A closer look to probabilistic state estimation – case: particle filtering

T. TAŞCI[*], C. ÖZ
Faculty of Computer & Information Sciences, Computer Engineering Department, Sakarya University, Esentepe -54187, Sakarya, Turkey

Particle Filter is a significant member of the group of methods aiming to provide reasonable solutions to the real-world problems by approximating the value of the posterior density function using probabilistic sampling. Particle filtering has been increasingly used by researchers for the last two decades with the advancements occurred in computational resources in order to solve such problems. This paper focuses on Particle Filtering in a way to be a complete tutorial for the beginner researchers by means of providing a quick theoretical framework of Particle Filtering in a step-by-step progressive manner starting with Bayesian Inference as well as providing a stimulating multi-target tracking example problem with solution.

(Received January 8, 2014; accepted May 15, 2014)

Keywords: Probabilistic state estimation, Particle filter, Tracking

1. Introduction

There exist so many systems and services in use today, in order to ensure daily-life requirements in certain areas such as health, education or production; to raise the quality of human life through a great variety of services in transportation, telecommunication or entertainment; to seek answers for eternal questions of existence concerning mankind, living creatures or universe by continuous scientific research initiatives; and to surpass other countries in global competition with regards to economy, politics or military. Disease diagnosis by MR imaging, traveling by plane as an extremely complex system, entertainment experience with motion-sensitive games on game consoles, conducting research with unmanned air or underwater vehicles, effective business management by fleet tracking systems, increasing international deterrence by nuclear powered missiles and a huge number of others may be considered as the examples of such systems and services. Almost all of the systems and daily-life facilitating services mentioned above are actually required some very tough technical problems including single/multiple target tracking, inertial/GPS navigation, audial/visual signal processing and natural events estimation to be solved mostly in real-time with acceptable accuracy and robustness rates. In other words, it is necessary to extract practicable information from the data obtained by several types of sensors such as mechanic, optic, acoustic, haptic, environmental or navigation in order to put the developed systems into force and to operate them efficiently. Likewise, changing needs and conditions, expectations for higher living standards and of course exploration-driven nature of human being imply novel and more capable systems and services to be introduced with each passing day. Hence, modern systems equipped with better hardware and sensors at the same time supported by smarter software gain currency today more than ever. The most significant requirement from the point of architectures for such type of systems is to have a kind of mechanism allowing acquiring information at any time associated with the states of the system in order to take the appropriate action [1]. However, in most situations it is quite difficult to reach this information without any restrictions due to the non-linear and non-Gaussian structures. Within this context, the requested information about the system states which is not directly obtainable is attained via the available sensor measurements, of course, in a distorted form owing to the noise [1]. The noise is included through both dynamic system process and measurements. In this case, it is needed to remove noise from obtained data in order to form a basis for estimating the value of parameters contained by the system states. The act of providing such a basis undoubtedly refers to the well-positioned research field namely “Estimation Theory” and “Filtering” which can be deemed as its sequential and time-dependent application by a majority.

In this paper, theory of estimation and optimal filtering is briefly introduced with historical milestones. Optimal filtering is also handled from Bayesian point of view in detail and the well-known non-linear filtering methods based on Bayesian estimation are outlined with merits and disadvantages as well as with familiar applications. Particle filter (PF) which is one of the most preferred methods of probabilistic estimation in recent years is given both providing the derivation of its equations and the explanation of its algorithm. The study is supported by a particle filter based real-time multi-target tracking example with formulation of the system dynamics and observation models and the implementation results. Actually, there exist a considerable number of books, dissertations, review papers and tutorials related to recursive Bayesian estimation and particle filtering. Although it is not claimed that this study is outstanding with reference to the existing works within the field, several shortcomings found in the examined studies are simply addressed. Some of the papers regarding this study are written by physicians or statisticians in which different terminologies are used and various concepts in theory of probability are emphasized for certain problems unfamiliar to the engineering society. Some other works include deep theoretical definitions and complex derivations which are not made easier to understand the principal topic of interest. In some of the studies, the derivations of PF are given disconnectedly which makes difficult to comprehend the underlying process of evolution. Some other studies, on the other hand, are not included any numerical or tangible examples which prevent to thoroughly get familiar to the probabilistic estimation which has intensively abstract concepts by definition. It is aimed to form a kind of quick start guide for the researchers newly entered into the field of Bayesian non-linear filtering. An expositional statement with a step by step progress starting from basic Bayes' Rule to particle filtering algorithm is adopted throughout this study providing a multi-target tracking problem with its formulation, analysis and solution.

2. Particle filtering theory

a. Estimation theory & filtering

Estimation theory (E.T.) is a field, focusing on estimating observed system parameters based on empirical findings and statistical analysis. Methods of E.T. attempt to predict the value of an unknown magnitude (i.e., position of a ship with respect to radar base) which is not directly accessible in most situations as accurately as possible through noisy observations [1][21]. E.T. has a wide range of possible applications in numerous fields such as signal processing, quality control, telecommunication, software engineering, project management and network intrusion detection [14]. The aim of the E.T. methods is always to construct an optimal (i.e., capable of inferring all available information) and implementable estimator which accepts measurement data and produces the estimation of requested system parameters [22]. Researchers usually regard estimation and filtering in the same meaning such as emphasized in [1] and [23]. While filtering has different definitions in various fields, basically it is a process of separating a mixture of signals into requested and needless fractions [23]. Thus, it is allowed that the requested part of the signal can be evaluated expediently. Filtering in this sense, may be considered as an application of estimation theory. Filtering methods try to reveal behaviors of the observed system so as to provide a basis for further analysis by using measurement data and the transition relationship between consecutive states of the system. The states of the real-world systems are usually hidden and the useful information can only be obtained by processing on observation data corrupted by incongruent noise. Filtering is said to be an essential application of most real-world problems within this context.

Estimation, as a technical field of research, was brought to scientific literature for the first-time by the work of K. F. Gauss regarding the estimation of planet motions. The method of "Least-Squares" which is considered in scientific society as an ancestral method of E.T. was firstly developed by Gauss in the end of 17th century and by Legendre later on in the very beginning of 18th century independent of each other [21]. The second milestone in the field of E.T. is the introduction of the method "Maximum Likelihood Estimation (M.L.E.)" in 1912 by R. A. Fisher who first uses a number of very important field concepts and definitions such as "parameter", "likelihood", "Bayesian" [24]. The M.L.E. method is considered as a basis for many inferential techniques in statistics [25]. Another very important advancement in the field, occurred with the development of the method "Linear Minimum Mean Square Estimation" successively and independently by Kolmogorov in 1941 and Wiener in 1942. The importance of this method, in particular, arises from constituting the foundations of the well-known and perhaps the most remarkable progress in the field: Kalman Filter (KF) [21].

Kalman filter, developed by R. E. Kalman in 1960, traces its origins from the "Least-Squares" method. However, a number of factors such as method's convenience for state estimation of time-varying systems through both continuous and discrete observations, and change of problem formulations with the availability of state-space representation which allows difference and differential rather than integral equations to be used and also provides a basis for recursive algorithms to be effectively applied, have made KF method receives significant attention from researchers of various fields. In addition, KF equations providing a highly practical procedure for digital computer implementation and the method's allowance for real-time estimation open the door for great improvements in the fields of filtering and estimation at least within the subsequent three decades [21]. After the development of the KF., the applications of E.T. has appeared in a wide variety of fields including satellite navigation, planetary orbit estimation, control theory, video tracking systems, geodesic research, neuroscience, machine learning, telecommunications, signal processing, economy, finance, political sciences, operations research and bio-engineering with an ever-increasing use up to early 1990s. Detailed information on Kalman filtering theory, equations with derivations and various applications can be found in [26], [27] and [28].

Kalman filter was first derived as a mean squared error minimizer method. In fact, it was implemented in later years [1] in terms of a different field of research namely Bayesian inference which is the inclusive scope of approaches approved to follow in addressing the problem of object tracking throughout this study.

b. Bayesian inference & optimal Bayesian filtering

Bayesian inference is a general inference framework utilizing the Bayes rule and a number of other methods belong to the theory of probability [14]. The main base of Bayesian inference is the well-known Bayes rule invented by 18th century mathematician and theologian Thomas Bayes. Actually, the general form of Bayes' theorem known today was derived by P. S. Laplace in 1814. Laplace is accepted as one of the pioneer researchers in the field of probability today due to his development of Bayesian inference and applying probability to numerous fields [21]. The fundamental philosophy of the Bayesian approach claims that existing beliefs about an event may be changed depending on obtained new evidence. In the light of this philosophy, it is obviously seen that the problem of state estimation in the sense of time-varying dynamic systems can be readily attacked by Bayesian inference based methods. Besides, the solutions gained through Bayesian inference are viewed as "optimal" for many linear and Gaussian scenarios. In this regard, Bayesian inference was involved in filtering literature by numerous studies namely "Optimal Bayesian Filtering (O.B.F.)" [14,21].

In O.B.F., the state space representation of the problem is replaced with the probabilistic forms of dynamic and observation models that are ideally suited for the Bayesian approach. It is aimed to construct the posterior probability density function (*pdf*) of the state based on all available information with the set of received measurements in using O.B.F. for dynamic state estimation. The posterior *pdf* can be considered as the entire solution to the estimation problem due to the fact that it contains all available statistical information with the feasibility of providing an optimal estimate of the state [29]. It is needed to generate an estimate just after receiving a new measurement for many state estimation problems. In this case, a recursive filter allowing the received data can be processed sequentially is a convenient solution. O.B.F. also called Recursive Bayesian Estimation (R.B.E.) in the literature is a recursive filtering approach having two main stages: Prediction and update. The dynamic model including process noise is used to predict the state *pdf* in the prediction stage and the latest measurement is employed through noisy observation model to refine the predicted state *pdf* in the update stage. These two stages are executed consecutively with Bayes' theorem as the underlying mechanism, until certain conditions have been met.

c. Sub-optimal Bayesian filtering

O.B.F., in fact, can be deemed as a general form of KF where both system dynamics and measurement models are linear and disturbed by Gaussian noise [12]. However, most of the real-world systems are non-linear and also non-Gaussian [5,6]. There exists no closed form or analytical solution of state estimate for such type of systems [2-4,12]. Arising from this situation, various sub-optimal Bayesian approach based filtering methods have appeared in time. A taxonomy of those methods can be found in [4]. Extended Kalman Filter (EKF) is probably one of the earliest and the most well-known methods of such type, based on linearization of the system using Taylor series and simply applying KF solution subsequently [6,7]. EKF. has been widely used for long years in solving problems of diverse fields but it was replaced by other methods due to its poor response for the case of non-linearity with high degrees [7,10]. Gaussian Sum Filter (GSF) is another method attacking non-linear estimation problems by using local linearization similar to EKF. Actually, the GSF is a method executing a set of EKF in parallel [8]. The underlying principle of the GSF is to approximate the requested posterior density by a weighted sum of Gaussian density functions [8]. The performance of GSF is based on the selection of the number of mixture components and their weights depending on measurements. It has been shown that the filtering estimates produced by GSF are more unbiased in comparison with EKF [5]. The use of GSF is quite reasonable when the posterior density is multi-modal, on the other hand, the method has a disadvantage of keeping the weights of Gaussian mixture as constant while propagating the uncertainty through the nonlinear system and updating the weights only in the presence of measurement data [5]. The Unscented Kalman Filter (UKF) is a relatively new approximate filtering method with respect to EKF and GSF. UKF employs a statistical linearization technique based on picking a set of sample points around the mean and propagating these points through the non-linear system. A faster and computationally inexpensive solution with more accurate estimates is produced by UKF with respect to both EKF and GSF [5][9]. Another group of approximate Bayesian non-linear filtering methods is the collection of grid based methods (GBMs) using numerical integration to solve multi-dimensional integrals and accordingly to approximate the posterior *pdf* of the state [5]. The main drawback of GBMs is the dramatic increase of the computational cost in the existence of high-dimensional state space. As a consequence, the performance of GBMs in general is open to question due to the high-dimensionality exposed by almost any contemporary non-linear system [2][10].

Another branch of methods to deal with non-linear state estimation from the point of Bayesian philosophy is sampling based approaches. Such approaches, in fact, are various implementations of recursive Bayesian filter by Monte Carlo (MC) simulation technique [2]. MC simulation technique traces its origin from a simple random number generation operation on a roulette by 1770s, however, it has drawn the attention of physics, statistics and engineering communities in sequence starting from the Second World War to subsequent few years [2].

The basic principle of MC based sampling methods is to represent the requested posterior density function by a set of weighted random samples [13]. It can obviously be seen that in such a representation, using more random samples inherently provides a more approximate version of usual posterior *pdf* and a closer prediction to the optimal Bayesian estimate. MC methods have both advantages and limitations. The main advantage of MC methods over other approximation methods is the reduction of the approximation error variance regardless of the dimension of the state space. MC methods, on the other hand, expose two fundamental disadvantages: difficulty of sampling high-dimensional probability distribution and increasing computational complexity in case of sampling is possible [14]. As might be expected, the latter is overcome by ever-increasing computational resources. The problem of sampling difficulty by MC methods, on the other hand, conduced to the improvement of two main classes of algorithms namely Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC).

MCMC methods such as Metropolis-Hastings and Gibbs methods are based on sampling directly from a target distribution by defining a proper Markov Chain such that its stationary distribution is identical to the target distribution [35]. A detailed explanation of MCMC based sampling methods with applications can be found in [17]. SMC methods are referred in the literature with different names such as bootstrap filtering, the condensation algorithm, particle filtering and survival of the fittest [2]. Particle Filtering, as the preferred definition for SMC approach is the primary interest of this study. PF is based upon the idea of sampling from a proposal distribution rather than the target distribution which may not be convenient to sample from [13]. The proposal distribution (also called importance function) usually allows drawing samples and evaluating the target *pdf* for given samples. Thus, it becomes possible to obtain samples with importance weights with respect to the target *pdf*. Because the successive steps of the algorithm strongly rely on the samples obtained by importance sampling, the proper selection of the importance function in PF is crucial [2]. Otherwise, useless solutions can simply be encountered.

PF has been formally introduced by [15]. Since then, numerous successful applications of PF have been presented in various fields such as signal processing, robotics, and computer vision [10][31]. Although there exist a number of improved variants of PF in the literature today, such as Unscented PF [18], Ensemble PF [19] and Rao-Blackwellized PF [20], the basic PF is addressed in this study in accordance with the aim of making easier to comprehend the recursive Bayesian estimation and subsequently probabilistic sampling based estimation.

In particle filtering, usually, no assumptions are made about the problem addressed [10]. The problem can be associated with a non-linear and non-Gaussian dynamic system. However, two main issues should be taken into consideration: (i) high dimensionality of the state space, (ii) dynamics and observation models of the system [3,35]. While the first issue is highly overcome by today's computational resources, resolving the second issue strongly depends on difficulty of the problem and built methodology for the solution including received measurements and extracted information. Acceptable solutions can be obtained by using convenient system dynamics and observation models.

3. Particle filter derivation

a. Dynamic systems and state-space representation

It is firstly required to learn definition of a system from a technical perspective in order to comprehend dynamic system concept. A system can be defined as an entirety consisting of interrelated components. If the behaviors of the system change in time then the system is described as dynamic. Process can be stated as the evolution of a dynamic system in time. A dynamic system can be linear or non-linear in terms of relationship between its consecutive states. Behaviors of a dynamic system are observed by generating mathematical models depending on this transition relationship. It is mostly required to examine extremely small changes of the value of some parameters with respect to others due to the requirement of finding more accurate and more general solutions. Within this context, dynamic systems that are mostly continuous in time are generally modelled by changes in velocity and acceleration of the state vectors. In such situations, mathematical model of the observed dynamic system is obtained by using first and second order derivatives for velocity and acceleration changes respectively. The states of some specific dynamic systems are required to be known in certain time steps. The next time step is modeled as a function of current time step for such systems and difference equations are employed in models in place of differential equations.

Almost all systems in real-world exist as dynamic systems, have non-linear state transition functions and contain high level uncertainties (i.e., non-Gaussian noise) associated with their parameters. Dynamic systems are firstly needed to be represented mathematically in order for performing a state estimation. State-space models are the most common type of representation for this purpose [30]. There are two models in state-space representation: dynamic and observation models.

Dynamic model defines the evolution of state vector in time which contains estimation of the requested system parameters.

$$x_k = f_{k-1}\left(x_{k-1}, v_{k-1}\right),\ k > 0 \tag{1}$$

where x_k is the state vector to be estimated; k is the time step; f_{k-1} is a non-linear function; and v_{k-1} is the process noise. Observation model describes the relationship between state vector and received measurements.

$$z_k = h_k\left(x_k, w_k\right),\ k > 0 \tag{2}$$

where z_k is the measurement vector obtained at time step k; h_k is the measurement function; and w_k is the measurement noise.

b. Probabilistic representation of dynamic systems

Probability based representation of dynamic systems is a way applied particularly in case of Bayesian methods are at issue. In such situations, Bayesian methods are employed so as to provide approximate solutions by stochastic modelling of system dynamics and inferring statistical properties of requested system parameters. Hence, dynamic processes and the uncertainties in sensor measurements are assumed as stochastic processes such as speech signal, digital computer data, or noise while building probabilistic models. A stochastic process is a collection of random variables, $\{X_k\}$ indexed by time k which is used to express the time evolution of a statistical event according to probabilistic laws. In fact, it is involved enormous number of random variables in working with stochastic processes. The number of the random variables may be finite or infinite depending on the parameter space of processes under evaluation. Probability distribution of a random variable is the main property characterizing itself. The probabilistic characterization of a random process is derived from the joint probability distribution of underlying random variables. Therefore, joint distribution and some statistical parameters such as mean, covariance and correlation of random variables are investigated for dynamic systems modelled as stochastic process allowing computerized solutions. Within this context, for the case where the state of the system is assumed as a Markov stochastic process, the posterior *pdf* of a dynamic system can be expressed as in Eq.(3):

$$p(x_{0:k}) \sim p(x_0)\prod_{i=1}^{k} p\left(x_i | x_{i-1}\right) \tag{3}$$

where $p(x_0)$ is the prior distribution at time step 0, and $\sim$ symbol characterizes the sampling process of independent and identically distributed random variables from a sequence. Similarly, the observation model is constructed as in Eq.(4) where $p\left(z_k | x_k\right)$ is the observation likelihood.

$$p\left(z_{1:k} | x_{0:k}\right) \sim \prod_{i=1}^{k} p\left(z_i | x_i\right) \tag{4}$$

Through this representation, behaviors of the dynamic system are examined and values of the requested parameters are estimated probabilistically by modelling state transition and measurement equations as conditional probabilities.

c. Recursive Bayesian estimation

Particle filter which is also known as Sequential Monte Carlo, is a generic name for the methods implementing an improved version of importance sampling recursively in order to approximate posterior *pdf* and providing a basis for analyzing the states of dynamic systems by employing recursive Bayesian estimation.

All available information regarding system dynamics should be included into the probabilistic estimation model while intending to estimate the parameters of a discrete-time dynamic system with a reasonable accuracy level. In such systems, measurements can only be obtained in certain time steps even though the system evolves perpetually. Current measurement information is required to be used in order for estimating the parameters of the next state of dynamic system without receiving a new measurement. It is obviously needed a recursive mechanism for this type of usage. Such a mechanism can be obtained using Bayes' Rule with Markov chain assumptions.

A discrete-time dynamic system with finite states is modeled using discrete time Markov process forming a stationary distribution in time and having its current state depends only its previous state. The two main assumptions of discrete time Markov process are expressed as in Eq.(5) and Eq.(6) respectively.

(i) The current state of the system is detached from all past states and measurements received at those time steps except the previous system state.

$$p\left(x_k | x_{0:k-1}, z_{1:k-1}\right) = p\left(x_k | x_{k-1}\right) \tag{5}$$

where $x_k \in \mathrm{R}^n$ represents the state vector and $z_k \in \mathrm{R}^m$ is the measurement vector at time step k.

(ii) The measurement information received at current time step is free of all past states of the system and measurements obtained at that time steps, however it depends on the current state.

$$p\left(z_k | z_{1:k-1}, x_{0:k}\right) = p\left(z_k | x_k\right) \tag{6}$$

The joint distribution of random variables representing states of a discrete-time dynamic system in each time step are expressed by using Bayes Theorem, chain rule of Probability Theory and Markov process assumptions in R.B.E.

In the first step, the joint distribution of random variables is written in terms of substantial state transition probabilities. In the second step, the posterior *pdf* of discrete-time dynamic system is stated using Bayes Theorem as below:

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B)} \Rightarrow p\left(x_{0:k} | z_{1:k}\right) = \frac{p\left(z_{1:k} | x_{0:k}\right) \times p\left(x_{0:k}\right)}{p\left(z_{1:k}\right)} \tag{7}$$

where $p\left(x_{0:k} | z_{1:k}\right)$ is the joint probability distribution of all states given all measurements up to time step k . $p\left(z_{1:k}|x_{0:k}\right)$ is defined as likelihood of all measurements given all system states up to time step k .

$$p\left(z_{1:k}|x_{0:k}\right)=p\left(z_k,z_{1:k-1}|x_{0:k}\right) \tag{8}$$

Eq.(8) can be formed using Chain Rule of Probability Theory. $p\left(z_k,z_{1:k-1}|x_{0:k}\right)$ is substituted for $p\left(z_{1:k}|x_{0:k}\right)$ in Eq.(7) and Eq.(9) is obtained.

$$p\left(x_{0:k}|z_{1:k}\right)=\frac{p\left(z_k,z_{1:k-1}|x_{0:k}\right)\times p\left(x_{0:k}\right)}{p\left(z_{1:k}\right)} \tag{9}$$

$$p\left(z_k,z_{1:k-1}|x_{0:k}\right)=p\left(z_k|z_{1:k-1},x_{0:k}\right)\times p\left(z_{1:k-1}|x_{0:k}\right) \tag{10}$$

Similar to forming Eq.(8), Eq.(10) can also be written by using Chain Rule. Eq.(11) is constructed by substituting Eq.(10) in Eq.(9).

$$p(x_{0:k}|z_{1:k})=\frac{p(z_k|z_{1:k-1},x_{0:k})\times p(z_{1:k-1}|x_{0:k})\times p(x_{0:k})}{p(z_{1:k})} \tag{11}$$

$$p\left(z_{1:k-1}|x_{0:k}\right)=\frac{p\left(x_{0:k}|z_{1:k-1}\right)\times p\left(z_{1:k-1}\right)}{p\left(x_{0:k}\right)} \tag{12}$$

Using Bayes Rule, an equality can ordinarily be constituted as in Eq.(12). The expression $p\left(z_{1:k-1}|x_{0:k}\right)$ in Eq. (11) is replaced by its equivalent in Eq.(12) and Eq.(13) is achieved as below.

$$p(x_{0:k}|z_{1:k})=\frac{p(z_k|z_{1:k-1},x_{0:k})\times p(x_{0:k}|z_{1:k-1})\times p(z_{1:k-1})\times p(x_{0:k})}{p(z_{1:k})\times p(x_{0:k})} \tag{13}$$

Eq.(14) is obtained by performing the required simplifications on Eq.(13).

$$p\left(x_{0:k}|z_{1:k}\right)=\frac{p\left(z_k|z_{1:k-1},x_{0:k}\right)\times p\left(x_{0:k}|z_{1:k-1}\right)\times p\left(z_{1:k-1}\right)}{p\left(z_{1:k}\right)} \tag{14}$$

$$p\left(z_{1:k}\right)=p\left(z_k|z_{1:k-1}\right)p\left(z_{1:k-1}\right) \tag{15}$$

Eq.(15) is formed by using chain rule one more time. $p\left(z_{1:k}\right)$ in Eq.(14) is replaced by its equivalent in Eq.(15) and Eq.(16) is obtained.

$$p(x_{0:k}|z_{1:k})=\frac{p(z_k|z_{1:k-1},x_{0:k})\times p(x_{0:k}|z_{1:k-1})\times p(z_{1:k-1})}{p(z_k|z_{1:k-1})\times p(z_{1:k-1})} \tag{16}$$

Eq.(17) is formed by simplificating Eq.(16).

$$p\left(x_{0:k}|z_{1:k}\right)=\frac{p\left(z_k|z_{1:k-1},x_{0:k}\right)\times p\left(x_{0:k}|z_{1:k-1}\right)}{p\left(z_k|z_{1:k-1}\right)} \tag{17}$$

In this step, Eq.(18) is achieved by using the second assumption of Markov Process given in Eq.(6).

$$p\left(x_{0:k}|z_{1:k}\right)=\frac{p\left(z_k|x_k\right)\times p\left(x_{0:k}|z_{1:k-1}\right)}{p\left(z_k|z_{1:k-1}\right)} \tag{18}$$

$$p\left(x_{0:k}|z_{1:k-1}\right)=p\left(x_k|x_{0:k-1},z_{1:k-1}\right)\times p\left(x_{0:k-1}|z_{1:k-1}\right) \tag{19}$$

Eq.(19) can be written by employing Chain Rule. The expression $p\left(x_{0:k}|z_{1:k-1}\right)$ in Eq.(18) is substituted by its equivalent in Eq.(19) and Eq.(20) is obtained.

$$p\left(x_{0:k}|z_{1:k}\right)=\frac{p\left(z_k|x_k\right)\times p\left(x_k|x_{0:k-1},z_{1:k-1}\right)\times p\left(x_{0:k-1}|z_{1:k-1}\right)}{p\left(z_k|z_{1:k-1}\right)} \tag{20}$$

Eq.(21) is derived by applying the first assumption of Markov Process given in Eq.(5).

$$p\left(x_{0:k}|z_{1:k}\right)=\frac{p\left(z_k|x_k\right)\times p\left(x_k|x_{k-1}\right)\times p\left(x_{0:k-1}|z_{1:k-1}\right)}{p\left(z_k|z_{1:k-1}\right)} \tag{21}$$

$$p\left(x|z\right)=\frac{p\left(z|x\right)\times p\left(x\right)}{p\left(z\right)}\Rightarrow p\left(z\right)=\int p\left(z|x\right)p\left(x\right)dx \tag{22}$$

According to Bayes Rule, the denominator $p\left(z\right)$, also called evidence, in Eq.(22) is in fact the marginal probability distribution of the nominator terms and it has no effect on the value of posterior probability distribution $p\left(x|z\right)$. Thus, $p\left(z\right)$ is considered as a normalizing constant.

$$p\left(z_k|z_{1:k-1}\right)=\int p\left(z_k|x_k\right)p\left(x_{0:k}|z_{1:k-1}\right)dx_k \tag{23}$$

It is therefore, the expression $p\left(z_k|z_{1:k-1}\right)$ in Eq.(21) is replaced by its equivalent given in Eq.(23) and Eq.(24) is achieved.

$$\eta=\frac{1}{p\left(z_k|z_{1:k-1}\right)}=\frac{1}{\int p\left(z_k|x_k\right)p\left(x_k|z_{1:k-1}\right)dx_k} \tag{24}$$

The equivalent of normalizing constant η in Eq.(24) is substituted in Eq.(21) and Eq.(25) is formed.

$$p\left(x_{0:k}|z_{1:k}\right)=\eta\times p\left(z_k|x_k\right)\times p\left(x_k|x_{k-1}\right)\times p\left(x_{0:k-1}|z_{1:k-1}\right) \tag{25}$$

Eq.(25) can be ordered as in Eq.(26) where the symbol $\propto$ means “up to a normalizing constant”.

$$p\left(x_{0:k}|z_{1:k}\right)\propto p\left(z_k|x_k\right)\times p\left(x_k|x_{k-1}\right)\times p\left(x_{0:k-1}|z_{1:k-1}\right) \tag{26}$$

where $p\left(x_{0:k}|z_{1:k}\right)$ is the posterior *pdf* of the system and $p\left(z_k|x_k\right)$ is the likelihood of observation given system state at time step k; $p\left(x_k|x_{k-1}\right)$ is called transition(temporal) prior; and $p\left(x_{k-1}|z_{1:k-1}\right)$ is the posterior *pdf* of the system at previous time step. With this formulation in Eq.(26), a recurrence is acquired so as to the posterior *pdf* of the system at any time step can be obtained using the previous posterior *pdf* recursive Bayesian estimation consists of two phases: prediction and update. In prediction phase, $p\left(x_k|z_{1:k-1}\right)$ distribution is predicted without receiving the measurement at actual time step. The value of $p\left(x_{0:k}|z_{1:k}\right) \propto p\left(z_k|x_k\right) \times p\left(x_{0:k}|z_{1:k-1}\right)$ expression is calculated using measurement likelihood at actual time step and the posterior *pdf* of the previous time step is updated in update phase.

d. Problem formulation with particle filter

The last form of Eq.(18) is obtained as Eq.(28) by substituting Eq.(27) which is also known as Chapman-Kolmogorov equation and Eq.(23).

$$p\left(x_{0:k}|z_{1:k-1}\right) = \int p\left(x_k|x_{k-1}\right) p\left(x_{0:k-1}|z_{1:k-1}\right) dx_{k-1} \qquad (27)$$

$$p\left(x_{0:k}|z_{1:k}\right) = \frac{p\left(z_k|x_k\right) \times \int p\left(x_k|x_{k-1}\right) p\left(x_{0:k-1}|z_{1:k-1}\right) dx_{k-1}}{\int p\left(z_k|x_k\right) p\left(x_{0:k}|z_{1:k-1}\right) dx_k} \quad (28)$$

Eq.(28) is the final form of R.B.E. It is mostly impossible to calculate the integrals in this equation analytically except very special conditions and strong assumptions. Therefore, the common way in the literature is to approximate the posterior *pdf* $p\left(x_{0:k}|z_{1:k}\right)$ through various methods such as EKF, UKF, GSF, MCMC or PF which is the focal point of this study.

Problem formulation with PF requires MC simulation concept and importance sampling method to be known. MC is a collection of methods aiming to produce approximate results for the values of integrals which are impossible to evaluate analytically. MC simulation based methods try to find the approximate values of the integrals as shown in Eq.(29) below.

$$I = \int f\left(x\right) dx = \int g\left(x\right) p\left(x\right) dx \qquad (29)$$

where $f\left(x\right)$, $g\left(x\right)$ are any functions of x and $p\left(x\right)$ is the *pdf* of x. It is assumed that is possible to obtain independent random samples from the associated *pdf*. According to the Strong Law of Large Numbers (Eq.(30)), the mean of a known random variable sequence approximates to its expected values while the number of samples goes to infinity. Thus, it is verged on the value of the integral under evaluation while more samples are available.

$$\frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right) \Rightarrow E\left[f\left(x\right)\right] = \int_a^b f\left(x\right) p\left(x\right) dx \qquad (30)$$

In this context, the expected value of $x_{0:k}$ is expressed as in Eq.(31) below:

$$E_{p\left(x_{0:k}|z_{1:k}\right)}\left[g\left(x_{0:k}\right)\right] = \int g\left(x_{0:k}\right) \times p\left(x_{0:k}|z_{1:k}\right) \qquad (31)$$

where $p\left(x_{0:k}|z_{1:k}\right)$ is the *pdf* of x and $(.)$ is a function of $x_{0:k}$. This expected value is approximated by MC simulation based methods as in Eq.(32).

$$E_{p\left(x_{0:k}|z_{1:k}\right)}\left[g\left(x_{0:k}\right)\right] = \frac{1}{N}\sum_{i=1}^{N} g\left(x_{0:k}^{(i)}\right) \qquad (32)$$

According to the Central Limit Theorem, sum of enormous number of random variables approximate to Gaussian distribution as shown in Eq.(33) below:

$$\frac{\sum_{i=1}^{N} x_i - N \times \mu}{\sigma \times \sqrt{N}} \rightarrow \aleph\left(0;1\right),\ N \rightarrow \infty \qquad (33)$$

where x_i is an independent random variable sequence, μ is the mean and σ^2 is the finite variance. It can be seen in this situation that the approximation of MC simulation is suitable and unbiased.

Accept-Reject, Inverse CDF Transform, Control Variable and Importance Sampling can be considered among MC simulation methods. However, importance sampling method as the base of particle filtering is exclusively explained in this study. It is primarily required to be familiar with the importance sampling method in order to be comprehended particle filtering. Importance sampling method approximates the value of target integral as shown in Eq.(34):

$$I = \int_{\mathrm{R}^n} f\left(x\right) \pi\left(x\right) dx = \int_{\mathrm{R}^n} f\left(x\right) \frac{\pi\left(x\right)}{q\left(x\right)} q\left(x\right) dx \qquad (34)$$

where $\pi\left(x\right)$ is a *pdf* impossible to sample and $q\left(x\right)$ is called proposal distribution or importance function implying $\pi\left(x\right) > 0 \Rightarrow q\left(x\right) > 0, \forall x \in \mathrm{R}^n$ condition.

Unnormalized and normalized importance weights are expressed in Eq.(35) and Eq.(36) respectively as below:

$$f_N = \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right) w\left(x_i\right),\ w\left(x_i\right) = \frac{\pi\left(x_i\right)}{q\left(x_i\right)} \qquad (35)$$

$$w\left(x_i\right) \propto \frac{\pi\left(x_i\right)}{q\left(x_i\right)} \qquad (36)$$

where $w\left(x_i\right)$ is the importance weight sequence. It can be seen in Eq.(36) that normalized importance weights is proportional to the ratio of target *pdf* over importance function.

In particle filtering, the target posterior *pdf* is approximated by Eq.(37):

$$p\left(x_{0:k}|z_{1:k}\right) \approx \sum_{i=1}^{N} \tilde{w}_k^{(i)} \delta\left(x_{0:k} - x_{0:k}^{(i)}\right) \qquad (37)$$

where $\delta(.)$ is the Dirac-delta function, $x_{0:k}^{(i)}$ is the i th particle and $\tilde{w}_k^{(i)}$ is the normalized weight of that particle at time step k as shown in Eq.(38).

$$\tilde{w}_k^{(i)} = \frac{w\left(x_{0:k}^{(i)}\right)}{\sum_{j=1}^{N} w\left(x_{0:k}^{(j)}\right)} \tag{38}$$

It is started with Eq.(39) in particle filter problem formulation by omitting the normalizing constant $p\left(z_k|z_{1:k-1}\right)$ in Eq.(21) obtained by recursive Bayesian estimation (Eq.(21)).

$$p\left(x_{0:k}|z_{1:k}\right) \propto p\left(z_k|x_k\right) \times p\left(x_k|x_{k-1}\right) \times p\left(x_{0:k-1}|z_{1:k-1}\right) \tag{39}$$

In this step, it is required to select an importance function (proposal distribution). If this selection is performed as in Eq.(40), then the normalized importance weights can be calculated by using Eq.(41).

$$q\left(x_{0:k}|z_{1:k}\right) \triangleq q\left(x_k|x_{0:k-1}, z_{1:k}\right) \times q\left(x_{0:k-1}|z_{1:k-1}\right) \tag{40}$$

$$\tilde{w}\left(x_k^{(i)}\right) \propto \frac{p\left(x_{0:k}|z_{1:k}\right)}{q\left(x_{0:k}|z_{1:k}\right)} = \frac{p\left(z_k|x_k\right) \times p\left(x_k|x_{k-1}\right) \times p\left(x_{0:k-1}|z_{1:k-1}\right)}{q\left(x_k|x_{0:k-1}, z_{1:k}\right) \times q\left(x_{0:k-1}|z_{1:k-1}\right)} \tag{41}$$

It can be seen when Eq.(41) is examined that it contains important weights associated with sequential time steps. The expressions $p\left(x_{0:k-1}|z_{1:k-1}\right)$ in the nominator and $q\left(x_{0:k-1}|z_{1:k-1}\right)$ in the denominator are in fact the posterior *pdfs* of the previous time step. Therefore, Eq.(41) can be re-written after a simple arrangement as below:

$$\tilde{w}\left(x_k^{(i)}\right) \propto \tilde{w}\left(x_{k-1}^{(i)}\right) \times \frac{p\left(z_k|x_k\right) \times p\left(x_k|x_{k-1}\right)}{q\left(x_k|x_{k-1}, z_k\right)} \tag{41*}$$

Particle Filtering in the literature is performed based on various methods. The determinative property of distinctness of those methods is the assumption they assigned. One of the simplest but on the other hand reasonably effective methods for applications especially with real-time constraint is the Bootstrap PF or Condensation algorithm. In BPF, the expression $q\left(x_k|x_{k-1}, z_k\right)$ in Eq.(41) is selected as the equivalent of the temporal prior as below:

$$q\left(x_k|x_{k-1}, z_k\right) \triangleq p\left(x_k|x_{k-1}\right) \tag{42}$$

where the symbol $\triangleq$ is meant as selection operation. Thus, the particle filter problem formulation based on recursive Bayesian estimation is obtained as in Eq.(43):

$$\tilde{w}\left(x_k^{(i)}\right) \propto \tilde{w}\left(x_{k-1}^{(i)}\right) \times p\left(z_k|x_k\right) \tag{43}$$

In typical particle filtering, there are three main steps. In the first step, certain numbers of particles with initial values are created for hidden parameters depending on the problem to be solved or the dynamic system under evaluation. Each of these particles is constructed as a random variable array (i.e. random vector) consisting of requested system parameters and associated with a weight which is equal to $1/N$ where N indicates the number of particles. For some convenient applications, the initial weights of the particles may be determined in a way to be different from each other using existing knowledge about the system before receiving any measurements.

The second step of the PF algorithm is called importance sampling step. This is the most significant step of the algorithm affecting the performance. In this step, the particles are sampled from the importance function selected (i.e., the particles are moved through the dynamic model) and the weights are calculated according to the observation or measurement model developed for the problem. It is very crucial to make use of accurate operating dynamic and measurement models in this step, because the particles containing the requested system parameters get their new values according to the dynamic model and the weights of particles are obtained by inferring suitable information regarding the system depending on how likely the measurements are given the system states. Improper formation of dynamic and/or measurement models definitely leads to unwanted results. For each time step or state of the observed dynamic system, there exists a particle set. Particles having larger weights are retained with increasing their numbers in proportion to their weights; on the other hand particles with smaller weights are discarded in each time step. Depending on the solution criteria of the problem tackled, the whole set or a single particle is evaluated after normalizing the weights in obtaining the values of requested system parameters using some well-known techniques such as mean square error (MSE).

The last step of the PF algorithm contains a resampling process. This step is largely required due to the sample impoverishment arising from recurring weight obtainment process which causes a group of particles have negligible weights in time [12]. The impoverishment case is precluded by resampling the particles based on a pre-defined threshold called degeneration coefficient. If the number of effective particles becomes smaller than this threshold, a new particle set having equally-weighted particles is sampled from the current particle set by using some techniques such as multi-nominal, stratified, residual and systematic resampling schemes. Thus, it is guaranteed that both effective and marginal particles are allowed to be in existence proportionately. A performance comparison for common resampling techniques can be found in [32] and [34]. A graphical illustration of particle filtering algorithm is given in Fig. 1.

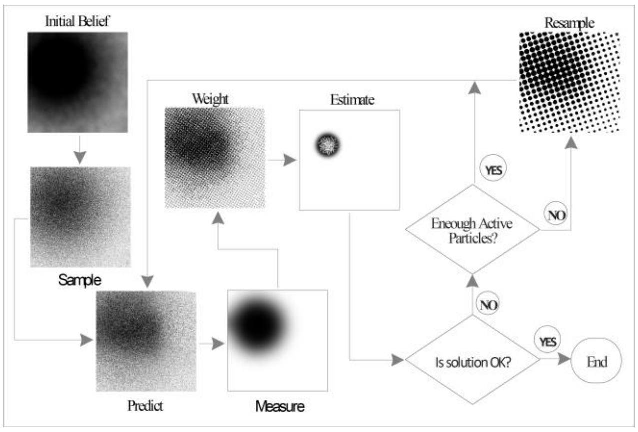


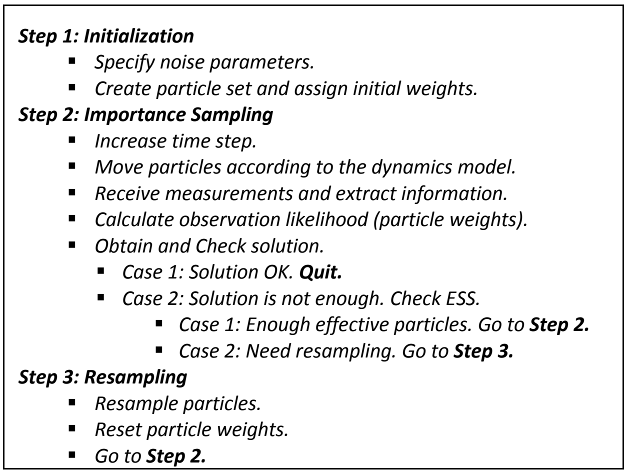
Fig. 1. Graphical illustration of PF algorithm.

Particle Filter is capable of supporting multiple hypotheses theoretically. This means that a posterior *pdf* with more than a single mode can be approximated by using PF. In particle filtering on the other hand, there exist no restrictive assumptions with regards to the operation of the dynamic system. State transitions may be non-linear and the process and measurement noises do not have to be Gaussian. Within this context, a well-structured PF based method may provide acceptable solutions to most of the real-world problems including real-time tracking.

It can clearly be seen as Eq.(43) is examined that the importance weights in each time step depend on the weights of previous time step and the current likelihood represented by $p\left(z_k|x_k\right)$. The likelihood is defined as the value of the extent to which the measurements received via available sensors in each time step suit the actual state dynamics. The acceptable solution to the problem under investigation is approximated more as this suitability rate is increased. Besides that, it can be understood that the importance weights have to get possession of some initial values.

4. PF based multi-ball tracking: an example

Formulation of an example problem is presented in this section with solution steps within the scope of particle filtering. It is aimed to track the position of three balls having different diameters and moving unrestrainedly on a static background. The problem is so called multi-ball tracking (M-BT). The main purpose to prefer such a problem is to make easier of understanding the operation of PF algorithm. The solution of M-BT problem is provided by implementing the particle filter sequential importance resampling (PF SIR) algorithm given below:



Step 1: Initialization
- *Specify noise parameters.*
- *Create particle set and assign initial weights.*

Step 2: Importance Sampling
- *Increase time step.*
- *Move particles according to the dynamics model.*
- *Receive measurements and extract information.*
- *Calculate observation likelihood (particle weights).*
- *Obtain and Check solution.*
    - *Case 1: Solution OK.* ***Quit.***
    - *Case 2: Solution is not enough. Check ESS.*
        - *Case 1: Enough effective particles. Go to* ***Step 2.***
        - *Case 2: Need resampling. Go to* ***Step 3.***

Step 3: Resampling
- *Resample particles.*
- *Reset particle weights.*
- *Go to* ***Step 2.***

Problem-oriented particle structure containing the requested system parameters is needed at first in particle filtering. A $M \times N \times O$ dimensional matrix is formed as particle to represent candidate solutions for M-BT problem. In such a particle structure, M is the dimension of state vector, N is the number of particles and O is the number of objects to be tracked.

The state vector of MB-T problem is defined in Eq.(44) where p_x, p_y are the horizontal and vertical positions of an arbitrary ball and v_x, v_y are the velocities through those directions.

$$X = \left[p_x \quad p_y \quad v_x \quad v_y \right]^T \tag{44}$$

In particle filtering, it is required to form dynamics and observation models fitting the target problem. The evolution of particles in time is expressed by dynamics model. Observation model, on the other hand, is used to extract reference information from the data obtained by using available sensor measurements. Similar to many real-world problems, noise is added to both models in M-BT problem due to the uncertainties in both system process and measurements. Several motion models may be applied in non-linear tracking problems such as constant velocity; constant acceleration, Brownian or more advanced ones [33]. Dynamics model in M-BT problem given in Eq.(45) is selected as constant velocity motion model with Gaussian noise for simplicity.

$$\begin{bmatrix} p_x(k) \\ p_y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} p_x(k-1) \\ p_y(k-1) \\ v_x(k-1) \\ v_y(k-1) \end{bmatrix} + \aleph(0;\sigma_{x,y}) \tag{45}$$

In Eq.(45), k is the time step and $\aleph(0;\sigma_{x,y})$ is the Gaussian process noise with zero mean and covariance σ_x and σ_y through horizontal and vertical directions respectively. The planar position of each ball at each time step in M-BT problem is obtained via this dynamics motion model. For instance, the vertical position of a ball at time step k is calculated by adding its vertical position and velocity at time step $k-1$ and the process noise. The initial positions of the particles are specified before the first time step and the dynamics model is used to make a prediction at each time step without receiving any sensor measurements.

In M-BT example, the initial positions of each particle set for each balls are specified at the center of the background stage with different noise covariance.

The video containing the free motions of the three balls is used as the observation resource in M-BT example. Each frame of this video is accepted as a time step. The planar positions of the balls are obtained by basic image processing techniques for each frame. However, upon the assumption of the sensory information is not certain, Gaussian measurement noise is added to the detected positions of the balls and the observation model is formed with Eq.(46).

$$\begin{bmatrix} p_x^{(k)} \\ p_y^{k} \end{bmatrix} = \begin{bmatrix} c_x^{(k)} \\ c_y^{(k)} \end{bmatrix} + \aleph(0;\rho) \tag{46}$$

$c_x^{(k)}$ and $c_y^{(k)}$ in this observation model are the horizontal and vertical positions of a single ball at time step k. These positions are obtained by using basic *thresholding* and MATLAB’s Image Processing Toolbox command *RegionProps:Centroid*. Distinction of the balls from each other is provided by sorting and matching the different area information acquired with the use of *RegionProps:Area* command. $\aleph(0;\rho)$ here is Gaussian measurement noise with zero mean and covariance ρ. Planar positions of the balls at each time step are obtained using this observation model in M-BT example.

In order to calculate observation likelihood in particle filtering, it is necessary to find a similarity measure or a kind of performance metric. In video based tracking, comparison of tracked target trajectory and trajectory of the ground truth is typically accepted as a performance metric [16]. In M-BT problem, the Euclidean distance between ball positions obtained using observation model and the generated particle positions by dynamics model is employed as the similarity measure at each time step. The similarity measure equation is given by Eq.(47).

$$L^{(k)} = \left\{ \left(p_{x_i}^{(k)} - c_x^{(k)} \right)^2 + \left(p_{y_i}^{(k)} - c_y^{(k)} \right)^2, i = 1..N \right\} \tag{47}$$

where $L^{(k)}$ is the distance matrix at time step k, $p_{x_i}^{(k)}$ and $p_{y_i}^{(k)}$ are the horizontal and vertical positions of i th particle and N is the number of particles. Calculation of $L^{(k)}$ is repeated for all balls, however the object index is omitted here for ease of notation. The calculated distances may be very far from each other depending on the problem tackled. Therefore, it is typically applied a normalization process in order to keep the calculated values within an interval as well as to instantly observe the post-process changes and to provide a basis for easier subsequent processes. In M-BT example, the calculated distances are normalized into $[0,1]$ interval.

Observation likelihood values or particle weights in other terms in M-BT problem are obtained from the distance matrix so as to ensure inverse proportionality between ball-to particle distance and particle weight. Particle weights are calculated by Eq.(48) which produces high likelihood for short distances and vice versa.

$$Lh^{(k)} = e^{\frac{-\hat{L}_i^{(k)}}{2\rho}}, i = 1..N \tag{48}$$

In Eq.(48), $Lh^{(k)}$ is defined as the particle weights matrix at time step k, $\hat{L}_i^{(k)}$ is the normalized distance of i th particle, N is the number of particles and ρ is the measurement noise covariance. In M-BT example, different ρ values are selected for each ball object so as to investigate the effect of measurement noise to the results.

After obtaining the particle weights, the posterior *pdf* is reached in proportion to those weights in the previous time step in PF algorithm. (See. Eq.(43)). The expected solution for M-BT problem is to get an estimation of planar position for each ball at each time step. The mean, weighted mean of the position values carried by the particles or the position values of the particle having the highest weight may be accepted as the solution of the problem. In M-BT problem, the first one of these options is selected.

The steps of PF algorithm may be repeated up to end of a pre-defined duration or until reaching a certain threshold value. In M-BT example, the algorithm is applied for 120 time steps and results are obtained according to this condition.

The last but an optional step of PF algorithm is called “resampling” step. As it is mentioned in Section 3, resampling is a step applied in case of the particles are increasingly exposed to degeneration and become ineffective during the simulation time. In resampling step, the insignificant particles are discarded by providing the higher-weighted particles are sampled more in conjunction with the occurrence of a situation such that the effective particle size (ESS) N_{eff}, becomes smaller than the degeneration coefficient N_{th}. Once this criterion $(N_{eff} < N_{th})$ is met, a new set of particles is reproduced from the existing set using some resampling techniques in which the samples are regarded in proportion to their weights. Given in Eq.(49) and Eq.(50) respectively, the values used for both N_{eff} and N_{th} in M-BT example, are same as the ones shown commonly in the particle filtering literature.

$$N_{eff}^{(k)} = \frac{1}{\sum_{i=1}^{N} \left(Lh_i^{(k)} \right)^2} \tag{49}$$

$$N_{th} = 2/3N \tag{50}$$

N is the number of particles, N_{th} is the degeneration coefficient, $N_{eff}^{(k)}$ is the ESS at time step k and $Lh_i^{(k)}$ is the weight of i th particle in Eq.(49) and Eq.(50).

In the PF implementation of M-BT example, multi-nominal resampling (MnR) which is used due to the fact that it is considered as the simplest resampling method among others and thus more straightforward for comprehension. An illustration of the first iteration of MnR technique for 10 samples is given in Fig. 2.

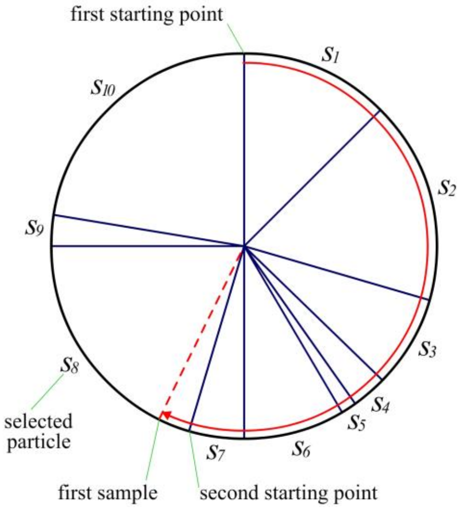


Fig. 2. Graphical illustration of Multi-nominal Resampling.

In MnR method, it is considered that each of the particles in the existing particle set is placed on a wheel so as to cover an angular region in proportion to its weight. Because the sum of all particle weights in the set is equal to 1, one can simply behold that a particle with weight 0.5 covers 180° on the wheel. After the placement process, N independent random samples uniformly distributed in the interval $(0,1)$ are generated. Starting from the beginning point of the first particle's region, each random sample value is compared with the cumulative weight of corresponding particles. The particle which coincides with the value of random sample is selected for the new set. This process is repeated for the number of particles (N). For each loop the beginning point of the selected particle's region is accepted as the starting point. As it can be understood from this simple algorithm, particles with higher weights tend to be selected more than the ones with lower weights. After the process of creating the new set of particles, all the weights of these particles are set to $1/N$. Hence, higher weighted particles form higher percentage within all particles and it becomes more possible to get better results at each time step.

One cycle of the PF algorithm ends with the resampling step. The algorithm is operated passing the next time step.

5. Results and evaluation

Multi-ball tracking problem solution was implemented using particle filter sequential importance resampling algorithm on a two-minute (120 frame) video with $320 \times 240\,px$ frame size containing three freely moving balls. Planar positions of the balls were initialized by the vector $\left[30\ 50\ 144\ 222\ 287\ 92\right]^T$ in which each ordered pair refers to horizontal and vertical positions of green, orange and magenta balls respectively.

The initial planar positions of all particles in each particle set belonging to a single ball were randomly generated around the vector $\left[160\ 120\right]^T$ which was the centroid of any video frame.

The process noise variances for horizontal and vertical directions were selected differently for both positions and velocities as well as for each ball. Such a specification was preferred in order to observe filtering performance explicitly depending on different parameters. The covariance matrix for each ball was formed by the vector $\left[\sigma_{p_x}\ \ \sigma_{p_y}\ \ \sigma_{v_x}\ \ \sigma_{v_y}\right]^T$ as below:

$$\boxed{\begin{array}{l} green \rightarrow \left[80\ \ 60\ \ 40\ \ 30\right]^T \\ orange \rightarrow \left[40\ \ 30\ \ 20\ \ 15\right]^T \\ magenta \rightarrow \left[160\ \ 120\ \ 80\ \ 60\right]^T \end{array}}$$

In M-BT example, measurement noise variances were also initialized with different values for each of the balls. The measurement noise vector was formed as $\left[\rho_g\ \ \rho_o\ \ \rho_m\right]^T = \left[200\ \ 300\ \ 400\right]^T$, in which element was refer to green, orange and magenta balls respectively.

During the implementation process of M-BT example, a number of executions with changing particle counts were performed after specifying initial planar positions, and process and measurement noises. However, only results of executions with particle counts 50, 250 and 1000 are depicted here due to consideration that such a number of visualizations provide enough insight for distinguishing the effect of different number of particles (N). The movements of particle sets for $N = \{50, 250, 1000\}$ at frames $1, 5, 9, 13$ are shown in Fig. 3.

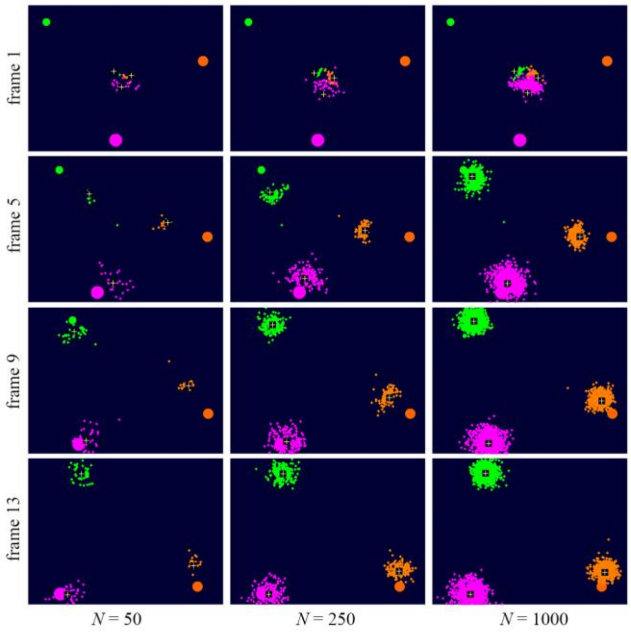


Fig. 3. M-BT problem simulation results.

It can be clearly understood from Fig. 3 that all particle sets get closer to the balls as the number of particles increases. However, because of different process and measurement noise variances, each particle set approximates with different speed rate. The selected initial values and noise parameters are considered as distinctive performance measures for almost all filtering problems. Therefore, it is strongly required that these values and parameters to be properly specified according to the characteristics of the problem attacked.

Trajectories of the balls and associated particle filtering estimates are illustrated in Fig. 4, 5 and 6 consecutively for $N = \{50, 250, 1000\}$ particles. It can be seen when Fig. 4, 5 and 6 examined together that particle filtering estimates remain smoother tracks and get more accurate values corresponding with real trajectories of the balls as the number of particles goes from 50 to 1000. On the other side, the difference between approximation rates is also in evidence.

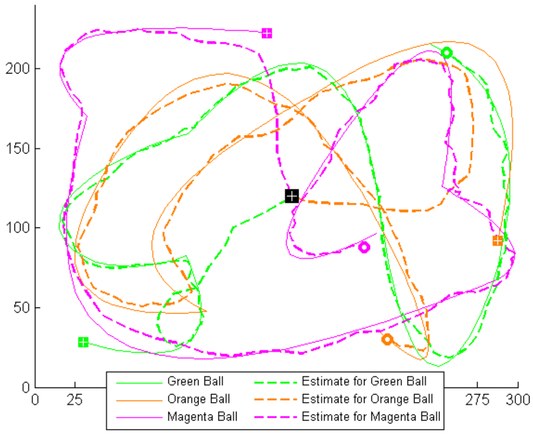


Fig. 4. Ball trajectories and PF estimates for N = 50.

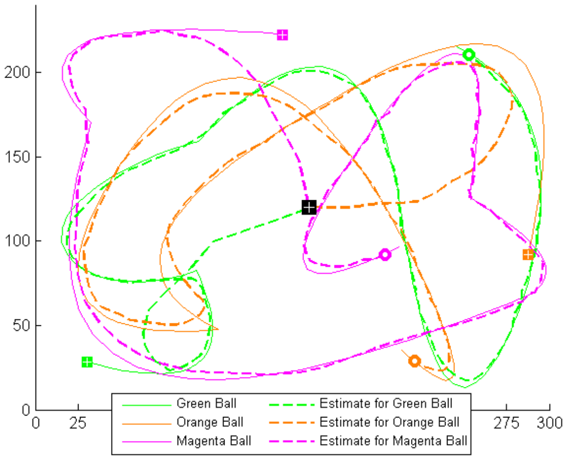


Fig. 5. Ball trajectories and PF estimates for N = 250.

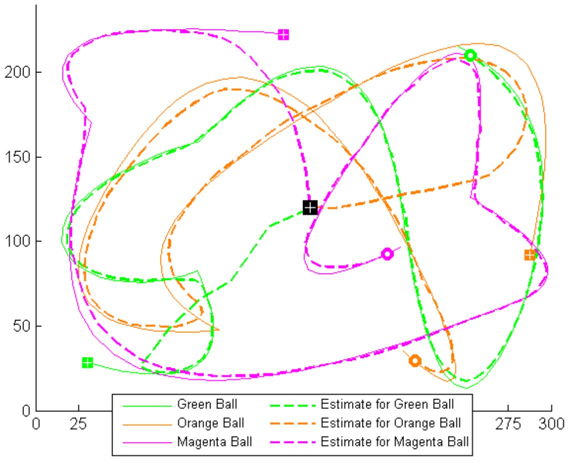


Fig. 6. Ball trajectories and PF estimates for N = 1000.

The root mean square error values are visualized in Fig. 7 for $N = \{50, 250, 1000\}$ particles. A detailed version of RMS errors is given in Table 1 with mean values. The simulation durations are also listed for all execution attempts. It can be extracted from the results in Table I that simulation duration increases as the number of particles increases while the mean RMS error decreases. An acceptable solution for the M-BT problem may be the case *F*. In this case, each of the mean RMS error values is smaller than 1 while the simulation duration remains at a reasonable level. As it can be observed from the results provided with Table 1 mean RMS error values achieved for magenta ball are the least ones for all execution attempts except the case *A*. Consequently, in PF implementation of M-BT problem, selected initial values and noise parameters for the magenta ball can be considered as the best fitting ones.

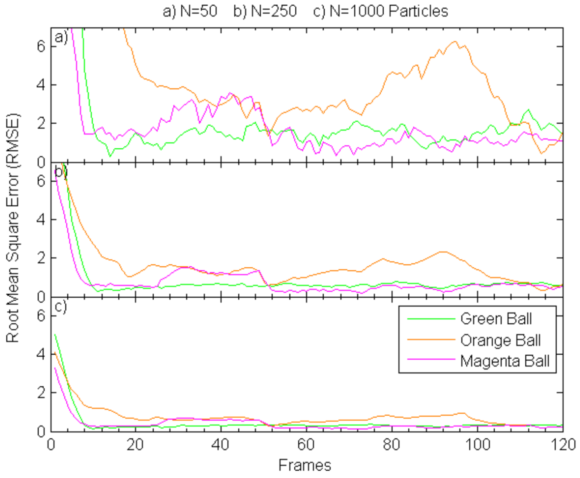


Fig. 7. RMS error values for PF estimates.

Table 1. Mean Rms error values.

Case	Particles	Mean RMS Error (for balls)			Time (sec)
		Green	Orange	Magenta	
A	5	23.21	26.59	23.81	2.767
B	10	10.43	14.71	9.19	2.820
C	25	3.6	6.43	3.13	2.842
D	50	2.31	4.61	1.94	2.900
E	100	1.46	2.87	1.3	3.002
F	250	0.89	1.66	0.81	3.332
G	500	0.61	1.16	0.58	3.811
H	750	0.51	0.86	0.45	4.433
I	1000	0.43	0.75	0.39	4.835
J	5000	0.19	0.32	0.17	13.496
K	*10000*	*0.13*	*0.23*	*0.12*	*24.523*

6. Conclusion

Non-linear state estimation is a usual problem for a great number of research fields including almost all engineering branches, life and even social sciences. Particle filtering has been a popular and frequently referenced method over the recent years for addressing such type of problem. However, it should be noted that solutions provided by PF is mostly not optimal but reasonably approximate in case of operating with proper dynamics and observation models according to the particular problem.

Problem-depended structure of particle filtering method and concentrated presentation of papers by senior researchers lead to deficiency for new researchers in comprehension of the method which has an extensive probabilistic background. In this study, an inclusive and step by step approach is adopted in revealing the underlying theoretical principles of PF in order to accommodate new researchers with the field of non-linear estimation. The solution of an example problem is also provided concerning multi-object tracking with the initial values of parameters, solution steps and comments on results which has a potential of forming an adequate foundation for new researchers to address non-linear state estimation problems of various different fields.

References

[1] A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.
[2] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, IEEE Trans. on Signal Processing, **50**(2), 174 (2002).
[3] A. Doucet, S. Godsill, C. Andrieu, Statistics and Computing, **10**(3), 197 (2000).
[4] A. J. Haug, The MITRE Corporation, USA, Tech. Rep., 05W0000004, Jan. 2005.
[5] B. Ristic, S. Arulampalam, Beyond the Kalman filter: Particle filters for tracking applications, Artech House, Boston, MA, 2004.
[6] K. Demirbas, (2011, Apr., 26), Real-time Recursive State Estimation for Nonlinear Discrete Dynamic Systems with Gaussian or non-Gaussian Noise, Discrete Time Systems, Dr. Mario Alberto Jordán (Ed.), InTech, Available: http://www.intechopen.com/books/discrete-time-systems
[7] G. Terejanu, P. Singla, T. Singh, P. D. Scott, in 11th International Conference on Information Fusion, 1 (2008).
[8] W. Dong, X. Fang, Z. Chen, W. Zhang, Journal of Information & Computational Science, **9**(2), 377 (2012). Available: http://www.joics.com
[9] A. E. Wan, R. Van der Merwe, Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC), IEEE, Lake Louise, Alberta, Canada, Oct. 2000, pp. 153–158.
[10] O. Cappé, S. J. Godsill, E. Moulines, IEEE Proc., **95**(5), 899 (2007).
[11] M. S. Grewal, A. P. Andrews, Control Systems, IEEE, **30**(3), 69 (2010).
[12] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, J. Míguez, Signal Processing Magazine, **20**(5), 19 (2003).
[13] Z. Chen, Adaptive Systems Lab., McMasters University, Hamilton, ON, Tech. Rep., 2003.
[14] S. Särkkä, Bayesian Filtering and Smoothing, Cambridge University Press, 2013.

[15] N. J. Gordon, D. J. Salmond, A. F. M. Smith, IEEE Proceedings F on Radar and Signal Processing, **140**(2), 107 (1993).

[16] P. Sebastian, V. V. Yap, R. Comley, IETE Technical Review, **28**(6), 493 (2011).

[17] M. Richey, The American Mathematical Monthly, **117**(5), 383 (2010).

[18] R. Van der Merwe, N. de Freitas, A. Doucet, E. Wan, Advances in Neural Information Processing Systems, Dec. 2000.

[19] J. L. Anderson, S. L. Anderson, Monthly Weather Review, **127**(12), 2741 (1999).

[20] G. Casella, C. P. Robert, Biometrika, **83**(1), 81 (1996).

[21] A. G. Siapas, AI Lab., Massachusetts Institute of Technology, Tech. Rep., AIM-1402, Dec. 1992.

[22] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice Hall, Englewood Cliffs, NJ, 1993.

[23] A. Saberi, A. Stoorvogel, P. Sannuti, Filtering Theory: With Applications to Fault Detection, Isolation, and Estimation, Birkhäuser Boston, 2007.

[24] J. Aldrich, Statistical Science, **12**(3), 162 (1997).

[25] I. J. Myung, Journal of Mathematical Psychology, **47**(1), 90 (2003).

[26] M. Welling, Caltech Class Notes, Tech. Rep., Jan. 2000. Available: http://www.ics.uci.edu/~welling/classnotes/

[27] G. Welch, G. Bishop, UNCChapel Hill, Tech. Rep., TR 95-041, Jul. 2006.

[28] M. I. Ribeiro, Institute for Systems and Robotics, Lisbon, Tech. Rep., Feb. 2004. Available: http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf.

[29] W. Chen, Ph.D. dissertation, Dept. Chem. Eng., Ohio State Univ., 2004.

[30] R. Frigola, C. E. Rasmussen, In IEEE 52nd Annual Conference on Decision and Control (CDC), Dec. 2013.

[31] M. Isard, A. Blake, Intl. J. of Computer Vision, **29**(1), 1998.

[32] R. Douc, O. Cappé, E. Moulines, In 4th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, Sep. 64 (2005).

[33] X. R. Li, V. P. Jilkov, A Survey of Maneuvering Target Tracking: Dynamic Models. In Proc. 2000 SPIE Conf. on Signal and Data Processing of Small Targets, **4048**, Florida, 212 (2000).

[34] J. D. Hol, T. B. Schon, F. Gustafsson, Nonlinear Statistical Signal Processing Workshop, 2006 IEEE, Sept. 2006, pp.79–82.

[35] B. Azimi-Sadjadi, P. S. Krishnaprasad, Automatica, **41**(6), 945 (2005).

[*] Corresponding author: ttasci@sakarya.edu.tr