i}\}_{i=1}^{N}$ where $\mathbf{x}_{0:k-1}^{i}$ is random sample and $w_{k-1}^{i}$ is associated normalized weight of particle such that particle weight satisfy $\Sigma_{i=1}^{N}\ w_{k-1}^{i} = 1$. It is very difficult and sometimes impossible to draw random samples from the true posterior. These random samples are drawn using the principle of importance sampling [74]. In importance sampling, one approximates a target distribution $p(x)$, using samples drawn from a proposal distribution $q(x)$ where proposal distribution is an easy to evaluate the distribution and proposal distribution can be factorized in this form $q(\mathbf{x_k}|\mathbf{x^i_{0:k-1}}, \mathbf{y_{1:k}})$. To compensate for the discrepancy between the target and proposal distributions, one has to weight each sample $x^i$ by $w^i \propto \dfrac{\pi(x^i)}{q(x^i)}$ where $\pi(x)$ is a function that is proportional to $p(x)$ and that we know how to evaluate and proposal distribution $q(x)$. Applied to our posterior distribution at the time $k-1$, importance sampling yields:

$$p(\mathbf{x_{0:k-1}}|\mathbf{y_{1:k-1}}) \approx \Sigma_{i=1}^{N}\ w_k^i\delta(\mathbf{x}_{0:k-1} - \mathbf{x}_{0:k-1}^i\ ) \tag{3.40}$$

where $\delta()$ is the Dirac function. These particles are recursively updated based on their previous weight to obtain an approximation of the posterior distribution $p(\mathbf{x_{0:k}}|\mathbf{y_{1:k}})$ at the next time step $k$ each particles weight is set to

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{y_k}|\mathbf{x_k^i}).p(\mathbf{x_k^i}|\mathbf{x_{k-1}^i})}{q(\mathbf{x_k}|\mathbf{x_{0:k-1}^i}, \mathbf{y_{1:k}})} \tag{3.41}$$

Finally, the posterior probability distribution is approximated as

$$p(\mathbf{x_{0:k}}|\mathbf{y_{1:k}}) \approx \Sigma_{i=1}^{N}\ w_k^i\delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i\ ) \tag{3.42}$$

### 3.2.3 Degeneracy problem

A major problem with particle filtering is degeneracy where the discrete random measure degenerates quickly [16]. Therefore, after a few iterations, very few particles can contribute to the estimation. These particles with negligible weights have almost no contribution in the estimation of the posterior distribution. Therefore, carrying them forward is simply a waste of computational power. The performance of the particle filter will deteriorate because of this degeneracy.

### 3.2.4 Resampling

One common way to deal with degeneracy is resampling. In resampling, one draws (with replacement) a new set of N particles from the discrete approximation to the filtering distribution $p(\mathbf{x_k}|\mathbf{y_{1:k}})$ (or to the full posterior distribution, if one is interested in computing the full posterior) provided by the weighted particles according to equation (3.42).

A complete PF algorithm is presented by the flow chart in figure 3.2.

### 3.2.5 Resampling impoverishment problem

Since resampling is done with replacement, a particle with a large weight is likely to be drawn multiple times and conversely particles with very small weights are not likely to be drawn at all. Also, note that the weights of the new particles will all be equal to $1/N$. Thus, resampling effectively deals with the degeneracy problem by getting rid of the particles with very small weights. This, however, introduces a new problem called the sample impoverishment problem. Because particles with large weights are likely to be drawn multiple times during resampling, whereas particles with small weights are not likely to be drawn at all, the diversity of the particles will tend to decrease after a resampling step. In the extreme case, all particles might collapse into a single particle. This will negatively impact the quality of the approximation, as one will be trying to represent a whole distribution with a fewer number of distinct samples (in the case of a single particle, one will be representing the full distribution with a single point estimate).

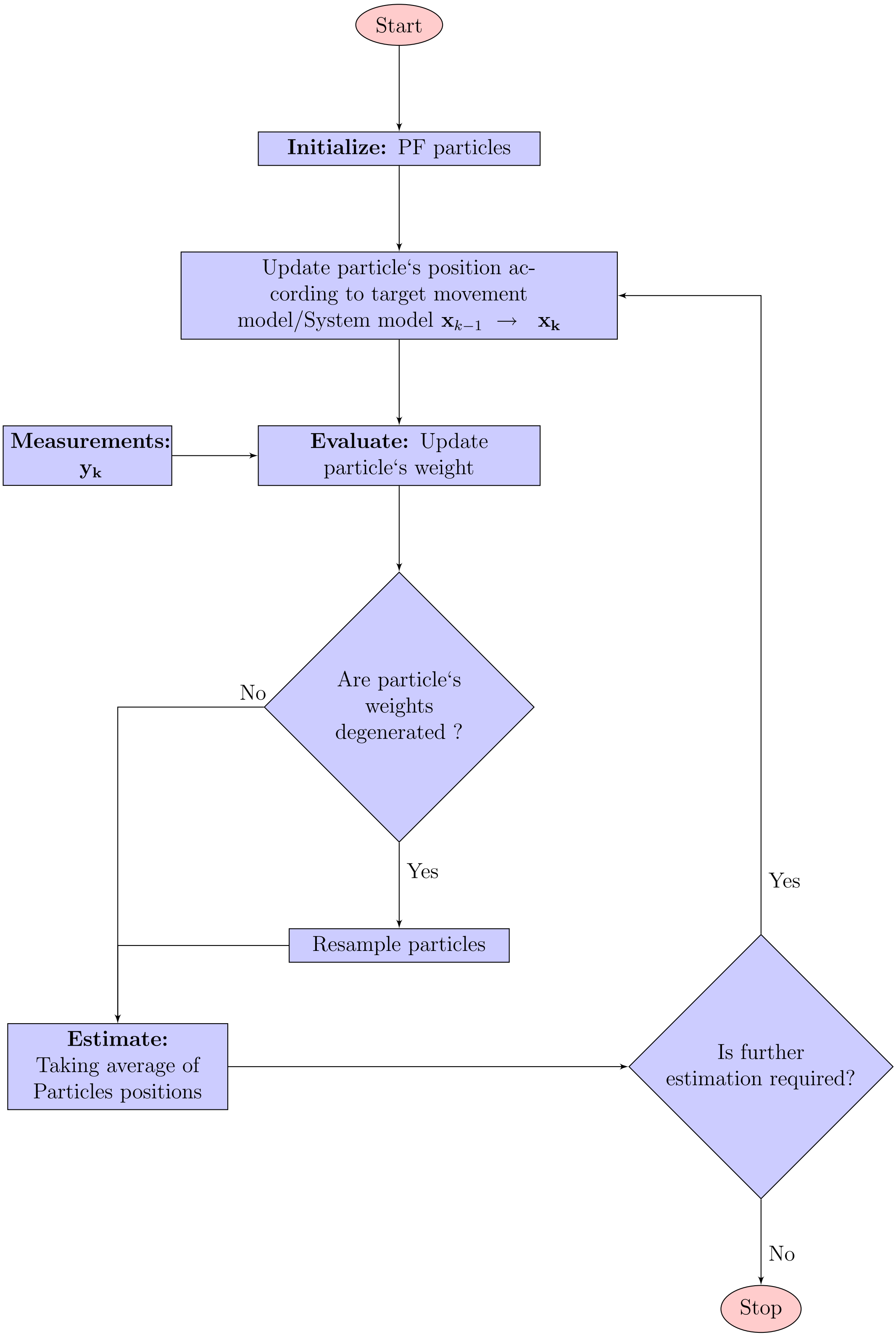
Start
**Initialize:** PF particles
Update particle‘s position according to target movement model/System model $\mathbf{x}_{k-1} \rightarrow \mathbf{x_k}$
**Measurements:** $\mathbf{y_k}$
**Evaluate:** Update particle‘s weight
Are particle‘s weights degenerated ?
No
Yes
Resample particles
**Estimate:** Taking average of Particles positions
Is further estimation required?
Yes
No
Stop

Figure 3.2: PF algorithm

## 3.3 PARTICLE SWARM OPTIMIZATION

A detail discussion of particle swarm optimization is given in literature [20]. So here we discuss the only functionality of particle swarm optimization.

### 3.3.1 Functionality of PSO algorithm

First, a swarm is formulated and it consists of particles. This swarm can move in predefined search space. A particle represents a point in multidimensional space. Particle position of each particle represents a potential solution in the search space. Particle swarm optimization algorithm finds the optimal solution by probabilistic and recursive modification of the these available solutions.

Let a swarm consists of $N_{PSO}$ particles in two dimensional Cartesian plane. Particle position is represented by $\mathbf{p_i} = [x_i, y_i]^T$ where $x_i$ is x-axis position and $y_i$ is in y-axis position of $i_{th}$ particle. Each particle is moving with a velocity $\mathbf{v}_i = [v_{x_i}, v_{y_i}]^T$ where $v_{x_i}$ is x-axis velocity and $v_{y_i}$ is in y-axis of $i_{th}$ particle. A fitness function is defined which assigned values to each particle based on measurements. Every particle in the swarm is moving randomly by some controlled parameter which we will discuss later.

After every movement of particle, particle evaluates its fitness value with the previous best value that particle has attained until now. If current fitness value is better then it is updated in record with corresponding position of particle as $\mathbf{p}_{i,b} = [x_{i,b}, y_{i,b}]^T$ where $\mathbf{p}_{i,b}$ is particle best position, $x_{i,b}$ is x-axis position, $y_{i,b}$ is y axis position of particle.

A global best value is attained by comparing all particles fitness values and selecting the best value in between them and comparing it to previous achieved global best value if previous global best value is less than current global best value then the global best value is updated in record with its position and this position is a global best position as $\mathbf{g}_b = [x_{gb}, y_{gb}]^T$ where $\mathbf{g}_b$ is the global best position, $x_{gb}$ is the x-axis position, $y_{gb}$ is the y axis position of the global best particle. The movement of particles is controlled by three control parameter, that are inertia weight($w_i$) , local acceleration constant($c_l$) and global acceleration constant($c_g$). Inertia weight($w_i$) is used to control the contribution of past particle velocity in current particle velocity, local acceleration constant($c_l$) is used to control velocity contribution due to the difference between current particle position and particle best position and global acceleration constant($c_g$) is used to control velocity contribution due to the difference between current particle position and global best position.

PSO algorithm steps are following:

Initialize swarm size, particles velocities, particles positions, inertia weight, local acceleration constant, global acceleration constant, particle minimum position limit maximum position limit, particle minimum velocity limit, particle maximum velocity limit, iteration number and also defines stopping criteria of PSO algorithm.

Evaluate the fitness function for every particle and update its fitness value, particle best position and global best position by comparing with previous values.

Update particle velocity by following equation

$$\mathbf{v}_i^{k+1} = w_i \mathbf{v}_i^k + c_l r_1 (\mathbf{p}_{i,b} - \mathbf{p}_i^k) + c_g r_2 (\mathbf{g}_b - \mathbf{p}_i^k) \tag{3.43}$$

where $k$ indicates iteration number, $r_1$ and $r_2$ are uniform distributed random variables.

If particle velocity crossed maximum or minimum limit assign them maximum or minim velocity limit value accordingly.

Update particle position by following equation

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \mathbf{v}_i^{k+1} \circ \mathbf{p}_i^k \tag{3.44}$$

here $\circ$ presents a Hadamard product. If the particle position crossed maximum or minimum limit, assign them maximum or minim position limit value accordingly. Check if stop criteria is met or not, if yes then stop and global best is problem solution otherwise repeats all steps from calculating fitness values until stopping criteria met.

### 3.3.2 PSO bias and slow convergence issues

Standard PSO has a bias issue. PSO particles have a clear bias in their movement direction that depends on the direction of the coordinate axis [22–24]. Due to this biased movement, sometime optimal solution is missed by PSO mainly in noisy environment. Standard PSO also has slow convergence issues, if the convergence of PSO is improved, it also causes an increase of steady state error [75].

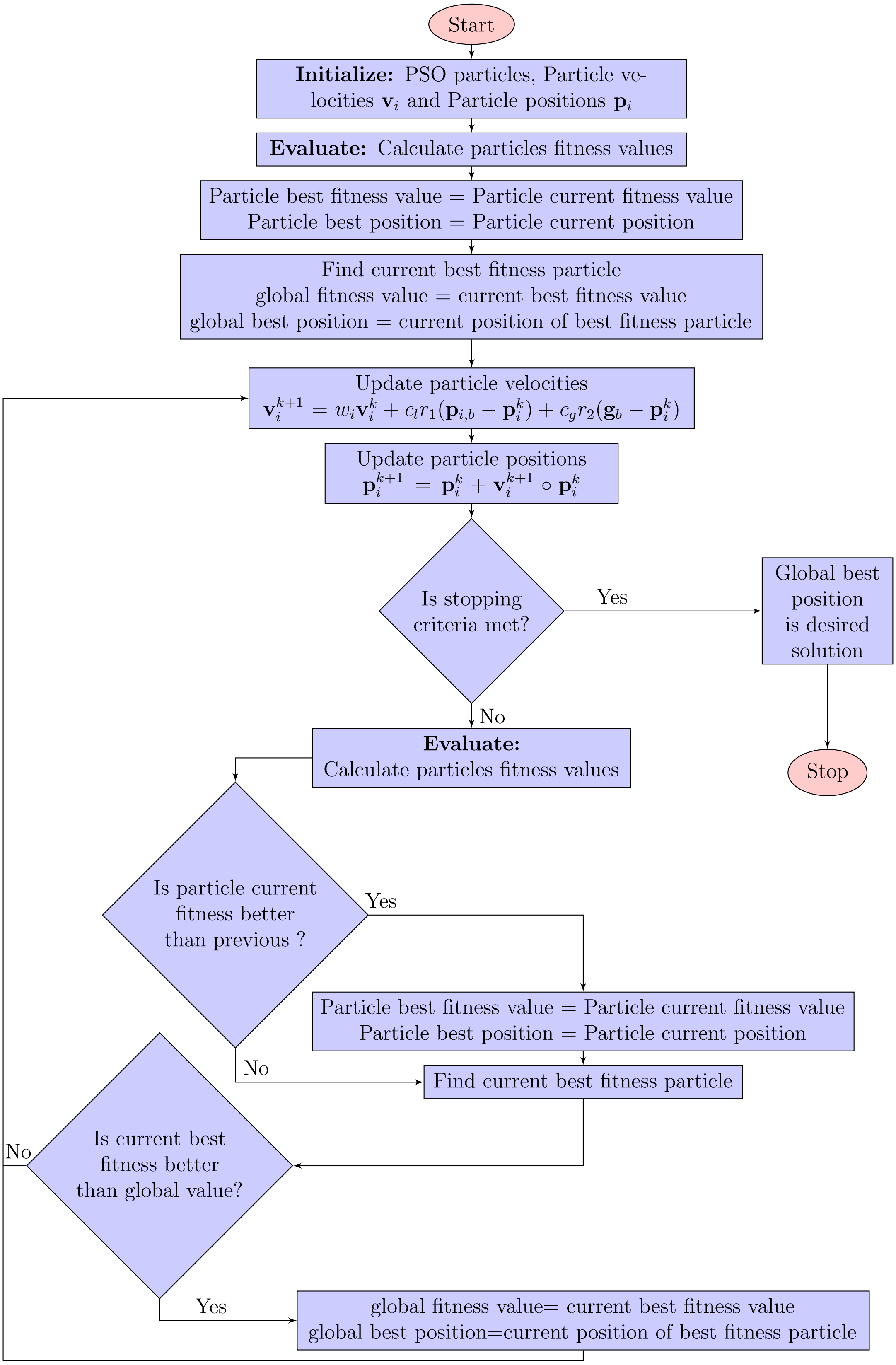
Start
**Initialize:** PSO particles, Particle velocities $\mathbf{v}_i$ and Particle positions $\mathbf{p}_i$
**Evaluate:** Calculate particles fitness values
Particle best fitness value = Particle current fitness value
Particle best position = Particle current position
Find current best fitness particle
global fitness value = current best fitness value
global best position = current position of best fitness particle
Update particle velocities
$\mathbf{v}_i^{k+1} = w_i \mathbf{v}_i^k + c_l r_1(\mathbf{p}_{i,b} - \mathbf{p}_i^k) + c_g r_2(\mathbf{g}_b - \mathbf{p}_i^k)$
Update particle positions
$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \mathbf{v}_i^{k+1} \circ \mathbf{p}_i^k$
Is stopping criteria met?
Yes
Global best position is desired solution
Stop
No
**Evaluate:**
Calculate particles fitness values
Is particle current fitness better than previous ?
Yes
Particle best fitness value = Particle current fitness value
Particle best position = Particle current position
No
Find current best fitness particle
Is current best fitness better than global value?
No
Yes
global fitness value= current best fitness value
global best position=current position of best fitness particle

Figure 3.3: PSO Algorithm

## CHAPTER 4

# PROPOSED ALGORITHMS

In this work, two novel UWB multi-static radar target tracking approaches based on Particle Filter embedded Particle Swarm Optimization are proposed.

## 4.1 AW-PSO-Particle Filter

A novel UWB multi-static radar target tracking approach based on Particle filter embedded Particle swarm optimization is proposed. PF has a sample impoverishment problem due to loss of diversity of particles after re-sampling step. The divergence of PF Particles is maintained by adding PSO before re- sampling stage in PF. PSO has inherit the bias problem due to particles movement towards their coordinate axis and this is addressed by PF, as PF maintains a small sample of the prior probability distribution to mitigate biasing effects of PSO. The slow convergence in PSO is tackled using adaptive inertia weights. The proposed algorithm not only handles above mentioned problems, but also improves target tracking accuracy and robustness at the cost of minor additional computational complexity.

We named this algorithm as a new algorithm is Adaptive inertia Weight Particle Swarm Optimization Particle Filter.(AW-PSO-PF).

The sample impoverishment problem arises as PF particles are changed or updated based on higher weight particles. The contribution from low weight particles to estimate the prior probability distribution is reduced in resampling stage of PF also results in poor particles diversity. This will degrade the quality of estimation in particle filter. This problem can be addressed if we estimate the current probability distribution instead of prior probability distribution. It will improve estimation accuracy and eliminate the sample impoverishment problem.

PSO particles are used to estimate the current probability distribution to be used in PF for target tracking. The convergence time of PSO algorithm depends on speed of particles towards an optimal solution. The speed of PSO particles can be controlled by following three factors:

1. Inertia weight
2. Difference between current particle position and best particle position.
3. Difference between current particle position and global best particle position.

If particle best position and global best position are also optimal/near-optimal solution, increase in speed of particles towards that position increases the convergence rate. For the case when particle best position/global best position are not an optimal solution, increase in speed of particles towards that position results in decrease of convergence rate. If we control particles speed such that it also dependents on particle fitness variation due to particle movements as:

1. If particle fitness value increases, i.e. particle is moving towards optimal /near-optimal solution and hence particle speed will be increased.

2. decrease in particle fitness value means particle is not moving towards optimal/near optimal solution so particle speed will be decreased.

3. If the particle fitness value does not change, its inertia weight is calculated based on current fitness value.

The above stated mechanism can be implemented if we set the inertia weight of each particle according to the difference of particle fitness value in each iteration. The inertia weight is increased if particle current fitness value is better than particle previous fitness value. If the particle current fitness value is less than particle previous fitness value, the inertia weight is decreased. The inertia weight is adjusted adaptively according to following equation

$$\omega^i(n) = \frac{s(1 - e^{(J^i_{norm}(n))})}{(1 + e^{-\Delta J^i_{norm}(n)})} \tag{4.1}$$

where

$$\Delta J^i_{norm}(n) = J^i_{norm}(n-1) - J^i_{norm}(n) \tag{4.2}$$

Here, $\omega^i(n)$ is the current inertia weight of $i_{th}$ PSO particle, $n$ is a PSO iteration number, $s$ is control parameter use to control the range of inertia weights, $\Delta J^i_{normi}(n)$ is the difference between particle current fitness value, $J^i_{norm}(n)$ , and particle previous fitness value, $J^i_{norm}(n-1)$ normalized from 0 to 1.

After assigning inertia weights to PSO particles, the conventional PSO is modified with these adaptive inertia weights and before its convergence we use these PSO particles with PF particles for target estimation. The block diagram of proposed algorithm is shown in figure 4.1. For simplicity and readability list of parameters used are described in table 4.1.

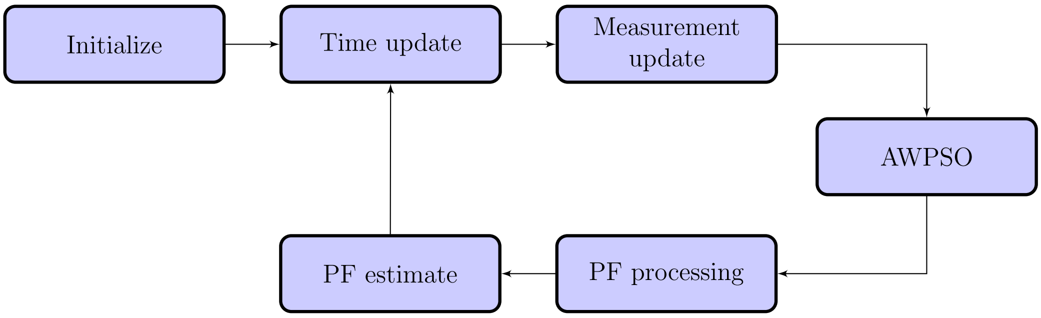

Figure 4.1: AWPSOPF Block diagram

The description of each block is as follows:

### 4.1.1 Initialization

First of all total number of PF and PSO particles are initialized. We assumed that initial target position is known. The PF particles are initialized as $\mathbf{s}_{PF}^{j} = \mathbf{s}_{tgt}$ where $j$ is a PF particle index and $j \in \{1, 2, ..., N_{PF}\}$. Values for $c_{l_{PSO}}$, $c_{g_{PSO}}$, $x_{min_{pso}}$, $x_{max_{pso}}$, $v_{min_{pso}}$, $v_{max_{pso}}$, PSO iteration limit PF resampling limit and PF particles weight threshold are initialized.

### 4.1.2 Time update

Target is moved according to following dynamic model:

$$\mathbf{s}_{tgt}^{k} = \mathbf{F}\mathbf{s}_{tgt}^{k-1} \tag{4.3}$$

here $k$ is current trajectory index and $\mathbf{F}$ is target movement transition matrix

Table 4.1: List of AW-PSO-PF parameters

| **Parameter symbol** | **Description** |
|---|---|
| $x_{tgt}$ | Target x axis position |
| $y_{tgt}$ | Target y axis position |
| $\mathbf{p}_{tgt}$ | Target position state vector<br>$\mathbf{p}_{tgt} = [x_{tgt}\ \ y_{tgt}]^T$ |
| $v_{x_{tgt}}$ | Target x axis velocity |
| $v_{y_{tgt}}$ | Target y axis velocity |
| $\mathbf{s}_{tgt}$ | Target state vector<br>$\mathbf{s}_{tgt} = [x_{tgt}\ \ v_{x_{tgt}}\ \ y_{tgt}\ \ v_{y_{tgt}}]^T$ |
| $x_{pso}$ | PSO particle x axis position |
| $y_{pso}$ | PSO particle y axis position |
| $\mathbf{p}_{pso}$ | PSO particle position vector<br>$\mathbf{p}_{pso} = [x_{pso}\ y_{pso}]^T$ |
| $vx_{pso}$ | PSO particle x axis velocity |
| $vy_{pso}$ | PSO particle y axis velocity |
| $\mathbf{v}_{pso}$ | PSO particle velocity vector<br>$\mathbf{v}_{pso} = [vx_{pso}\ y_{pso}]^T$ |
| $\omega_{PSO}$ | PSO particle inertia weight |
| $J_{PSO}$ | PSO fitness value |
| $N_{PSO}$ | Total PSO particles |
| $x_{Pb_{pso}}$ | PSO particle x axis best position |
| $y_{Pb_{pso}}$ | PSO particle y axis best position |
| $\mathbf{p}_{Pb_{pso}}$ | PSO particle best position vector<br>$\mathbf{p}_{Pb_{pso}} = [x_{Pb_{pso}}, y_{Pb_{pso}}]^T$ |
| $x_{gb_{pso}}$ | PSO x axis global best position |
| $y_{gb_{pso}}$ | PSO y axis global best position |
| $\mathbf{p}_{gb_{pso}}$ | PSO global position vector<br>$\mathbf{p}_{gb_{pso}} = [x_{gb_{pso}}, y_{gb_{pso}}]^T$ |
| $c_{l_{PSO}}$ | PSO local acceleration constant |
| $c_{g_{PSO}}$ | PSO global acceleration constant |
| $x_{min_{pso}}$ | min. limit of PSO particles positions in any axis |
| $x_{max_{pso}}$ | max. limit of PSO particles positions in any axis |
| $v_{min_{pso}}$ | min. limit of PSO particles velocity in any axis |
| $v_{max_{pso}}$ | max. limit of PSO particles velocity in any axis |
| $x_{PF}$ | PF particle x axis position |
| $y_{PF}$ | PF particle y axis position |
| $v_{x_{PF}}$ | PF particle x axis velocity |
| $v_{y_{PF}}$ | PF particle y axis velocity |
| $\mathbf{s}_{PF}$ | PF particle state vector<br>$\mathbf{s}_{PF} = [x_{PF}\ v_{x_{PF}}\ \ y_{PF}\ \ v_{y_{PF}}]^T$ |
| $N_{PF}$ | Total PF particles |

defined as

$$F = \begin{bmatrix} 1 & T_f & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

$T_F$ is frame time after which measurements are processed.

PF particles are updated according to following equation.

$$\mathbf{s}_{PF}^{k,j} = \mathbf{F}\mathbf{s}_{PF}^{k-1,j} + \nu_{\mathbf{k}} \tag{4.5}$$

$\nu_{\mathbf{k}}$ is a multivariate Gaussian noise, $\nu_{\mathbf{k}} \sim \mathcal{N}(0, Q)$ with zero mean and covariance $Q$. The covariance matrix used is as follows

$$Q = \sigma_a^2 \begin{bmatrix} T_f^4/4 & T_f^3/2 & 0 & 0 \\ T_f^3/2 & T_f^2 & 0 & 0 \\ 0 & 0 & T_f^4/4 & T_f^3/2 \\ 0 & 0 & T_f^3/2 & T_f^2 \end{bmatrix} \tag{4.6}$$

where $\sigma_a^2$ is target acceleration variance.

The positions of PSO particles are initialized as $\mathbf{p}_{pso}^i \sim \mathcal{U}(\mathbf{p}_{tgt}^k, \mathbf{Q}_{PSO}^P)$ where $i$ is PSO particle index, $i \in \{1, 2, ..., N_{PSO}\}$, $\mathcal{U}$ is unform distribution, $\mathbf{Q}_{PSO}^P = (x_{max_{pso}} - x_{min_{pso}})\mathbf{I}$, $\mathbf{I}$ is identity matrix and $\mathbf{v}_{pso}^i \sim \mathcal{U}(0, \mathbf{Q}_{PSO}^v)$, $\mathbf{Q}_{PSO}^v = (v_{max_{pso}} - v_{min_{pso}})\mathbf{I}$. $x_{min_{pso}}$ is updated as $x_{min_{pso}} = x_{tgt}^k - x_{min_{pso}}$, $x_{max_{pso}} = x_{tgt}^k + x_{max_{pso}}$. PSO particles best position is $\mathbf{p}_{Pb_{pso}}^i = \mathbf{p}_{pso}^i$.

### 4.1.3 Measurement update

Unmodulated first derivative Gaussian mono-cycle pulses $p(t)$ are transmitted by the multi static radar transmitter and multiple receivers wait for pulse repetition time to receive reflected pulses. PF weight and PSO fitness values are assigned by following equation(already discussed in 3.1.3)

$$w_{PF} = J_{PSO} = e^{\frac{\Sigma_{jR=1}^{N_R}(\Sigma_{ip=1}^{N} r_{ip}^{jR} p_{ip-k_{jR}})^2}{2\sigma^2(\Sigma_{ip=1}^{N} p_{ip-k_{jR}}^2)}} \tag{4.7}$$

where $N$ is total pulses in one frame. $ip$ is the pulse number in the frame, $j_R$ is receiver number, $N_R$ is number of total receivers, $p_{ip-k_jR}$ is delayed pulse at $jR_{th}$ receiver, $k_{jR}$ is delay at $jR_{th}$ receiver, $r_{ip}^{jR}$ is received pulse at $jR_{th}$ receiver and $\sigma^2$ is process noise.

All PF and PSO particles are updated using equation (4.7). The inertia weights $iw_{PSO}^{i}$ are assigned according to equation (4.1). $\mathbf{p}_{gb_{pso}}$ is updated by comparing fitness values of all PSO particles. The particle with maximum fitness value, its position is assigned as $\mathbf{p}_{gb_{pso}}$.

### 4.1.4 Adaptive inertia Weight Particle Swarm Optimization

The steps of Adaptive inertia Weight Particle Swarm Optimization (AWPSO) algorithm are as follows:

1. PSO particles velocities are updated using following equation

$$\mathbf{v}_{pso}^{i,n+1} = \omega_{PSO}^{i}\mathbf{v}_{pso}^{i,n} + c_{l_{PSO}} r_1(\mathbf{p}_{Pb_{pso}}^{i} - \mathbf{p}_{pso}^{i}) + c_{g_{PSO}} r_2(\mathbf{p}_{gb_{pso}} - \mathbf{p}_{pso}^{i}) \tag{4.8}$$

$r_1$ and $r_2$ are uniform distributed random variables.

2. The maximum and minimum limits of particles velocities are checked and if they exceed the specified limit, it is set to that limit.
3. The positions of PSO particles are updated using following equation

$$\mathbf{p}_{pso}^{i,n+1} = \mathbf{p}_{pso}^{i,n} + T_F \ \mathbf{v}_{pso}^{i,n+1} \tag{4.9}$$

4. The maximum and minimum limits of particles positions are checked and if they exceed the specified limit, it is set to that limit.
5. The new fitness values of PSO particles are calculated using the equation (4.7) and new inertia weights, particle best position and global best position are updated.
6. The condition for stopping criterion is checked, stop if it is met, otherwise repeat from step 1 of AWPSO.

### 4.1.5 Particle Filter Processing

PF processing steps are following.

1. First, PSO particles best fitness values are compared to PF particles weights, if PSO particles best fitness value is higher than PF weights, their positions and particle best fitness values are assigned to PF particle positions and weights respectively. Otherwise PF particles maintains their positions and weights.

2. In this step, the PF particle weight and resampling limit are checked against the set threshold and if the condition is met, we forward these particles to the estimation stage, otherwise PF particles are resampled using equation (4.5) and repeat the same step.

### 4.1.6 Particle Filter estimation Stage

The target position is estimated using following equation.

$$\hat{\mathbf{p}}_{\mathbf{tgt}}^{k} = \frac{\Sigma_{j=1}^{N_{PF}}(\mathbf{s}_{PF}^{k,j})}{N_{PF}} \tag{4.10}$$

If further target tracking is required, the whole process is started from time update step excluding the initialization step.

## 4.2 Distributed Particle swarm optimization Particle Filter

Our second proposed algorithm is an enhanced version of the AWPSOPF algorithm to further improve the particles diversity and tracking robustness. The social interaction of PSO particles is increased by dividing them in small groups. It makes the algorithm more robust. We named this algorithm as Distributed Particle Swarm Optimization Particle Filter (DPSOPF).

The conventional PSO algorithm relies on global best particle and PSO particles tends to move towards that global best particle position. It is possible that particles will not explore the search space significantly, due to the reason that if in

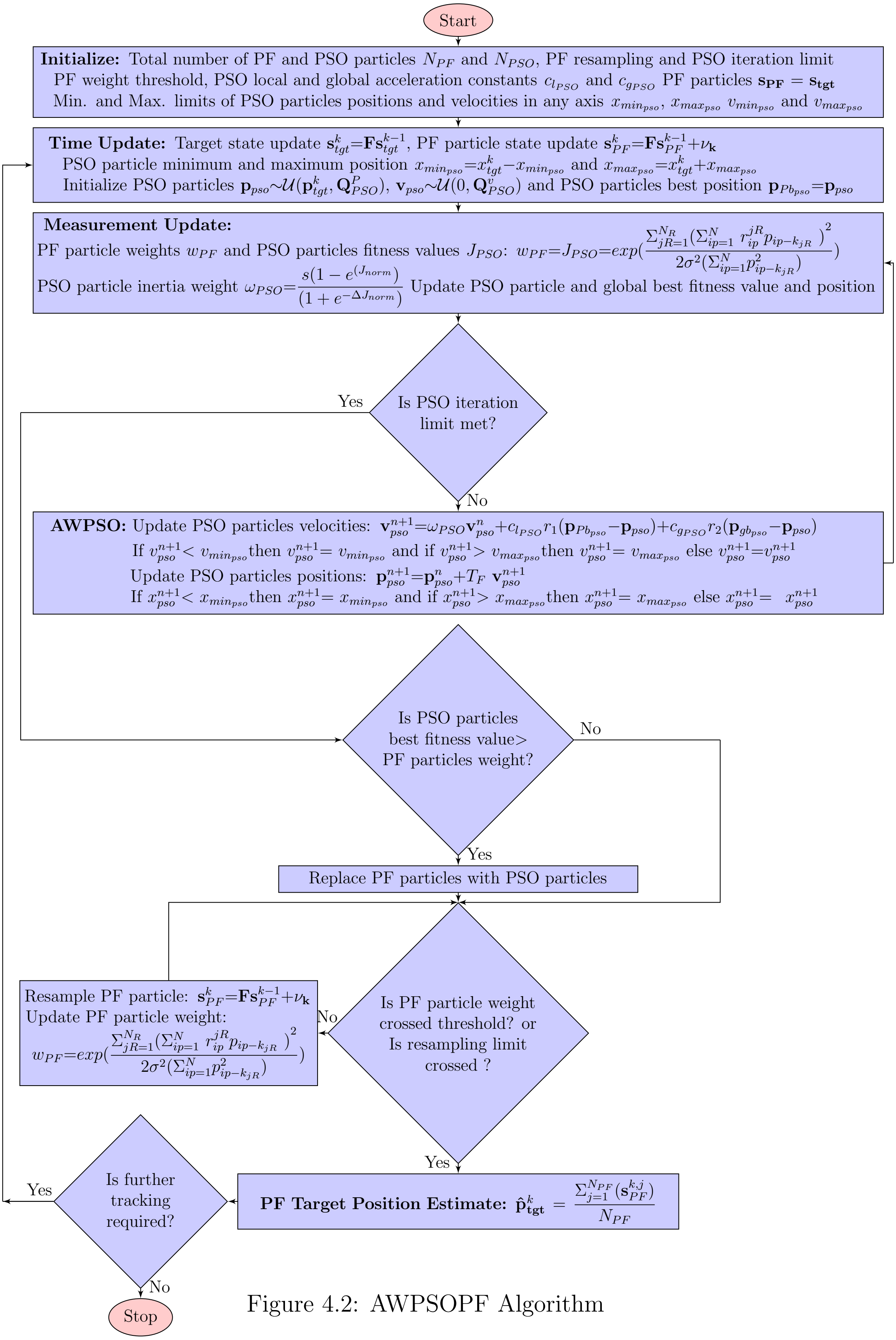
Start
**Initialize:** Total number of PF and PSO particles $N_{PF}$ and $N_{PSO}$, PF resampling and PSO iteration limit
PF weight threshold, PSO local and global acceleration constants $c_{l_{PSO}}$ and $c_{g_{PSO}}$ PF particles $\mathbf{s_{PF}} = \mathbf{s_{tgt}}$
Min. and Max. limits of PSO particles positions and velocities in any axis $x_{min_{pso}}$, $x_{max_{pso}}$ $v_{min_{pso}}$ and $v_{max_{pso}}$
**Time Update:** Target state update $\mathbf{s}^k_{tgt}{=}\mathbf{F}\mathbf{s}^{k-1}_{tgt}$, PF particle state update $\mathbf{s}^k_{PF}{=}\mathbf{F}\mathbf{s}^{k-1}_{PF}{+}\nu_{\mathbf{k}}$
PSO particle minimum and maximum position $x_{min_{pso}}{=}x^k_{tgt}{-}x_{min_{pso}}$ and $x_{max_{pso}}{=}x^k_{tgt}{+}x_{max_{pso}}$
Initialize PSO particles $\mathbf{p}_{pso}{\sim}\mathcal{U}(\mathbf{p}^k_{tgt}, \mathbf{Q}^P_{PSO})$, $\mathbf{v}_{pso}{\sim}\mathcal{U}(0, \mathbf{Q}^v_{PSO})$ and PSO particles best position $\mathbf{p}_{Pb_{pso}}{=}\mathbf{p}_{pso}$
**Measurement Update:**
PF particle weights $w_{PF}$ and PSO particles fitness values $J_{PSO}$: $w_{PF}{=}J_{PSO}{=}exp(\frac{\Sigma^{N_R}_{jR=1}(\Sigma^{N}_{ip=1}\ r^{jR}_{ip}p_{ip-k_{jR}}\ )^2}{2\sigma^2(\Sigma^N_{ip=1}p^2_{ip-k_{jR}})})$
PSO particle inertia weight $\omega_{PSO}{=}\frac{s(1-e^{(J_{norm})})}{(1+e^{-\Delta J_{norm}})}$ Update PSO particle and global best fitness value and position
Yes
Is PSO iteration limit met?
No
**AWPSO:** Update PSO particles velocities: $\mathbf{v}^{n+1}_{pso}{=}\omega_{PSO}\mathbf{v}^n_{pso}{+}c_{l_{PSO}}r_1(\mathbf{p}_{Pb_{pso}}{-}\mathbf{p}_{pso}){+}c_{g_{PSO}}r_2(\mathbf{p}_{gb_{pso}}{-}\mathbf{p}_{pso})$
If $v^{n+1}_{pso}{<}\ v_{min_{pso}}$then $v^{n+1}_{pso}{=}\ v_{min_{pso}}$ and if $v^{n+1}_{pso}{>}\ v_{max_{pso}}$then $v^{n+1}_{pso}{=}\ v_{max_{pso}}$ else $v^{n+1}_{pso}{=}v^{n+1}_{pso}$
Update PSO particles positions: $\mathbf{p}^{n+1}_{pso}{=}\mathbf{p}^n_{pso}{+}T_F\ \mathbf{v}^{n+1}_{pso}$
If $x^{n+1}_{pso}{<}\ x_{min_{pso}}$then $x^{n+1}_{pso}{=}\ x_{min_{pso}}$ and if $x^{n+1}_{pso}{>}\ x_{max_{pso}}$then $x^{n+1}_{pso}{=}\ x_{max_{pso}}$ else $x^{n+1}_{pso}{=}\ \ x^{n+1}_{pso}$
Is PSO particles best fitness value> PF particles weight?
No
Yes
Replace PF particles with PSO particles
Resample PF particle: $\mathbf{s}^k_{PF}{=}\mathbf{F}\mathbf{s}^{k-1}_{PF}{+}\nu_{\mathbf{k}}$
Update PF particle weight:
$w_{PF}{=}exp(\frac{\Sigma^{N_R}_{jR=1}(\Sigma^{N}_{ip=1}\ r^{jR}_{ip}p_{ip-k_{jR}}\ )^2}{2\sigma^2(\Sigma^N_{ip=1}p^2_{ip-k_{jR}})})$
No
Is PF particle weight crossed threshold? or Is resampling limit crossed ?
Yes
**PF Target Position Estimate:** $\hat{\mathbf{p}}^k_{\mathbf{tgt}} = \frac{\Sigma^{N_{PF}}_{j=1}(\mathbf{s}^{k,j}_{PF})}{N_{PF}}$
Yes
Is further tracking required?
No
Stop

Figure 4.2: AWPSOPF Algorithm

the global best is a local optimal solution in the beginning, there is a chance of misleading and consequently the particles miss the actual optimal position.

In this algorithm, the grouping of particles is based on their neighbors at minimum distance and local best position for each group is introduced. Now each PSO particle also holds its group information in addition to particle position, particle velocities, particle fitness value, particle best fitness value and particle best position. A variable variance parameter is also introduced in PSO velocity equation to control the influence of local best positions and it is defined as:

$$V_{PSO}^{n} = \frac{-A_{max}}{\left(1 + e^{(n - V_{mid})/S}\right)} + A_{max} \tag{4.11}$$

where $A_{max}$ is used to control the value for $V_{PSO}$, $n$ is a PSO iteration number, $V_{mid}$ is to indicate mid point of variance parameter from where the influence of local best is reduces compare to global best and $S$ is to control the decreasing speed of variance. The local best particles within the group are used to estimate current probability distribution function for particle filter along with previous prior probability distribution to improve the performance. Since initialization, time update, measurement update and PF estimation steps remain same as our previously proposed AWPSOPF algorithm. Here in DPSOPF proposed algorithm, AWPSO step is replaced by DPSO with slight modification in PF processing stage of AWPSOPF algorithm, are shown in figure 4.3 and discussed as follows:

### 4.2.1 Distributed Particle Swarm Optimization

1. The PSO particles are grouped based on minimum distance neighborhood. The group best position, $\mathbf{p}_{grpb_{pso}}^{ng}$, is calculated based on the best particle

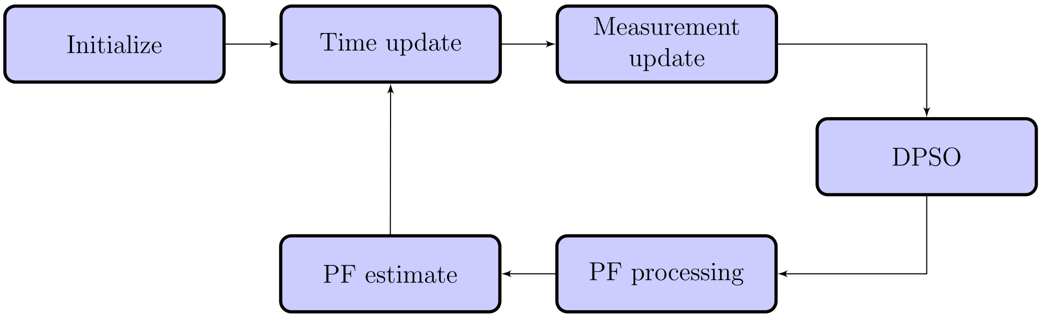

Figure 4.3: DPSOPF Block diagram

fitness value within a group, $\mathbf{p}^{ng}_{grpb_{pso}} = [x^{ng}_{grpb_{pso}}, y^{ng}_{grpb_{pso}}]^T$ where $x^{ng}_{grpb_{pso}}$ and $y^{ng}_{grpb_{pso}}$ are x, y axis positions accordingly and $n_g$ is a particle group number.

2. Since the group best position is different, the PSO particle velocity equation is updated in groups as follows:

$$\begin{aligned}\mathbf{v}^{i,n+1}_{pso} = \omega^{i}_{PSO}\mathbf{v}^{i,n}_{pso} + c_{l_{PSO}} r_1(\mathbf{p}^{i}_{Pb_{pso}} - \mathbf{p}^{i}_{pso}) + c_{p_{PSO}} r_2(\mathbf{p}_{gb_{pso}} - \mathbf{p}^{i}_{pso}) \\ +V^{n}_{PSO}\ r_3(\mathbf{p}^{ng}_{grpb_{pso}} - \mathbf{p}^{i}_{pso}))\end{aligned} \tag{4.12}$$

   where $r_3$ is a uniform distributed random variables. The details of other variables are already discussed before the preceding sections.

3. The maximum and minimum limits of particles velocities are checked and if they exceed the specified limit, it is set to that limit.

4. The positions of PSO particles are updated using following equation

$$\mathbf{p}^{i,n+1}_{pso} = \mathbf{p}^{i,n}_{pso} + T_F\ \mathbf{v}^{i,n+1}_{pso} \tag{4.13}$$

5. The maximum and minimum limits of particles positions are checked and if they exceed the specified limit, it is set to that limit.

6. The new fitness values of PSO particles are calculated using equation 4.7

and new inertia weights, particle best position, group best positions and global best position are updated.

7. The condition for stopping criterion is checked, stop if it is met, otherwise repeat from step 2 of DPSO.

### 4.2.2 Modified Particle Filter processing

PF processing steps are following.

1. First, PSO particles group best fitness values are compared to PF particle weights, if PSO particles best fitness value is higher than PF weight,their positions and particle group best fitness values are assigned to PF particle positions and weights respectively.
2. In this step, the PF particle weight and resampling limit are checked against the set threshold and if the condition is met, we forward these particles to the estimation stage, otherwise PF particles are resampled using equation (4.5) and repeat the same step.

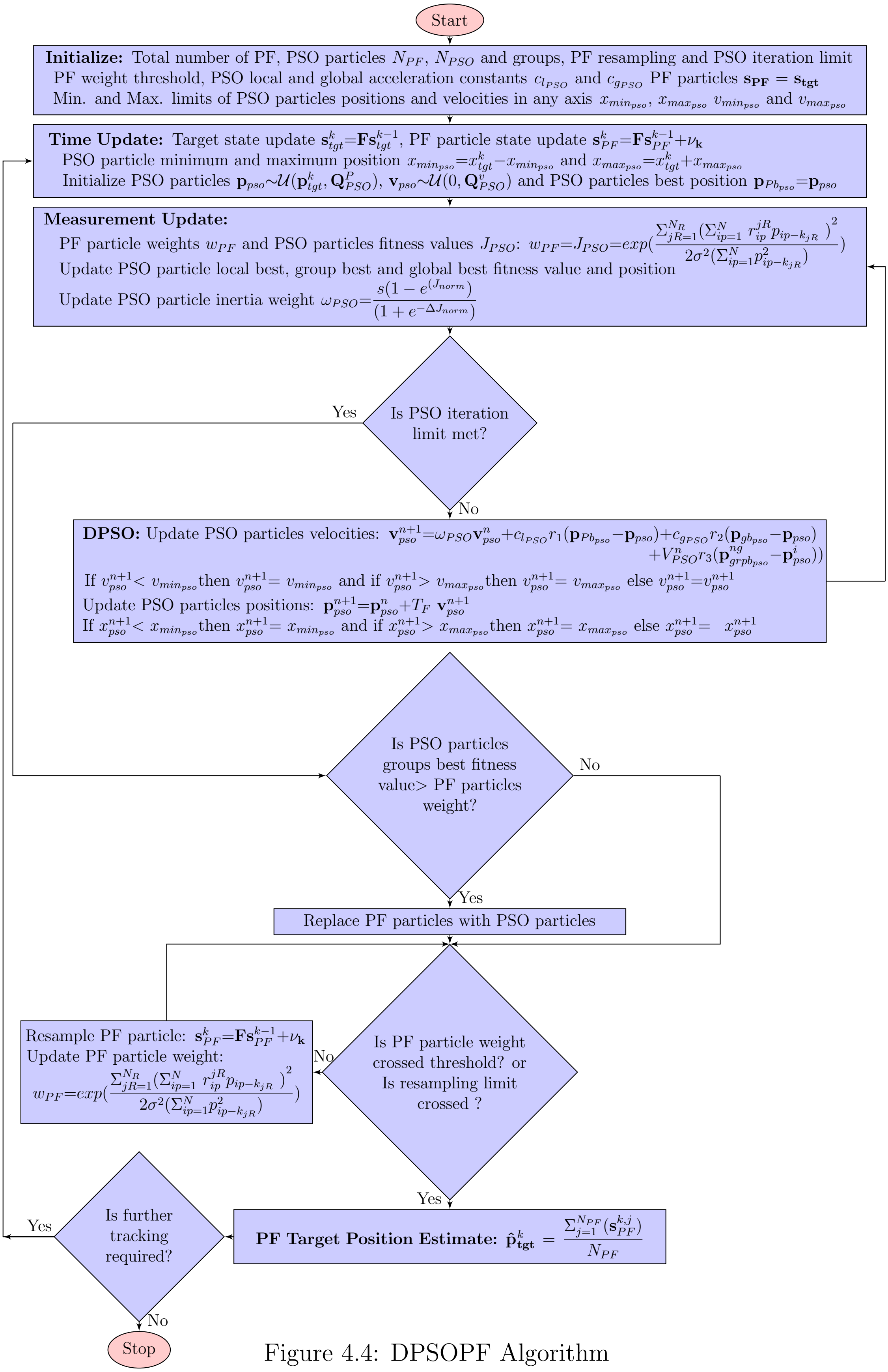
Start
**Initialize:** Total number of PF, PSO particles $N_{PF}$, $N_{PSO}$ and groups, PF resampling and PSO iteration limit
PF weight threshold, PSO local and global acceleration constants $c_{l_{PSO}}$ and $c_{g_{PSO}}$ PF particles $\mathbf{s_{PF}} = \mathbf{s_{tgt}}$
Min. and Max. limits of PSO particles positions and velocities in any axis $x_{min_{pso}}$, $x_{max_{pso}}$ $v_{min_{pso}}$ and $v_{max_{pso}}$
**Time Update:** Target state update $\mathbf{s}_{tgt}^{k}=\mathbf{F}\mathbf{s}_{tgt}^{k-1}$, PF particle state update $\mathbf{s}_{PF}^{k}=\mathbf{F}\mathbf{s}_{PF}^{k-1}+\nu_{\mathbf{k}}$
PSO particle minimum and maximum position $x_{min_{pso}}=x_{tgt}^{k}-x_{min_{pso}}$ and $x_{max_{pso}}=x_{tgt}^{k}+x_{max_{pso}}$
Initialize PSO particles $\mathbf{p}_{pso}\sim\mathcal{U}(\mathbf{p}_{tgt}^{k},\mathbf{Q}_{PSO}^{P})$, $\mathbf{v}_{pso}\sim\mathcal{U}(0,\mathbf{Q}_{PSO}^{v})$ and PSO particles best position $\mathbf{p}_{Pb_{pso}}=\mathbf{p}_{pso}$
**Measurement Update:**
PF particle weights $w_{PF}$ and PSO particles fitness values $J_{PSO}$: $w_{PF}=J_{PSO}=exp(\frac{\Sigma_{jR=1}^{N_R}(\Sigma_{ip=1}^{N}\ r_{ip}^{jR}p_{ip-k_{jR}}\ )^2}{2\sigma^2(\Sigma_{ip=1}^{N}p_{ip-k_{jR}}^2)})$
Update PSO particle local best, group best and global best fitness value and position
Update PSO particle inertia weight $\omega_{PSO}=\frac{s(1-e^{(J_{norm})})}{(1+e^{-\Delta J_{norm}})}$
Is PSO iteration limit met?
Yes
No
**DPSO:** Update PSO particles velocities: $\mathbf{v}_{pso}^{n+1}=\omega_{PSO}\mathbf{v}_{pso}^{n}+c_{l_{PSO}}r_1(\mathbf{p}_{Pb_{pso}}-\mathbf{p}_{pso})+c_{g_{PSO}}r_2(\mathbf{p}_{gb_{pso}}-\mathbf{p}_{pso})$
$+V_{PSO}^{n}r_3(\mathbf{p}_{grpb_{pso}}^{ng}-\mathbf{p}_{pso}^{i}))$
If $v_{pso}^{n+1}< v_{min_{pso}}$then $v_{pso}^{n+1}= v_{min_{pso}}$ and if $v_{pso}^{n+1}> v_{max_{pso}}$then $v_{pso}^{n+1}= v_{max_{pso}}$ else $v_{pso}^{n+1}=v_{pso}^{n+1}$
Update PSO particles positions: $\mathbf{p}_{pso}^{n+1}=\mathbf{p}_{pso}^{n}+T_F\ \mathbf{v}_{pso}^{n+1}$
If $x_{pso}^{n+1}< x_{min_{pso}}$then $x_{pso}^{n+1}= x_{min_{pso}}$ and if $x_{pso}^{n+1}> x_{max_{pso}}$then $x_{pso}^{n+1}= x_{max_{pso}}$ else $x_{pso}^{n+1}=\ \ x_{pso}^{n+1}$
Is PSO particles groups best fitness value> PF particles weight?
No
Yes
Replace PF particles with PSO particles
Resample PF particle: $\mathbf{s}_{PF}^{k}=\mathbf{F}\mathbf{s}_{PF}^{k-1}+\nu_{\mathbf{k}}$
Update PF particle weight:
$w_{PF}=exp(\frac{\Sigma_{jR=1}^{N_R}(\Sigma_{ip=1}^{N}\ r_{ip}^{jR}p_{ip-k_{jR}}\ )^2}{2\sigma^2(\Sigma_{ip=1}^{N}p_{ip-k_{jR}}^2)})$
No
Is PF particle weight crossed threshold? or Is resampling limit crossed ?
Yes
**PF Target Position Estimate:** $\hat{\mathbf{p}}_{\mathbf{tgt}}^{k}=\frac{\Sigma_{j=1}^{N_{PF}}(\mathbf{s}_{PF}^{k,j})}{N_{PF}}$
Yes
Is further tracking required?
No
Stop

Figure 4.4: DPSOPF Algorithm

## CHAPTER 5

# EXPERIMENTAL SETUP AND RESULTS

## 5.1 Test setup

A square surveillance area of 100 × 100 meters is monitored by a UWB multistatic radar, which consists of one transmitter (Tx) and three receiver nodes (Rx1, Rx2 and Rx3). Transmitter and receiver nodes are located in the center of each side of surveillance area.

Origin of the coordinate system is located at lower left corner of the surveillance area while the position of Tx node is (0,50) and Rx nodes are positioned at (50,0),(100,50) and (50,100).

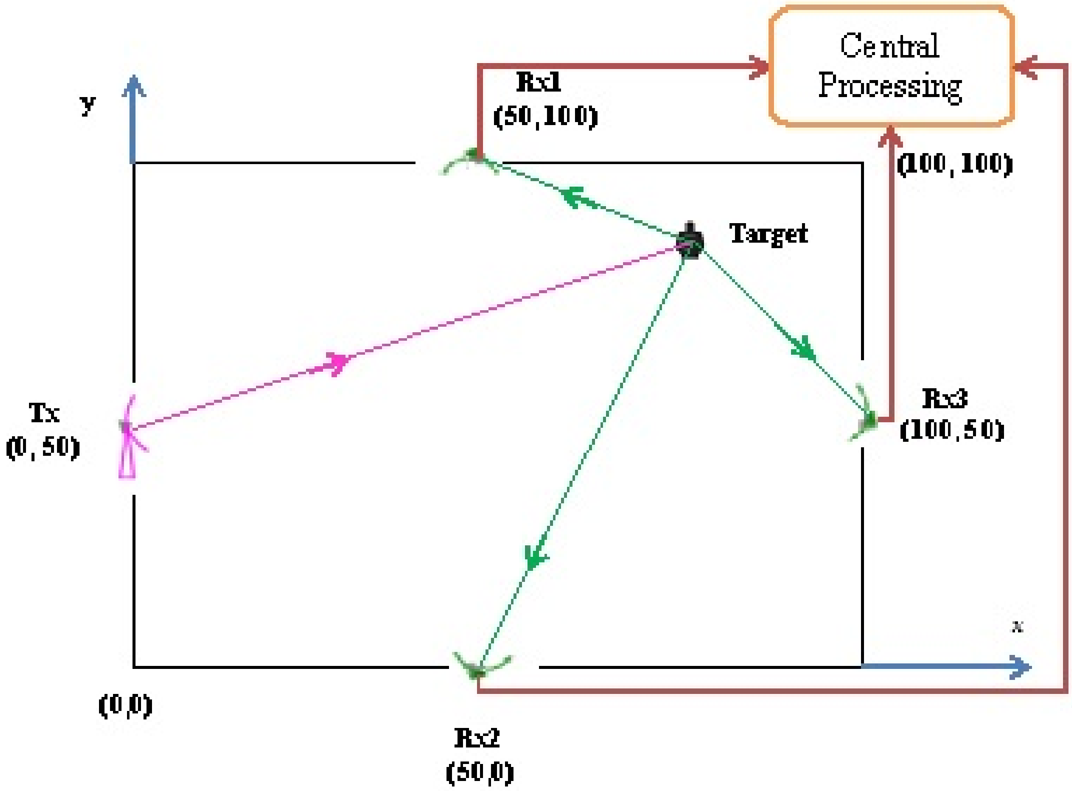

Figure 5.1: Radar Sensor Network Setup

### 5.1.1 Radar sensor network parameters

Tx node transmits 1.4 nanosecond width first derivative Gaussian mono-cycle pulse of Amplitude 1. After transmission of one pulse, wait time for reflected signal is 940 ns as shown in figure 5.2. All processing is done by taking one pulse in a frame where frame duration is 200 ms. The received signal is sampled at 1.5GHz.

Testing is performed from -20dB to 5dB SNR and channel gain is randomly varied between -1 to 1. All algorithms are tested by using same random seed.

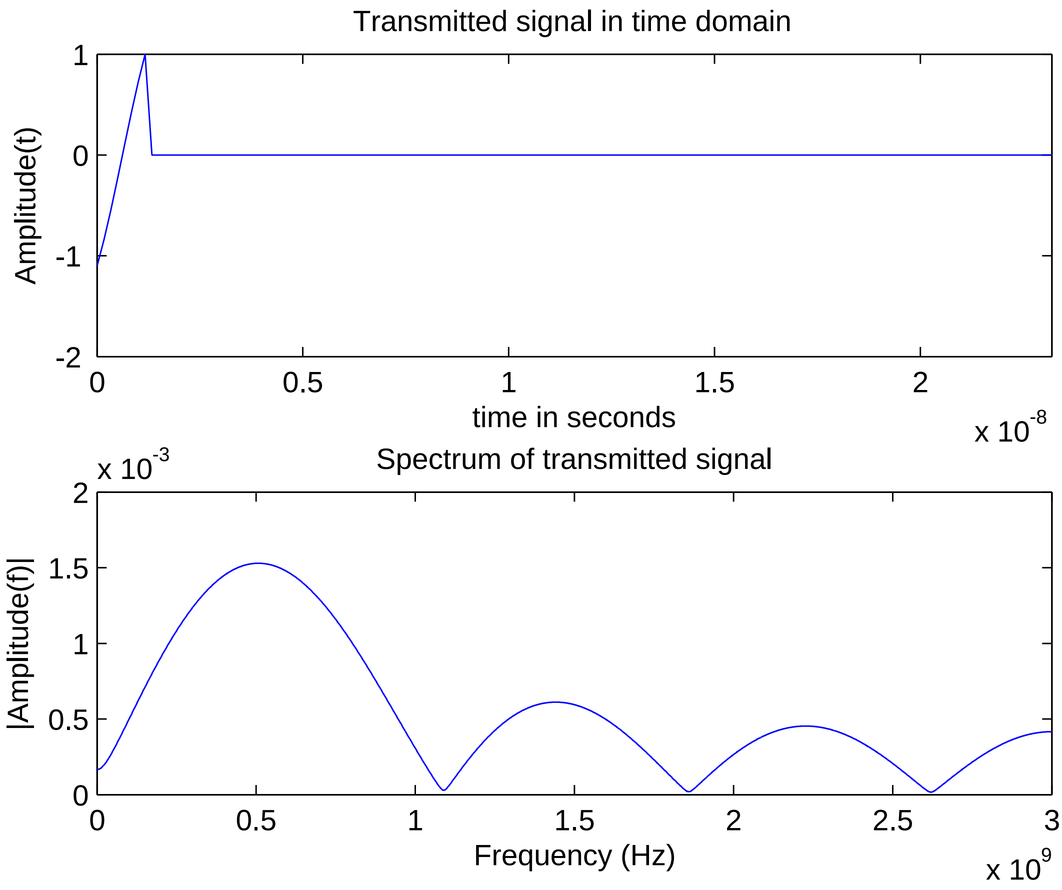

Figure 5.2: First derivative Gaussian monocycle pulse with reflection time wait and its spectrum

### 5.1.2 Target model

For radar systems, target is characterized by radar cross section (RCS) function [7]. A target's RCS represents the amount of energy reflected from the target toward the receiver as a function of the target aspect with respect to the transmitter/receiver pair. It is well known that this function is rapidly changing as a function of the target aspect [7]. Target radar cross section area (RCS) for human is assumed 1 meter square and shape of the target is rectangular similar to normal human width 0.45 cm and thickness 0.24 cm [76].

Target is modeled as Swerling 1 model. Swerling 1 models the target's RCS with two degrees of freedom Chi-square distribution. Target's RCS is vary from frame to frame, but not varying within one frame.

### 5.1.3 Testing trajectory and Target velocities

Four different trajectories are tested. First trajectory is a simple straight line shown in figure 5.3. Second trajectory is a circle shown in figure 5.4. Third trajectory is a mix of straight line and ellipse as shown in figure 5.5 and fourth trajectory is a zigzag as shown in figure 5.6 where the arrows represent the direction of movement. In all trajectories, target speed is 11KM/hour, but target x axis and y axis velocities are varying due to change of direction of the target. It is assumed that the target acceleration variance is twice of target speed.

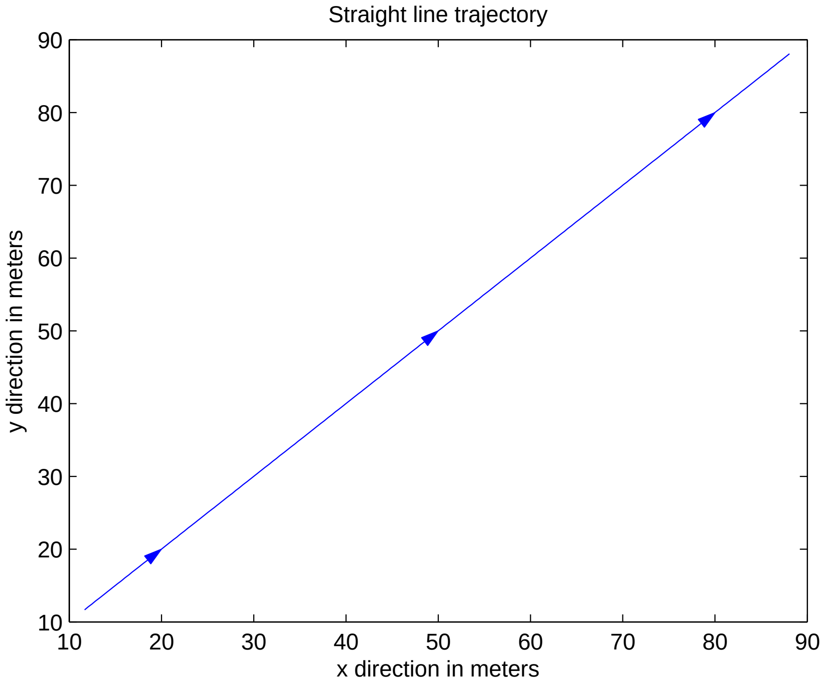

Figure 5.3: Straight line trajectory

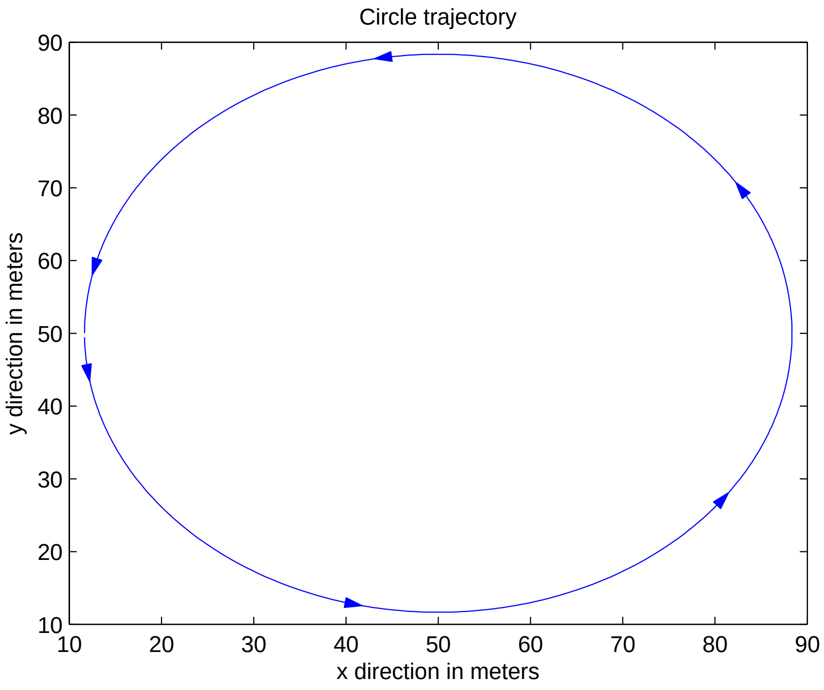

Figure 5.4: Circle trajectory

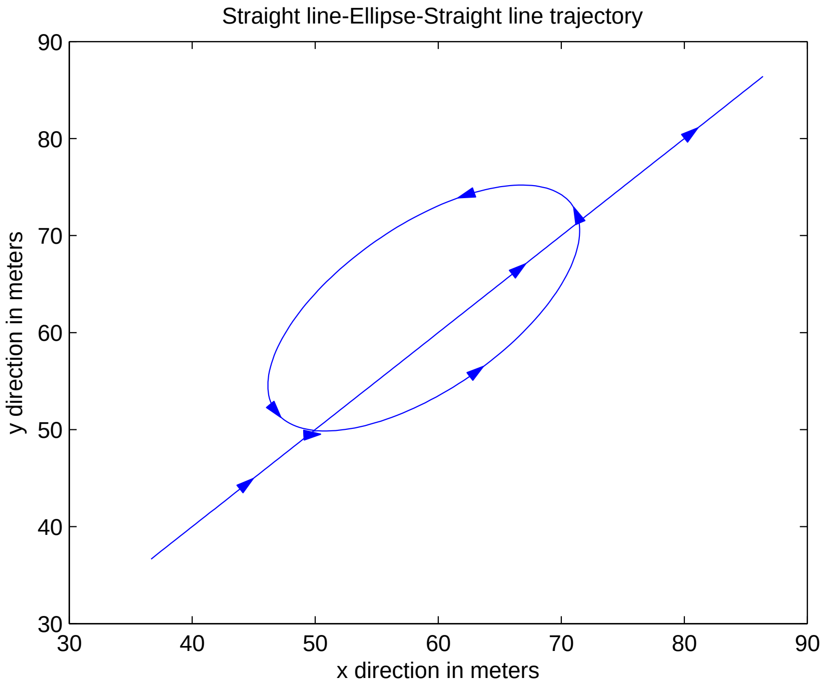

Figure 5.5: Straight line - Ellipse - Straight line trajectory

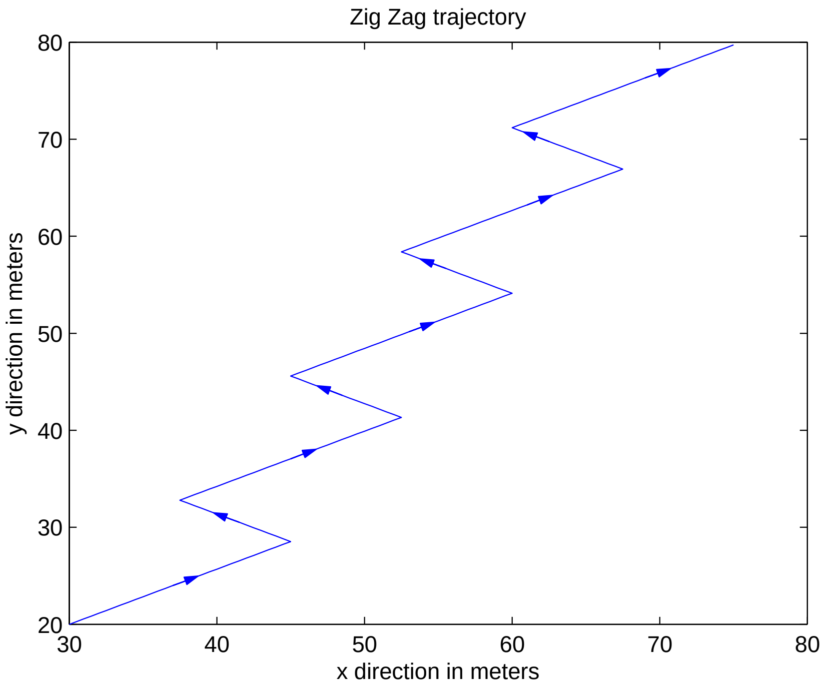

Figure 5.6: Zigzag trajectory

### 5.1.4 Algorithm parameters

All algorithms are tested by the implementation of 100 PSO particles. In DPSOPF, each group consists of 4 particles, therefore there are 25 group best particles. AWPSOPF also selects 25 particles for PF processing so its performance can be measured on the same scale. All PSO based algorithms are stopped after 10 iterations where PF particles resampling limit is also 10. So PSO and PF algorithms have the same number of particle movements, but AWPSOPF and DPSOPF have additional particle movements and DPSOPF also have additional computational complexity due to particles grouping.

## 5.2 Results

PF, AWPSO. AWPSOPF and DPSOPF are tested by same random seed on different trajectories and at different SNRs. All algorithms are tested in the low SNR scenario. Best and worst SNR due to channel gain and fluctuating target model is shown in figure 5.7 where SNR due to noise only is -20dB.

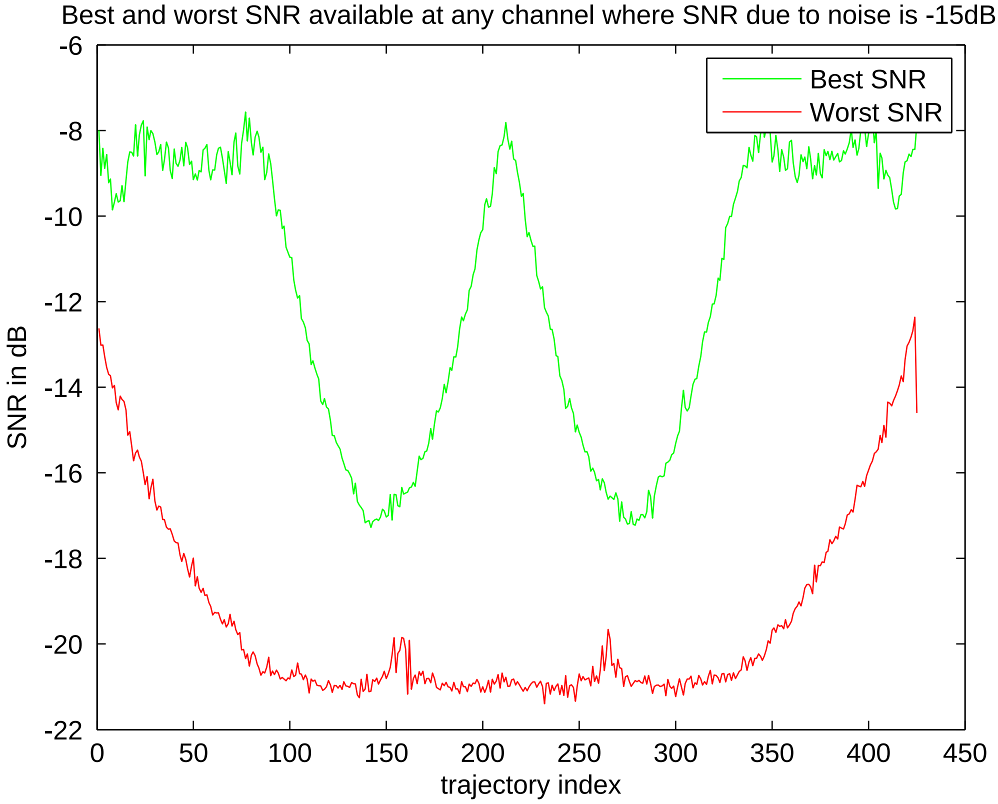

Figure 5.7: Best and worst SNR where SNR due to noise is -20dB

### 5.2.1 Algorithm tracking Error

The tracking error is calculated by following equation

$$E_{RMSE} = \sqrt{\frac{\Sigma_{t_i=1}^{T}(\mathbf{p}_{tgt_{t_i}} - \hat{\mathbf{p}}_{tgt_{t_i}})^2}{T}} \tag{5.1}$$

where $t_i$ is trajectory index, $T$ is total trajectory points, $\mathbf{p}_{tgt}$ is actual target position, $\hat{\mathbf{p}}_{tgt}$ is estimated target position and $E_{RMSE}$ is Root Mean Square Error (RMSE) of that trajectory. If in any trajectory target tracking is lost, then error after tracking lost is not included in calculations. Following figures 5.8, 5.9,5.10 and 5.11 show different algorithm errors in different trajectories.

### Root Mean Square Error of Straight line trajectory

It can be seen from figure 5.8 that PSO based target tracking has highest error rate at -20 dB, tracking error significantly reduce at -15 dB but it is still higher than AWPSOPF and DPSOPF. Tracking error of PSO is very close when the SNR is -10 dB or more than -10 dB . Where PF also has higher tracking error as compared to AWPSOPF and DPSOPF algorithms at -20 dB and from -15 dB and above PF tracking error is highest among all other tested algorithms. The AWPSOPF tracking error is also higher than DPSOPF algorithm at -20 dB, but at -15 dB and above both algorithms have similar tracking error. Figure 5.8 also shows that in RMSE terms DPSOPF is performing best and AWPSOPF results are also superior than PSO and PF algorithms.

### Root Mean Square Error of Circle trajectory

It can be seen from figure 5.9 that PSO based target tracking has highest error rate at -20dB, tracking error significantly reduce at -15dB and very close to AWPSOPF and DPSOPF tracking error. Where PF also have higher tracking error as compare to AWPSOPF and DPSOPF algorithms at -20dB. From -15 dB and

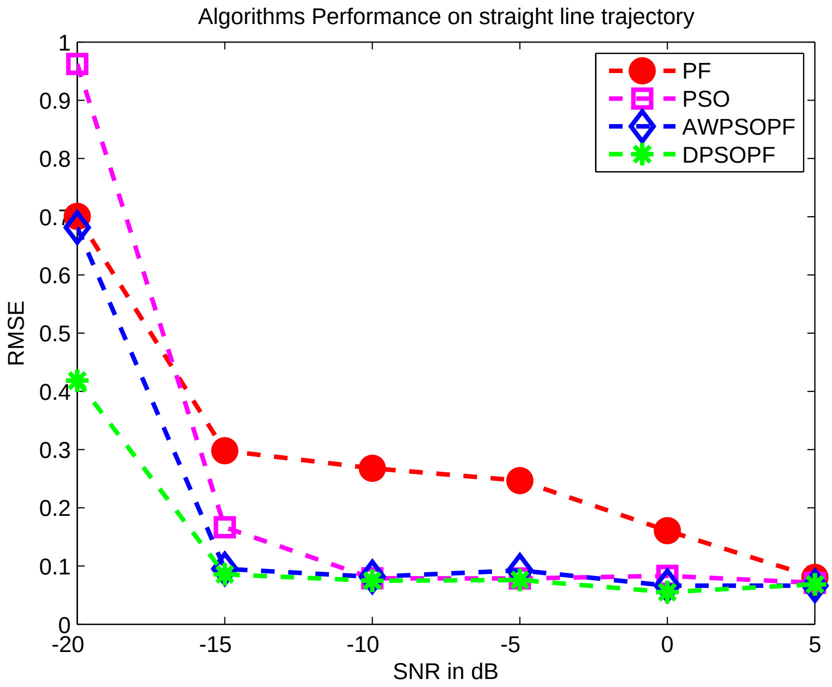

Figure 5.8: RMS tracking error on straight line trajectory

above SNRs, the PF tracking error is highest among all other tested algorithms. The AWPSOPF tracking error is also higher than DPSOPF algorithm at -20dB but at -15 dB and above both algorithms have similar tracking error. Figure 5.9 also shows that in RMSE terms DPSOPF is performing best and AWPSOPF results are also superior than PSO and PF algorithms in worst condition.

### Root Mean Square Error of Straight line Ellipse Straight line trajectory

It can be seen from figure 5.10 that PSO based target tracking has highest error rate at -20 dB and it remains highest up to -10 dB. Tracking error of PSO is very close when the SNR is -5 dB or more than -5 dB . Where PF also has higher tracking error as compare to AWPSOPF and DPSOPF algorithms at -20 dB, -15 dB and -10dB , From -5 dB and above PF tracking error is highest among

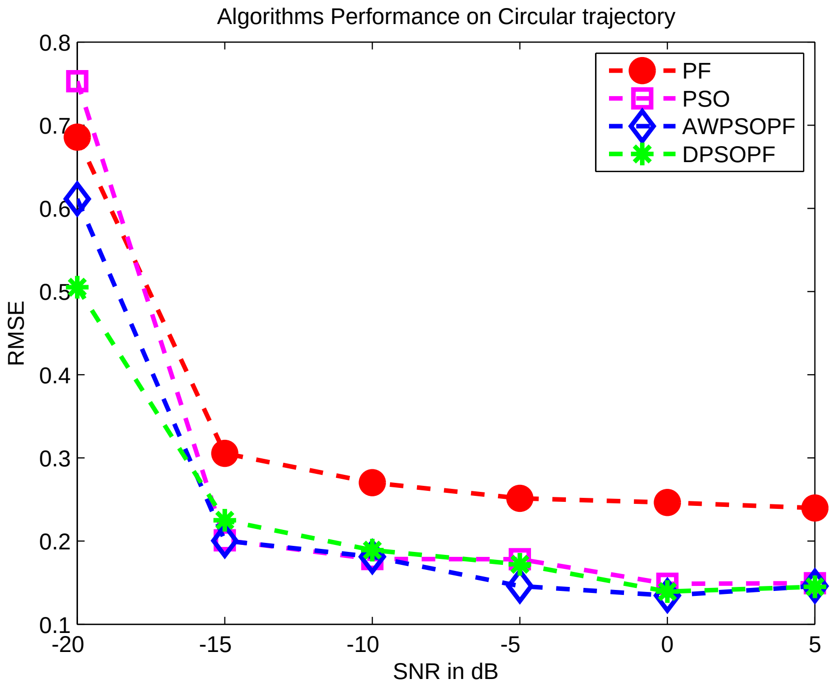

Figure 5.9: RMS tracking error on circular trajectory

all other tested algorithms. AWPSOPF and DPSOPF algorithms have similar tracking error.

### Root Mean Square Error of zigzag trajectory

As seen from figure 5.11 that PF has highest tracking error at -20 dB and PF tracking error is similar to all other remaining algorithms when the SNR is -15dB or more.

### Comparison of Root Mean Square Error Tracking Error

The tracking error is compared for different algorithms to evaluate performance of proposed algorithms. Tracking error of different algorithms is given in following Table.

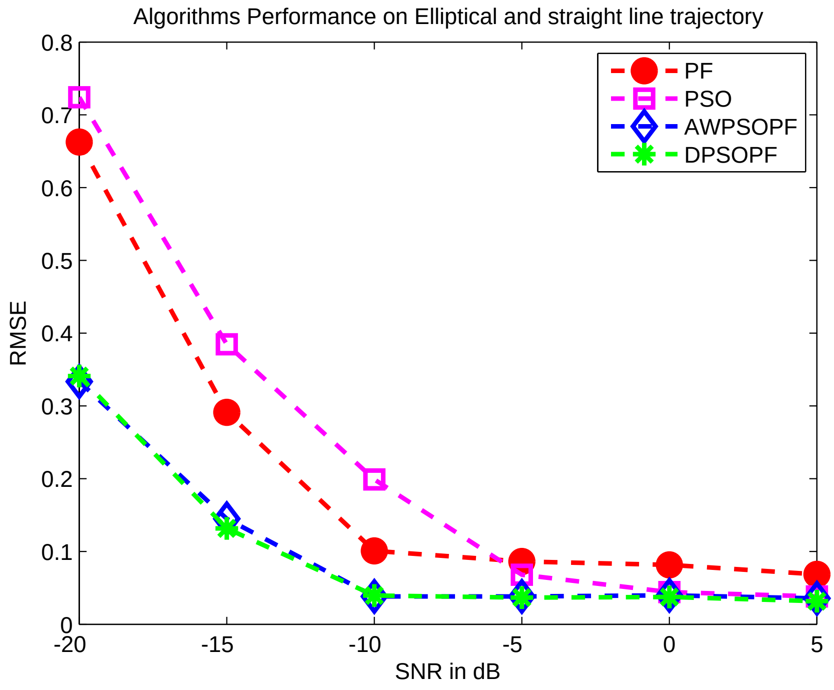

Figure 5.10: RMS tracking error on a straight line, ellipse & straight line trajectory

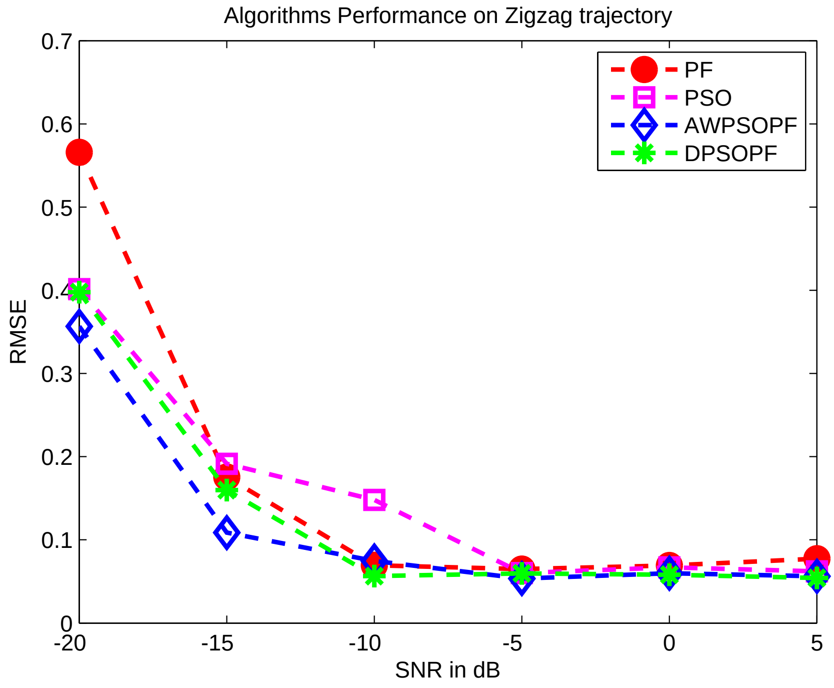

Figure 5.11: RMS tracking error on Zigzag trajectory

Table 5.1: Tracking Error Comparisons

| Method | Tracking error (in meters) |
|---|---|
| PF (UWB) [77] | 0.500 |
| PFPSO [78] | 0.2891 |
| Fuzzy logic cat 1 [79] | 0.263 |
| Fuzzy logic cat 2 [79] | 0.275 |
| Fuzzy logic cat 3 [79] | 0.356 |
| PF (UWB-IR)[Resimulated] | 0.3091 |
| PSO [Resimulated] | 0.3153 |
| AWPSOPF [Proposed] | 0.1977 |
| DPSOPF [Proposed] | 0.1922 |

It can be seen from table 5.1 that DPSOPF outperforms all other algorithms where AWPSOPF is also better than all other algorithms except DPSOPF.

### 5.2.2 Algorithm tracking Robustness

The tracking robustness of the algorithm is measured as how many times a complete trajectory is tracked with no tracking loss and calculating percentage with total tested trajectories at different SNRs. The tracking robustness of the proposed algorithms are compared with other methods as shown Table 5.2. The robustness of our algorithms is more than PSO and PF while DPSOPF outperforms all algorithms in terms of robustness.

Table 5.2: Tracking robustness comparisons

| Method | Tracking robustness (%) |
|---|---|
| PF | 54.17 |
| PSO | 87.50 |
| AWPSOPF | 95.80 |
| DPSOPF | 99.20 |

### 5.2.3 Computation Complexity Comparison of Algorithm

The computation complexity of the algorithm is calculated based on number of multiplication and number of additions required for the implementation of algorithm. Table 5.3 shows computational complexity where $l_{PF}$ is resampling limit for PF method, $l_{PSO}$ is the PSO iteration limit, $N_{PF}$ is the number of PF particles and $N_{PSO}$ is number of PSO particles. Each PF particle required three multiplications, one for particle movement and two multiplications to draw noise vector from multimodal distribution according to the equation 4.5. Each particle required two additions according to equation 4.5 and equation 4.10. Each PSO particle required four multiplications and three additions according to equation 3.43 and equation 3.44. In AWPSOPF algorithm PSO multiplication and addition operations remain the same where additional multiplications and additions are added due to PF. There are five multiplications and four additions for each PSO particle in DPSO according to equation 4.12 and equation 4.13 where additional multiplications and additions are added due to PF.

Table 5.3: Computational Complexity of Algorithms

| Method | Number of Multiplications | Number of Additions |
|---|---|---|
| PF | $3 \times l_{PF} \times N_{PF}$ | $2 \times l_{PF} \times N_{PF}$ |
| PSO | $4 \times l_{PSO} \times N_{PSO}$ | $3 \times l_{PSO} \times N_{PSO}$ |
| AWPSOPF | $4 \times l_{PSO} \times N_{PSO} + 3 \times l_{PF} \times N_{PF}$ | $3 \times l_{PSO} \times N_{PSO} + 2 \times l_{PF} \times N_{PF}$ |
| DPSOPF | $5 \times l_{PSO} \times N_{PSO} + 3 \times l_{PF} \times N_{PF}$ | $4 \times l_{PSO} \times N_{PSO} + 2 \times l_{PF} \times N_{PF}$ |

For PF method, $N_{PF}$ is 100 and $l_{PF}$ is 10. For PSO method $N_{PSO}$ is 100 and $l_{PSO}$ is 10. For AWPSOPF and DPSOPF method $N_{PSO}$ is 100, $l_{PSO}$ is $10, N_{PF}$ is 25 and $l_{PF}$ is 10.

Table 5.4: Numeric Computational Complexity of algorithms

| Method | Number of Multiplications | Number of Additions |
|---|---|---|
| PF | $3 \times 10 \times 100 = 3000$ | $2 \times 10 \times 100 = 2000$ |
| PSO | $4 \times 10 \times 100 = 4000$ | $3 \times 10 \times 100 = 3000$ |
| AWPSOPF | $4 \times 10 \times 100 + 3 \times 10 \times 25 = 4750$ | $3 \times 10 \times 100 + 2 \times 10 \times 25 = 3500$ |
| DPSOPF | $5 \times 10 \times 100 + 3 \times 10 \times 25 = 5750$ | $4 \times 10 \times 100 + 2 \times 10 \times 25 = 4500$ |

### 5.2.4 Tracked and untracked trajectories

This section shows a few samples of completely tracked trajectories and examples where tracking breaks.

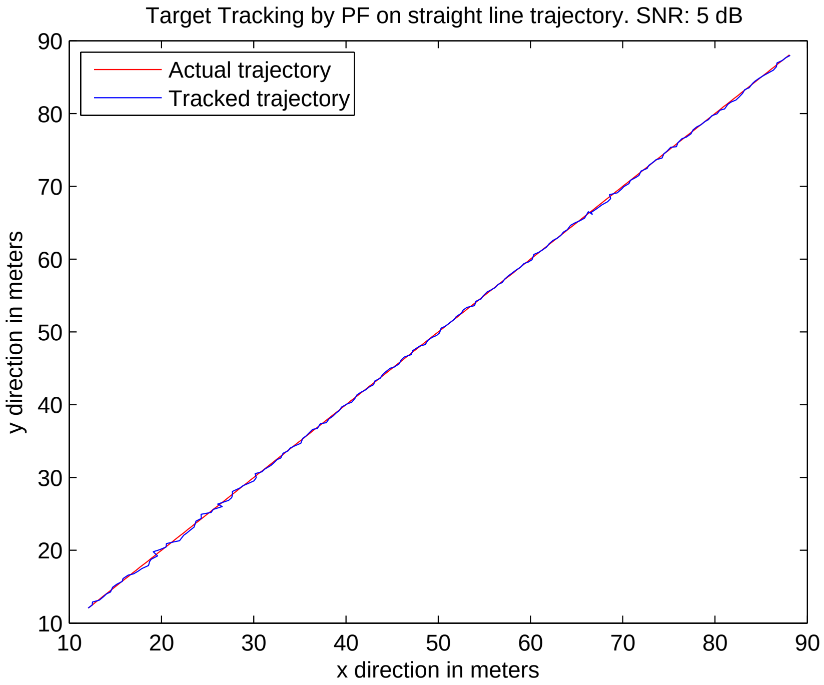

Figure 5.12: Target tracking on a straight line trajectory SNR 5 dB by PF

In figure 5.12, red shows the actual trajectory followed by the target, actual target is moving at a speed of 11 KM/hr on a straight line from left bottom to right top where this target is tracked by PF as shown by the blue and measurement SNR is 5dB where acceleration variance is 22 KM/hr.

In figure 5.13 red shows actual target is moving on a straight line trajectory

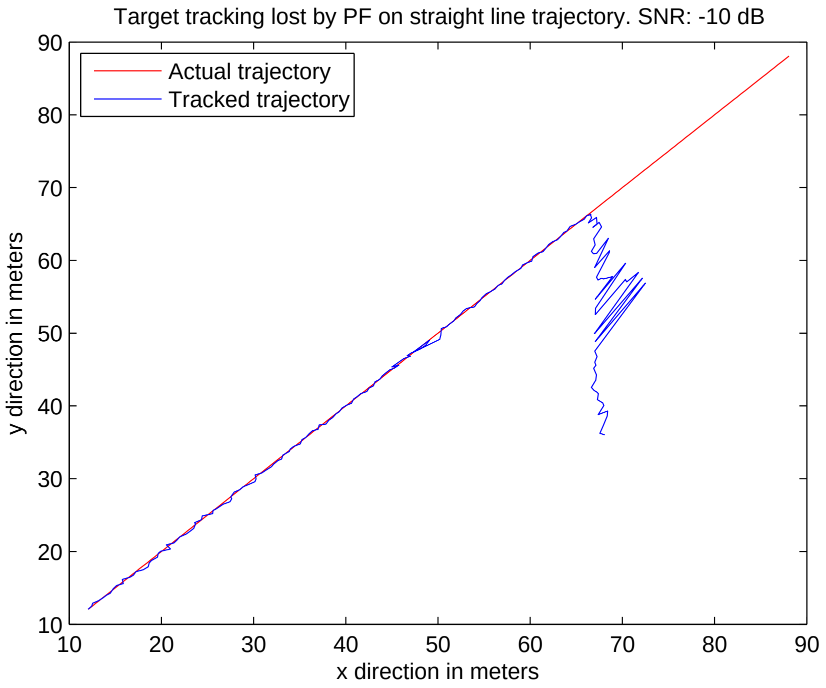

Figure 5.13: Target tracking lost on Straight line trajectory SNR -10 dB by PF

and blue shows that PF lost target tracking where measurement SNR is -10 dB where acceleration variance is 22 KM/hr. At the start of tracking target is tracked successfully, but after covering approximately more than mid trajectory tracking is lost and not recovered again. In figure 5.14 red shows that actual target is moving at a circular trajectory and blue shows that DPSOPF successfully tracked target where measurement SNR is 5 dB where acceleration variance is 22 KM/hr.

In figure 5.15 red shows actual target is moving at a circular trajectory and blue shows that PF lost target tracking where measurement SNR is 0 dB. At the start of tracking target is tracked successfully, but after covering approximately three quarters of trajectory, tracking is lost.

In figure 5.16 red shows that actual target is moving at a straight then ellipti-

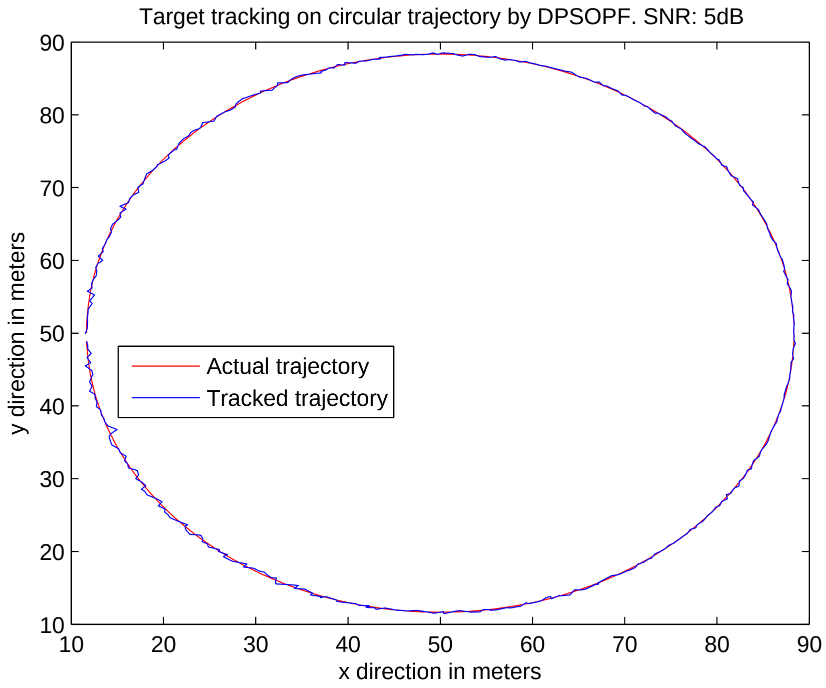

Figure 5.14: Target tracking on circular trajectory SNR 5dB by DPSOPF

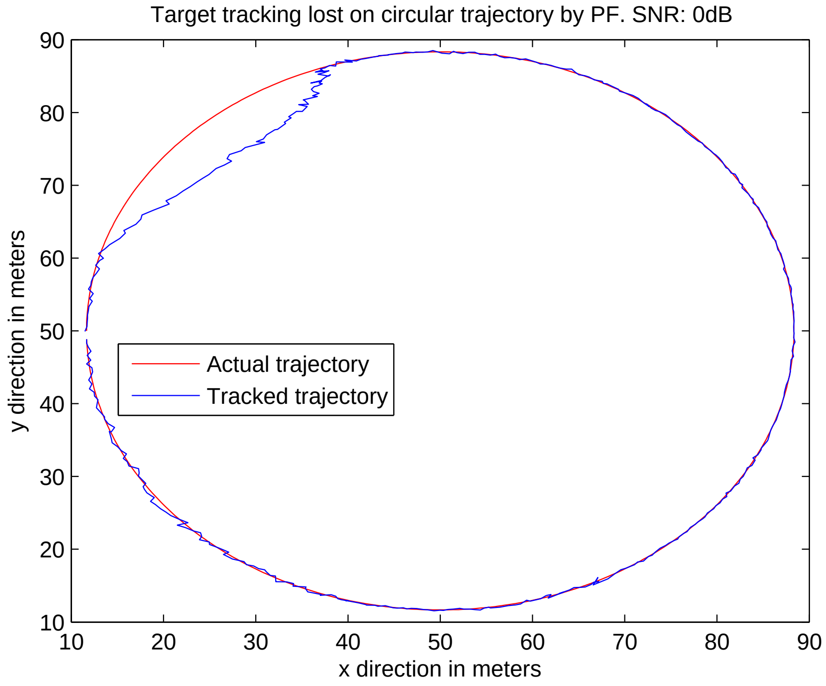

Figure 5.15: Target Tracking lost on Circular trajectory SNR 0dB by PF

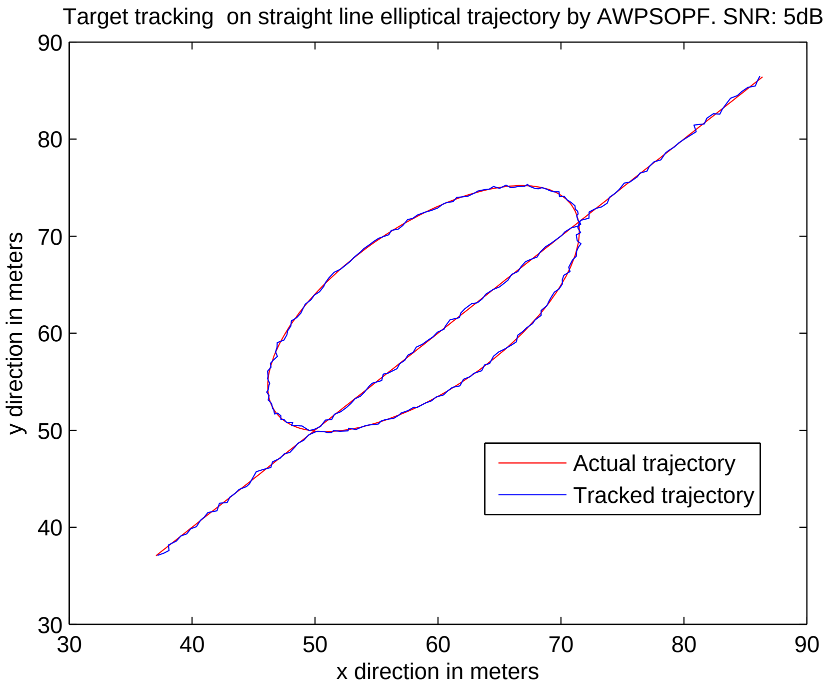

Figure 5.16: Target tracking on straight line elliptical trajectory SNR 5dB by AWPSOPF

cal then straight trajectory and blue shows that AWPSOPF successfully tracked target where measurement SNR is 5 dB where acceleration variance is 22 KM/hr.

In figure 5.17 red shows actual target is moving first on a straight, then an elliptical, then again on a straight trajectory and blue shows that PSO lost target tracking where measurement SNR is -15 dB where acceleration variance is 22 KM/hr. At the start of tracking target is tracked successfully, but when the target turns from elliptical to the straight path, target tracking is lost. In figure 5.18 red shows that actual target is moving at a zigzag trajectory and blue shows that PSO successfully tracked target where measurement SNR is 5 dB where acceleration variance is 22 KM/hr.

In figure 5.19 red shows actual target is moving on zigzag trajectory and blue shows

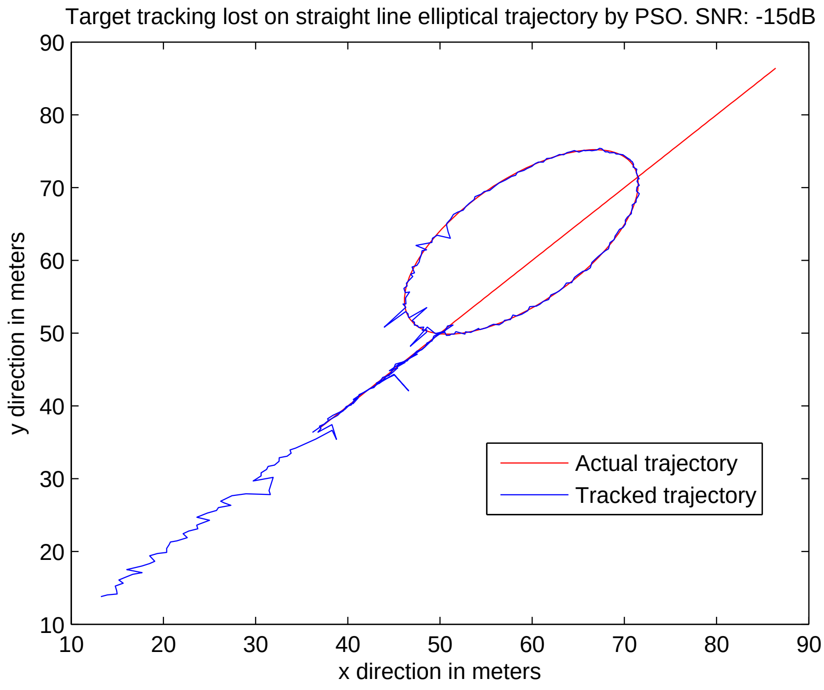

Figure 5.17: Target Tracking lost by PSO on Straight line elliptical trajectory and SNR -15dB

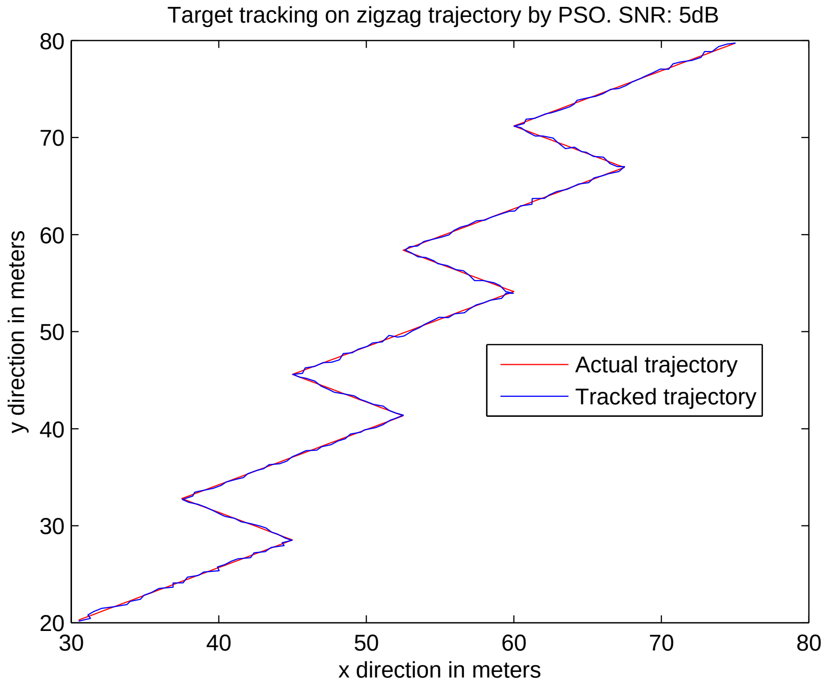

Figure 5.18: Target Tracking by PSO on zigzag trajectory and SNR 5dB

that PF lost target tracking where measurement SNR is -20 dB where acceleration variance is 22 KM/hr. At the start of tracking target is tracked successfully, but after covering half of trajectory tracking is lost.

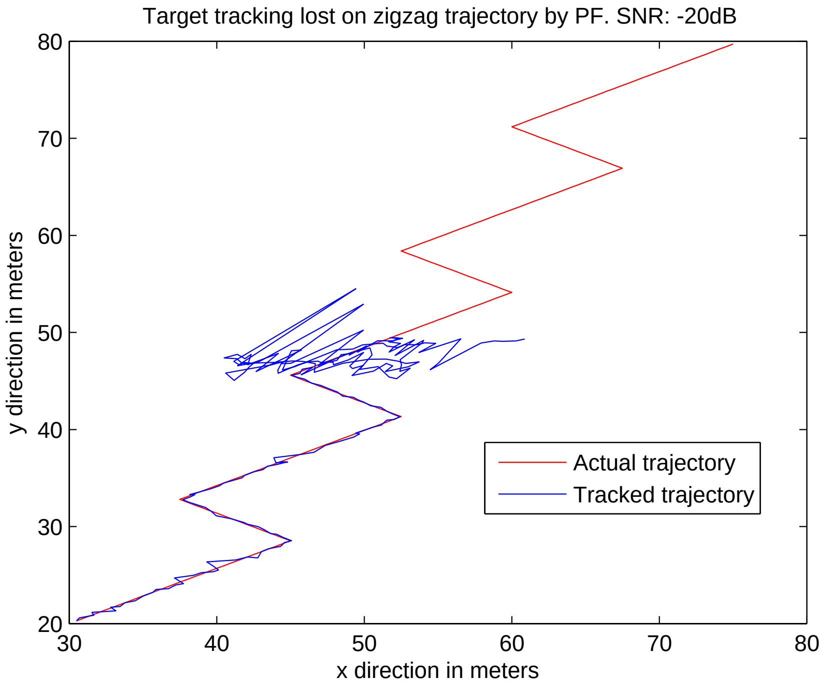

Figure 5.19: Target Tracking lost by PF on Zigzag trajectory and SNR -20dB

## CHAPTER 6

# CONCLUSION AND FUTURE WORK

In this work PF embedded PSO based on adaptive inertia weights and particle neighbourhood grouping based algorithms are proposed. The PF sample impoverishment problem is solved by using the diversity of PSO particles. The convergence of PSO algorithm is improved by using fitness based adaptive weights. The PSO bias problem is tackled by using PF algorithm. The performance of PSO algorithm is improved by increasing social interaction. The robustness of PF algorithm is improved by embedding PSO algorithm. The combination of these two PSO variants with PF not only improves accuracy, but also provides a robust solution for target tracking problem as in the case of human tracking in indoor environments.

The combination of PSO and PF increases the computational complexity. It is required to reduce this computational complexity using efficient implementation techniques like drop the PSO particles if these particles fitness value is decreasing compared to other PSO particles after some iterations.

The proposed work can be further extended using clutter mitigation techniques for indoor environment and fine tuning of different algorithm parameters for better performance in cluttered environments. The proposed work can be further extended by using different distribution techniques of small swarm other than minimum distance. These different distribution techniques can be analyzed for different scenarios. The search space can be divided into geometric regions by squares or cubes, circles or spheres, etc. Particles can be equally distributed in these search space divisions where particles can be randomly distributed within distributed regions. Figure 6.1 shows such division where 96 particles are divided into 16 regions and each region have 6 particles that are randomly distributed within each region.

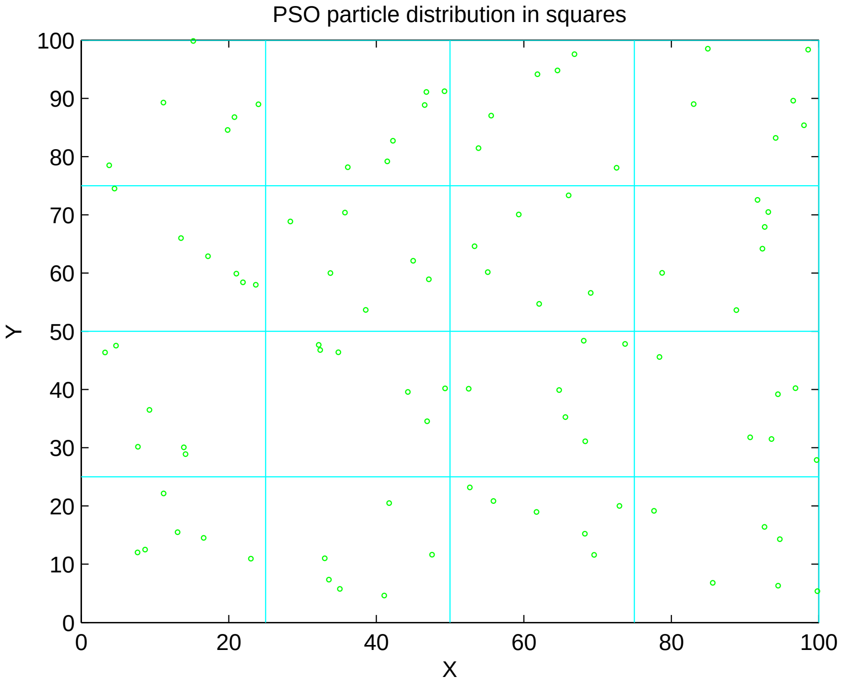

Figure 6.1: Particle distribution in squares

# REFERENCES

[1] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols.* Cambridge University Press, 2011.

[2] M. Mallick, V. Krishnamurthy, and B. Vo, *Integrated Tracking, Classification, and Sensor Management: Theory and Applications.* Wiley, 2012.

[3] B. EDDE, *Radar: Principles, Technology, Applications.* Pearson Education, 2008.

[4] P. Dutta, A. Arora, and S. Bibyk, "Towards radar-enabled sensor networks," in *The Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, 2006, pp. 467–474.

[5] R. Kozma, L. Wang, K. Iftekharuddin, E. McCracken, M. Khan, K. Islam, S. R. Bhurtel, and R. M. Demirer, "A radar-enabled collaborative sensor network integrating COTS technology for surveillance and tracking," *Sensors*, vol. 12, no. 2, pp. 1336–1351, 2012.

[6] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, “Ranging with ultrawide bandwidth signals in multipath environments,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404–426, 2009.

[7] M. Skolnik, *Introduction to Radar Systems.* McGraw Hill, 2002.

[8] V. S. Chernyak, *Fundamentals of Multisite Radar Systems: Multistatic Radars and Multistatic Radar Systems.* CRC Press, 1998.

[9] M. Chiani and A. Giorgetti, “Coexistence between UWB and narrow-band wireless communication systems,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 231–254, 2009.

[10] T. Kaiser and F. Zheng, *Ultra Wideband Systems with MIMO.* Wiley, 2010.

[11] H. Soganci, S. Gezici, and H. V. Poor, “Accurate positioning in ultra-wideband systems,” *Wireless Communications, IEEE*, vol. 18, no. 2, pp. 19–27, 2011.

[12] X. Wang, M. Fu, and H. Zhang, “Target tracking in wireless sensor networks based on the combination of KF and MLE using distance measurements,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 567–576, 2012.

[13] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* PTR Prentice-Hall, 1993.

[14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications.* Artech house Boston, 2004.

[15] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 25, no. 7, pp. 53–82, 2010.

[16] F. Daum, “Nonlinear filters: beyond the Kalman filter,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 20, no. 8, pp. 57–69, Aug 2005.

[17] J. M. Hammersley and K. W. Morton, “Poor Man’s Monte Carlo,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 16, no. 1, pp. pp. 23–38, 1954.

[18] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, Apr 1993.

[19] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1, 1995, pp. 39–43.

[20] J. Kennedy, R. Eberhart *et al.*, “Particle swarm optimization,” in *Proceedings of IEEE international conference on neural networks*, vol. 4, no. 2. Perth, Australia, 1995, pp. 1942–1948.

[21] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[22] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *IEEE Swarm Intelligence Symposium.* IEEE, 2007, pp. 120–127.

[23] S. Janson and M. Middendorf, "On trajectories of particles in PSO," in *IEEE Swarm Intelligence Symposium.* IEEE, 2007, pp. 150–155.

[24] W. M. Spears, D. T. Green, and D. F. Spears, "Biases in particle swarm optimization," *IJSIR*, vol. 1, no. 2, pp. 34–57, 2010.

[25] M. A. Youssef, A. Agrawala, and A. Udaya Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications(PerCom).* IEEE, 2003, pp. 143–150.

[26] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) Proceedings*, vol. 2. IEEE, 2000, pp. 775–784.

[27] R. Battiti, N. T. Le, and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," *Tech. Rep. DIT-020083, University of Trento*, 2002.

[28] W. C. Headley, C. R. da Silva, and R. M. Buehrer, "Indoor location positioning of non-active objects using ultra-wideband radios," in *IEEE Radio and Wireless Symposium.* IEEE, 2007, pp. 105–108.

[29] M. Svecova, D. Kocur, R. Zetik, and J. Rovnakova, "Target localization by a multistatic uwb radar," in *20th International Conference RADIOELEKTRONIKA*. IEEE, 2010, pp. 1–4.

[30] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 24–40, 2005.

[31] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70–84, 2005.

[32] H. L. Groginsky, "Position estimation using only multiple simultaneous range measurements," *IRE Transactions on Aeronautical and Navigational Electronics*, no. 3, pp. 178–187, 1959.

[33] D. E. Manolakis, "Efficient solution and performance analysis of 3-D position estimation by trilateration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1239–1248, 1996.

[34] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 93–101, 2005.

[35] N. Decarli, F. Guidi, and D. Dardari, “A Novel Joint RFID and Radar Sensor Network for Passive Localization: Design and Performance Bounds,” *IEEE Journal of Selected Topics in Signal Processing*, 2014.

[36] Y. Hong and R. Bartlett, *Routledge Handbook of Biomechanics and Human Movement Science (Routledge International Handbooks).* Routledge, 2008.

[37] I. Ganchev, M. Curado, and A. Kassler, *Wireless Networking for Moving Objects: Protocols, Architectures, Tools, Services and Applications.* Springer International Publishing, 2014.

[38] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[39] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[40] H. R. Hashemipour, S. Roy, and A. J. Laub, “Decentralized structures for parallel Kalman filtering,” *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.

[41] J.-H. Kim, M. West, E. Scholte, and S. Narayanan, “Multiscale consensus for decentralized estimation and its application to building systems,” in *American Control Conference.* IEEE, 2008, pp. 888–893.

[42] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *46th IEEE Conference on Decision and Control.* IEEE, 2007, pp. 5492–5498.

[43] B. Rao and H. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," in *Control Theory and Applications, IEE Proceedings D*, vol. 138, no. 5. IET, 1991, pp. 413–420.

[44] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[45] M. Browne, *Schaum's Outline of Theory and Problems of Physics for Engineering and Science.* McGraw Hill, 1999.

[46] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas and Propagation Magazine*, vol. 45, no. 3, pp. 51–82, 2003.

[47] Y. Song and J. W. Grizzle, "The extended Kalman filter as a local asymptotic observer for nonlinear discrete-time systems," in *American Control Conference.* IEEE, 1992, pp. 3365–3369.

[48] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Artech House, 1999.

[49] M. Di, E. M. Joo, and L. H. Beng, "A comprehensive study of Kalman filter and extended Kalman filter for target tracking in wireless sensor networks," in *IEEE International Conference on Systems, Man and Cybernetics (SMC).* IEEE, 2008, pp. 2792–2797.

[50] C. Hongyang, D. Ping, X. Yongjun, and L. Xiaowei, “A robust location algorithm with biased extended Kalman filtering of TDOA data for wireless sensor networks,” in *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 2. IEEE, 2005, pp. 883–886.

[51] M. A. Caceres, F. Sottile, and M. A. Spirito, “Adaptive location tracking by Kalman filter in wireless sensor networks,” in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB).* IEEE, 2009, pp. 123–128.

[52] A. Ribeiro, I. D. Schizas, S. Roumeliotis, and G. B. Giannakis, “Kalman filtering in wireless sensor networks,” *Control Systems, IEEE*, vol. 30, no. 2, pp. 66–86, 2010.

[53] D. F. Bizup and D. E. Brown, “The over-extended Kalman filter-dont use it!” in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 1, 2003, pp. 40–46.

[54] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[55] M. S. Arulampalam, N. Gordon, M. Orton, and B. Ristic, “A variable structure multiple model particle filter for GMTI tracking,” in *Proceedings of the*

*Fifth International Conference on Information Fusion*, vol. 2. IEEE, 2002, pp. 927–934.

[56] A. S. Housfater, X.-P. Zhang, and Y. Zhou, "Nonlinear Fusion of Multiple Sensors with Missing Data," in *ICASSP (4)*, 2006, pp. 961–964.

[57] G. Ing, "Distributed particle filters for object tracking in sensor networks," Ph.D. dissertation, McGill University, 2005.

[58] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[59] Q. He, L. Wang, and B. Liu, "Parameter estimation for chaotic systems by particle swarm optimization," *Chaos, Solitons & Fractals*, vol. 34, no. 2, pp. 654–661, 2007.

[60] M. Schwaab, E. C. Biscaia Jr, J. L. Monteiro, and J. C. Pinto, "Nonlinear parameter estimation through particle swarm optimization," *Chemical Engineering Science*, vol. 63, no. 6, pp. 1542–1552, 2008.

[61] K.-S. Low, H. Nguyen, and H. Guo, "Optimization of sensor node locations in a wireless sensor network," in *Fourth International Conference on Natural Computation (ICNC)*, vol. 5. IEEE, 2008, pp. 286–290.

[62] A. Gopakumar and L. Jacob, "Localization in wireless sensor networks using particle swarm optimization," in *IET Conference on Wireless, Mobile and Multimedia Networks.* IET, 2008.

[63] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *IEEE International Conference on Systems, Man and Cybernetics (SMC).* IEEE, 2009, pp. 205–210.

[64] Tzafestas, *Stochastic Large-Scale Engineering Systems.* CRC Press, 1992.

[65] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[66] L. Michael and R. Kohno, *Ultra Wideband Signals and Systems in Communication Engineering.* Wiley, 2007.

[67] I. Bekkerman and J. Tabrikian, "Target detection and localization using MIMO radars and sonars," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3873–3883, 2006.

[68] E. Fishler, A. Haimovich, R. S. Blum, L. J. Cimini, D. Chizhik, and R. A. Valenzuela, "Spatial diversity in radars-models and detection performance," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 823–838, 2006.

[69] H. Godrich, A. M. Haimovich, and R. S. Blum, "Target localization techniques and tools for MIMO radar," in *IEEE Radar Conference.* IEEE, 2008, pp. 1–6.

[70] P.-J. Nordlund and F. Gustafsson, "Sequential Monte Carlo filtering techniques applied to integrated navigation systems," in *Proceedings of the American Control Conference*, vol. 6. IEEE, 2001, pp. 4375–4380.

[71] Y. Huang and P. M. Djuric, "A blind particle filtering detector of signals transmitted over flat fading channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1891–1900, 2004.

[72] R. D. Green and L. Guan, "Tracking human movement patterns using particle filtering," in *IEEE International Conference on Acoustics, Speech, and Signal Processing ((ICASSP) Proceedings*, vol. 3. IEEE, 2003, pp. III–25.

[73] P. Z. Peebles, J. Read, and P. Read, *Probability, random variables, and random signal principles.* McGraw-Hill, 1987, vol. 3.

[74] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[75] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM review*, vol. 45, no. 3, pp. 385–482, 2003.

[76] G. Brooker, *Introduction to Sensors for Ranging and Imaging.* SciTech Publishing, 2009.

[77] J. Choliz, A. Hernandez-Solana, and A. Valdovinos, "Evaluation of algorithms for UWB indoor tracking," in *8th Workshop on Positioning Navigation and Communication (WPNC).* IEEE, 2011, pp. 143–148.

[78] B. Li, N. Wu, H. Wang, D. Shi, W. Yuan, and J. Kuang, "Particle swarm optimization-based particle filter for cooperative localization in wireless net-

works," in *Wireless Communications & Signal Processing (WCSP)*. IEEE, 2013, pp. 1–6.

[79] J.-H. Youn and Y. K. Cho, "Portable ultra-wideband localization and asset tracking for mobile robot applications," *Ultra Wideband. Intech*, 2010.

# Vitae

- Name: Muhammad Mujahid Amin
- Nationality: Pakistani
- Date of Birth: Feburary 12, 1981
- Email: *m.mujahid.amin@gmail.com*
- Permenant Address: 276-D, PCHS, Lahore Pakistan
- Academic Background:
  - MS in Electrical Engineering, King Fahd University of Petroleum and Minerals, KFUPM, Dhahran, KSA.
  - Bachelor of Science in Computer Engineering, COMSATS Institute of Information Technology, Wah Cantt. Pakistan
- Publications
  - **M Amin** and S Al-Ghadhban" Adaptive PSOPF based Target Tracking for UWB Multistatic Radar Sensor Networks", under preparation