Particle Filters

Muhammad Salman Bashir*

Abstract: Both the particle and Kalman filters attempt to approximate the minimum mean-square error (MMSE) estimate of the time-varying parameter. In this scenario, a prior model of the time evolution of the parameter of interest is assumed before the MMSE estimation takes place. The Kalman filter is the (optimal) MMSE estimator for a linear dynamical system with Gaussian noise. For a nonlinear system with nonGaussian noise, the particle filter approximates the mean of posterior distribution[1] at each discrete time step with a finite number of samples or particles. For these general nonlinear systems, the particle filter approaches the MMSE estimator as the number of particles approaches infinity.

1 Beam Tracking in Wireless Communications

One of the defining characteristics of 5G and the future generation of wireless communications is the use of high frequencies (Millimeter wave, Terahertz wave, free-space optics) in the electromagnetic spectrum. The use of microwave frequencies served us well for the 2G, 3G and 4G systems. However, the microwave spectrum was not deemed sufficient to cater to high data-rate demands of the present and future. This drove the push for the deployment of higher frequency bands in the current and future generation of wireless communications.

The high frequency signals will suffer a greater loss in energy per unit propagation distance in the atmospheric channel. Hence, there is a need to focus much of the energy of the high frequency signals in the direction of the receiver so that the receiver can receive enough energy to decode the transmitted symbol. These "directional" signals require the use of pointing and tracking subsystems in the receiver architecture so that the narrow, pencil-thin beams can be tracked efficiently in order to maintain the received signal-to-noise ratio above a certain threshold.

There is a substantial number of studies in literature that have addressed beam tracking for millimeter wave systems [1–14]. Additionally, a number of articles deal exhaustively with

*Correspondence: The author may be reached at muhammad.bashir@fulbrightmail.org or salman.bashir@alumni.purdue.edu for any comments regarding this article.

[1]The mean of posterior distribution is also know as the minimum mean-square error estimate.

beam tracking in free-space optical communications as well [15–26]. As the reader may find out, a significant number of these beam tracking papers deal with Bayesian filtering techniques such as different versions of Kalman and particle filters.

Even though there is no shortage of tutorials on particle filters, we believe that the majority of those tutorials do not provide a clear and easy introduction to the theory of particle filters. In this paper, we have tried to address this challenge by starting from the basics of Monte Carlo integration, and working our way up to the particle filtering algorithm. We mention here that this study is not a detailed exposition on any specialized version of a particle filter, but rather an attempt to furnish an easy-to-follow introduction to the general framework of a particle filtering algorithm.

1.1. Organization of This Study

This article on an introduction to particle filters is organized as follows. In Section 2, we describe the state space model for a nonlinear dynamical system. In Section 3, we present the concepts behind probabilistic or Monte Carlo integration. The Monte Carlo integration is needed to approximate the posterior mean with the help of a finite number of randomly chosen samples at each time step of the filtering process. Section 4 introduces the idea of importance sampling in which we sample from a proposal distribution instead of the target distribution to estimate the posterior mean.

Since the sampling and storage complexity grows with time for an importance sampling filter, we introduce the idea of sequential importance sampling (SIS) in Section 5 that maintains a constant sampling complexity of $\mathcal{O}(N)$ at each stage of time n, where N is the number of particles.

A common version of the SIS filter adopts the prior density as a proposal for particle sampling. For this choice of the proposal density, the weight at each stage of the SIS filter is proportional to the likelihood function. However, in case the support of the prior and the likelihood densities do not overlap significantly, most of the generated particles will possess insignificant weights. However, this problem can be overcome by resampling. In case the system suffers from sample impoverishment or loss of diversity of particles after resampling, the resampled particles can be perturbed about their original positions to generate diversity. All these ideas are discussed in Section 5.

2 State Space Model

For the sake of brevity, lets denote observations from time 0 to time n as $\mathbf{z}_{0:n} := \begin{bmatrix}\mathbf{z}_0 & \cdots & \mathbf{z}_n\end{bmatrix}^T$ and, and the (random) parameter process $\mathbf{X}_{0:n} := \begin{bmatrix}\mathbf{X}_0 & \cdots & \mathbf{X}_n\end{bmatrix}^T$. A realization of this random process is denoted by $\mathbf{x}_{0:n} := \begin{bmatrix}\mathbf{x}_0 & \cdots & \mathbf{x}_n\end{bmatrix}^T$. The evolution of the state sequence in general is

$$\mathbf{x}_n = \mathbf{f}_n(\mathbf{x}_{n-1}, \mathbf{u}_{n-1}), \qquad n \geq 1 \tag{1}$$

where $\mathbf{f}_n : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is a possibly nonlinear function of state parameter $\mathbf{x}_{n-1}$ and $\mathbf{u}_{n-1}$. The quantity $\mathbf{u}_{n-1}$ is i.i.d noise process (possibly not gaussian) statistics. The integers n_x and n_u represent dimensions of signal and noise processes, respectively.

The measurement model of the dynamical system is given by

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n, \mathbf{v}_n), \qquad n \geq 0, \tag{2}$$

where $\mathbf{h}_n : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ is a possibly nonlinear function, $\mathbf{v}_n$ is the measurement noise (possibly not gaussian) sequence and n_v is its dimension.

The *prediction* (or *prior*) stage at time k is obtained via Chapman-Kolmogorov equation as

$$p(\mathbf{x}_n|\mathbf{z}_{0:n-1}) = \int_{-\infty}^{\infty} p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(\mathbf{x}_{n-1}|\mathbf{z}_{0:n-1})\, d\mathbf{x}_{n-1}. \tag{3}$$

The observation $\mathbf{z}_n$ at time n is used to *update* the prior stage via Bayes' rule

$$p(\mathbf{x}_n|\mathbf{z}_{0:n}) = \frac{p(\mathbf{z}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{z}_{0:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{0:n-1})}, \tag{4}$$

where the normalizing constant

$$p(\mathbf{z}_n|\mathbf{z}_{0:n-1}) = \int_{-\infty}^{\infty} p(\mathbf{z}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{z}_{0:n-1})\, d\mathbf{x}_n \tag{5}$$

depends on likelihood function $p(\mathbf{z}_n|\mathbf{x}_n)$. Likelihood function is defined by measurement model given by (2).

From the point-of-view of *filtering*, we are interested in computing posterior means (or expectations)

$$\mathbb{E}[\mathbf{x}_n|\mathbf{z}_{0:n}] = \int_{-\infty}^{\infty} \mathbf{x}_n p(\mathbf{x}_n|\mathbf{z}_{0:n})\, d\mathbf{x}_n \tag{6}$$

at every instant of discrete where $p(\mathbf{x}_n|\mathbf{z}_{0:n})$ is termed the *posterior* density in (6). The posterior mean is also known as the *minimum mean-square error* (MMSE) estimate of the parameter in question. In case, the end goal is *smoothing* of data as opposed to filtering, the posterior is defined as $p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})$.

The integral in (6) will become computationally intractable to compute since the dimension of the integral is growing linearly with time-index n. In this case, one can use sub-optimal

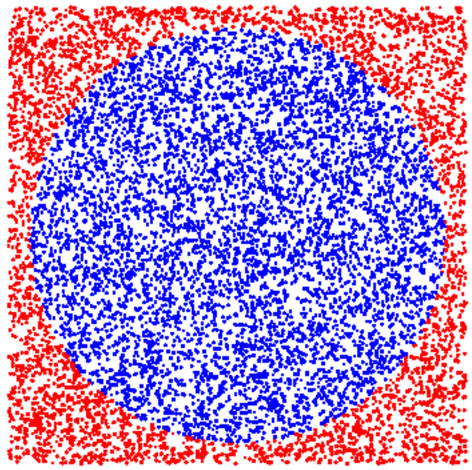


Figure 1. Estimation of area of circle with radius r contained in a square using Monte Carlo technique.

sample based methods—such as Monte Carlo techniques—in order to estimate posterior expectations using independent samples from the posterior density functions. For these systems, a sequential version of the sample based filter leads to a constant computational complexity at any stage n of the filtering process.

In the next section, we consider Monte Carlo integration techniques.

3 Monte Carlo approach

3.1. Monte Carlo methods

As discussed earlier, computation of the posterior expectation can be a daunting task because of the large dimensionality of the parameter space. However, we may solve probabilistic expectations approximately by replacing the integral operation with a finite sum. In this regard, one has to obtain a finite number of independent samples from the posterior distribution and then compute a weighted sum of those samples where the weight of each sample is the value of the posterior density at that sample. This method relies on the weak law of large numbers which states that as the number of samples grow large, the "sample based" Monte Carlo expectation converges to the "true" expectation computed using integrals.

We next illustrate the basic idea of Monte Carlo experiment by a very basic example.

3.2. A simple Monte Carlo technique

We want to estimate the area of circle of a given radius r contained inside a square of area A_s as shown in Fig. 1. If we were to use Monte Carlo approach to compute the area of the circle, we would first obtain N_T independent samples uniformly from the square. Thereafter, we count the number of samples whose Euclidean distance from the center of the square is less than or equal to r (lets call that number N_r). We then have that

$$A_c \triangleq \frac{N_r}{N_T} \times A_s \tag{7}$$

is the area of the circle computed using the Monte Carlo technique. Then, by weak law of large numbers, we have

$$A_c \to \pi r^2 \text{ as } N_T \to \infty. \tag{8}$$

3.3. Monte Carlo Integration Versus Deterministic Integration

Consider $\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)\,dx$. Then, from the Bayesian perspective, assume $p(x)$ to be the posterior density. The Bayesian approach integrates over high dimensional probability distributions to make inferences. Deterministic methods on the other hand involve numerical integration if integral is analytically intractable. The complexity of numerical integration increases exponentially with dimensionality of parameter space. On the other hand Monte Carlo technique positions its samples over regions of high probability density in order to approximate averages. Therefore computation resource is not wasted computing integrals at points that do not matter as much. Hence, the Monte Carlo approach can be more efficient in terms of computational complexity than numerical integration.

3.4. Monte Carlo Integrals: Expectations

As discussed earlier, Riemann integrals can be computed by Monte Carlo methods. This means that we can compute probabilistic expectations also with the help of Monte Carlo methods.

Suppose X is a random variable with density $p(x)$. Then the expectation of any function f of X can be computed as

$$\mathcal{I}(f) \coloneqq \mathbb{E}[f(X)] = \int_{-\infty}^{\infty} f(x)p(x)\,dx. \tag{9}$$

However if we can obtain N i.i.d X_i's from $p(x)$, then by strong law of large numbers,

$$\mathcal{I}_N(f) \coloneqq \frac{1}{N}\sum_{i=1}^{N} f(X_i) = \frac{1}{N}\sum_{i=1}^{N}\int_{-\infty}^{\infty} f(x)\delta(x - X_i)\,dx \xrightarrow{a.s.} \mathcal{I}(f) \text{ as } N \to \infty$$

where $\mathcal{I}(f) = \int_{-\infty}^{\infty} f(x)p(x)\,dx$. In other words, if we define

$$p_N(x) \coloneqq \frac{1}{N}\sum_{i=1}^{N} \delta(x - X_i) \tag{10}$$

then $p_N(x) \xrightarrow{a.e.} p(x)$ as N gets large.

It can be shown that $I_N(f)$ is unbiased estimate of $I(f)$:

$$\mathbb{E}[\mathcal{I}_N(f)] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[f(X_i)] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[f(X)] = \mathbb{E}[f(X)]. \tag{11}$$

If σ_f^2 represents variance of $f(X)$, then

$$\mathrm{Var}[I_N(f)] = \frac{\sigma_f^2}{N}, \tag{12}$$

which goes to zero as $N \to \infty$.

3.5. Monte Carlo Integration Example

Lets suppose we want to compute the integral

$$\mathcal{I} = \int_C g(x)\, dx. \tag{13}$$

Monte Carlo approach is to factorize the integrand as

$$g(x) = f(x)p(x) \text{ where } p(x) \geq 0, \text{ and, } \int_{-\infty}^{\infty} p(x)\, dx = 1.$$

Then

$$\mathcal{I} = \int_C g(x) = \int_C f(x)p(x)\, dx = \mathbb{E}[f(X)] \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i), \tag{14}$$

where X_i are independently drawn from $p(x)$. Thus, we have (approximately) computed the integral $\mathcal{I}$ with the help of a finite sum.

4 Importance Sampling

It is not always possible to get independent samples from a given $p(x)$ to compute $\mathcal{I}$ in (14). One possible solution is to use *Importance Sampling*.

In importance sampling, instead of obtaining samples from $p(x)$, we obtain samples from an *importance sampling distribution*, $q(x)$. In this case, the expectation is calculated as

$$\mathbb{E}_p[f(X)] = \int_{-\infty}^{\infty} f(x)p(x)\, dx = \int_{-\infty}^{\infty} f(x)\frac{p(x)}{q(x)}q(x)\, dx = \mathbb{E}_q\left[f(X)\frac{p(X)}{q(X)}\right], \tag{15}$$

where $\mathbb{E}_f$ denotes the expectation with respect to a certain probability density function $f(x)$. The density function $p(x)$ is known as the *target* distribution. In literature, $q(x)$ is also known as the *proposal* distribution.

The following conditions are necessary for importance sampling to hold:

- $\int_{-\infty}^{\infty} q(x)\,dx$ should be 1.
- $p(x) > 0 \Rightarrow q(x) > 0 \quad \forall x \in \mathbb{R}$.

Then the expectation can be written as

$$\mathbb{E}_p[f(X)] = \mathbb{E}_q\left[f(X)\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i)\frac{p(X_i)}{q(X_i)}. \tag{16}$$

We then have that,

$$\mathbb{E}[f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}] = \int_{-\infty}^{\infty} f(\mathbf{x}_{0:n})\left[\frac{p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})}\right] q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})\,d\mathbf{x}_{0:n}. \tag{17}$$

Let

$$\tilde{W_n} := \frac{p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})} = \frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})}{p(\mathbf{z}_{0:n})q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})}. \tag{18}$$

However we need to know $p(\mathbf{z}_{0:n}) = \int_{-\infty}^{\infty} p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})\,d\mathbf{x}_{0:n}$, which usually is not available. Alternatively, we can set

$$W_n := \frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})}. \tag{19}$$

Then

$$\begin{aligned}\mathbb{E}[f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}] &= \frac{1}{p(\mathbf{z}_{0:n})}\int_{-\infty}^{\infty} f(\mathbf{x}_{0:n})\left[\frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})}\right] q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})\,d\mathbf{x}_{0:n} \\ &= \frac{1}{p(\mathbf{z}_{0:n})}\int_{-\infty}^{\infty} W_n f(\mathbf{x}_{0:n})q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})\,d\mathbf{x}_{0:n} = \frac{\mathbb{E}_q[W_n f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}]}{p(\mathbf{z}_{0:n})}.\end{aligned} \tag{20}$$

Since $W_n q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) = p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})$ by (19), we have that

$$\mathbb{E}[f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}] = \frac{\mathbb{E}_q[W_n f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}]}{p(\mathbf{z}_{0:n})} = \frac{\mathbb{E}_q[W_n f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}]}{\int_{-\infty}^{\infty} W_n q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})\,d\mathbf{x}_{0:n}} = \frac{\mathbb{E}_q[W_n f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}]}{\mathbb{E}_q[W_n|\mathbf{z}_{0:n}]}. \tag{21}$$

Since we can sample from $q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})$, we have that $q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_{0:n} - \mathbf{x}_{0:n}^{(i)})$, where $\mathbf{x}_{0:n}^{(i)}$ are independent samples that are obtained from $q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})$. Substituting this estimate into (20) and using the sifting property of dirac delta functions along with the fact that $p(\mathbf{z}_{0:n}) = \int_{-\infty}^{\infty} W_n q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})\, d\mathbf{x}_n$, we obtain

$$\mathbb{E}[f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}] \approx \frac{1}{\frac{1}{N}\sum_{i=1}^{N} W_n^{(i)}} \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_{0:n}^{(i)}) \underbrace{\frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n}^{(i)})p(\mathbf{x}_{0:n}^{(i)})}{q(\mathbf{x}_{0:n}^{(i)}|\mathbf{z}_{0:n})}}_{W_n^{(i)}} = \frac{1}{\sum_{i=1}^{N} W_n^{(i)}} \sum_{i=1}^{N} f(\mathbf{x}_{0:n}^{(i)}) W_n^{(i)}. \tag{22}$$

Let us now define $\mathcal{W}_n^{(i)}$ as $\mathcal{W}_n^{(i)} := \frac{W_n^{(i)}}{\sum_{i=1}^{N} W_n^{(i)}}$. Then, the final estimate is

$$\mathbb{E}[f(\mathbf{X}_{0:n})|\mathbf{z}_{0:n}] \approx \sum_{i=1}^{N} \mathcal{W}_n^{(i)} f(\mathbf{x}_{0:n}^{(i)}). \tag{23}$$

In this case, the posterior is approximated as

$$p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) \approx \sum_{i=1}^{N} \mathcal{W}_n^{(i)} \delta(\mathbf{x}_{0:n} - \mathbf{x}_{0:n}^{(i)}). \tag{24}$$

We note from (22) that we need to sample N points from the $n+1$ dimensional space in order to compute the posterior expectation at each stage of time n[2]. Therefore, the dimensionality of the Monte Carlo expectation increases linearly with the discrete-time index n. In order to avoid this linear increase in computational complexity, we resort to a more efficient version of importance sampling which is known as *sequential importance sampling* or *recursive importance sampling*.

5 Sequential Importance Sampling

We have to satisfy three assumptions that will enable the sequential importance sampling algorithm to work.

5.1. Assumption on Importance Sampling Distribution

The importance distribution $q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})$ can be modified to enable a "sequential" estimation of posterior $p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})$ in order to save computational resources. The sequential form can be realized if $q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n})$ admits the following form:

$$q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) = q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n}) q(\mathbf{x}_{0:n-1}|\mathbf{z}_{0:n-1}), \tag{25}$$

[2]Here, we have assumed that the dimensionality of $\mathbf{x}_n$ is 1.

which satisfies the probability chain rule decomposition

$$q(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) = \prod_{k=0}^{n} q(\mathbf{x}_{n-k}|\mathbf{x}_{n-k-1}, \mathbf{z}_{0:n-k}), \tag{26}$$

where the last term is $q(\mathbf{x}_0|\mathbf{z}_0)$ and the second last term is $q(\mathbf{x}_1|\mathbf{x}_0, \mathbf{z}_{0:1})$. The weights are then calculated as

$$W_n = \frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})}{q(\mathbf{x}_{0:n-1}|\mathbf{z}_{0:n-1})q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n})}. \tag{27}$$

Multiplying and dividing (27) by $p(\mathbf{z}_{0:n-1}|\mathbf{x}_{0:n-1})p(\mathbf{x}_{0:n-1})$ we have that

$$W_n = \underbrace{\frac{p(\mathbf{z}_{0:n-1}|\mathbf{x}_{0:n-1})p(\mathbf{x}_{0:n-1})}{q(\mathbf{x}_{0:n-1}|\mathbf{z}_{0:n-1})}}_{\text{previous weight}} \times \frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})}{[p(\mathbf{z}_{0:n-1}|\mathbf{x}_{0:n-1})p(\mathbf{x}_{0:n-1})]q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n})}. \tag{28}$$

Thus, we can write (28) as

$$W_n = W_{n-1} \times \frac{p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n})}{[p(\mathbf{z}_{0:n-1}|\mathbf{x}_{0:n-1})p(\mathbf{x}_{0:n-1})]q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n})}. \tag{29}$$

5.2. Assumptions on Measurement Model

We first assume that the measurement model satisfies the i) conditional independence and the ii) *Markov* property. The conditional independence property implies that given that we know x_n and x_m for $n \neq m$, the information about z_m does not tell us anything about z_n and vice versa. The Markov property implies that the observation $\mathbf{z}_n$ only depends on $\mathbf{x}_n$ (please see the measurement model in (2)). This means that $p(\mathbf{z}_n|\mathbf{x}_{0:n}) = p(\mathbf{z}_n|\mathbf{x}_n)$. We thus have that

$$p(\mathbf{z}_{0:n}|\mathbf{x}_{0:n}) = \prod_{k=0}^{n} p(\mathbf{z}_k|\mathbf{x}_{0:n}) = \prod_{k=0}^{n} p(\mathbf{z}_k|\mathbf{x}_k) = p(\mathbf{z}_n|\mathbf{x}_n) \prod_{k=0}^{n-1} p(\mathbf{z}_k|\mathbf{x}_k). \tag{30}$$

5.3. Assumption on Evolution Model

By chain rule of probability density functions, we know that

$$p(\mathbf{x}_{0:n}) = \prod_{k=0}^{n} p(\mathbf{x}_k|\mathbf{x}_{0:k-1}). \tag{31}$$

where $p(\mathbf{x}_0|\mathbf{x}_{0:-1}) \coloneqq p(\mathbf{x}_0)$. We now assume that the evolution process is Markovian which means that $\mathbf{x}_k$ is only dependent on $\mathbf{x}_{k-1}$. This implies that (31) can be rewritten as

$$p(\mathbf{x}_{0:n}) = \prod_{k=0}^{n} p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}) \prod_{k=0}^{n-1} p(\mathbf{x}_k)|\mathbf{x}_{k-1}). \tag{32}$$

Now, we substitute (30) and (32) into (29) and we obtain

$$W_n = W_{n-1} \times \frac{p(\mathbf{z}_n|\mathbf{x}_n) \prod_{k=0}^{n-1} p(\mathbf{z}_k|\mathbf{x}_k)}{\prod_{k=0}^{n-1} p(\mathbf{z}_k|\mathbf{x}_k)} \times \frac{p(\mathbf{x}_n|\mathbf{x}_{n-1}) \prod_{k=0}^{n-1} p(\mathbf{x}_k)|\mathbf{x}_{k-1})}{\prod_{k=0}^{n-1} p(\mathbf{x}_k)|\mathbf{x}_{k-1})} \times \frac{1}{q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n})} \tag{33}$$

$$\implies W_n = W_{n-1} \times \frac{p(\mathbf{z}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})}{q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n})}. \tag{34}$$

Here, we note that we sample N points from the one-dimensional distribution $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{z}_{0:n})$ (assuming that the dimension of $\mathbf{x}_n$ is one) at any stage n of the algorithm. This is in stark contrast to (22) where the N points were sampled from the $n+1$ dimensional distribution at the nth stage. Having said that, we still need to store Nn samples at the nth stage (n samples corresponding to each $\mathbf{x}_{0:n-1}^{(i)}$). Hence, in this case, the sampling complexity is N, but the storage complexity is still Nn.

5.4. Summary of Sequential Importance Sampling algorithm

1. Draw N samples from the proposed importance distribution: $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{z}_{0:n})$.

2. Determine conditional distributions: $p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})$, $p(\mathbf{z}_n|\mathbf{x}_n^{(i)})$ and $q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{z}_{0:n})$

3. Calculate the unnormalized weights $W_n^{(i)}$ as

$$W_n^{(i)} = W_{n-1}^{(i)} \frac{p(\mathbf{z}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{z}_{0:n})}.$$

4. Obtain normalized weights $\mathcal{W}_n^{(i)}$.

5. Estimate posterior distribution as $p(\mathbf{x}_{0:n}|\mathbf{z}_{0:n}) \approx \sum_{i=1}^{N} \mathcal{W}_n^{(i)} \delta(\mathbf{x}_{0:n} - \mathbf{x}_{0:n}^{(i)})$.

5.4.1. Initialization

The initialization of this algorithm is done as follows:

1. $\mathbf{x}_0^{(i)} \sim q(\mathbf{x}_0)$, $\mathbf{x}_1^{(i)} \sim q(\mathbf{x}_1|\mathbf{x}_0^{(i)}, \mathbf{z}_{0:1})$; $i = 1, \ldots, N$.

2. $$W_1^{(i)} = \frac{p(\mathbf{z}_1|\mathbf{x}_1^{(i)})p(\mathbf{x}_1^{(i)}|\mathbf{x}_0^{(i)})}{q(\mathbf{x}_1^{(i)}|\mathbf{x}_0^{(i)}, \mathbf{z}_{0:1})}.$$

3.
$$\mathcal{W}_1^{(i)} = \frac{W_1^{(i)}}{\sum_{i=1}^{N} W_1^{(i)}}.$$

5.5. Summary of Sequential Importance Sampling (SIS) Filter algorithm

If only an estimate of $\mathbf{x}_n$ (filtering) is required instead of $\mathbf{x}_{0:n}$ (smoothing) at time n, the sequential importance sampling algorithm can be simplified further via another assumption on the importance sampling distribution $q(\cdot)$. If we assume that $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{0:n}) = q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n)$, then we only need to store $\mathbf{x}_{n-1}^{(i)}$ and generate the particle at next time step from $q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n)$. In this case, the storage complexity is N and the sampling complexity is also N. For this particular case, the conditional mean of $p(\mathbf{x}_n|\mathbf{z}_{0:n})$ can be obtained as follows:

1. Draw N samples from the proposed importance distribution: $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n)$.

2. Determine conditional distributions: $p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})$, $p(\mathbf{z}_n|\mathbf{x}_n^{(i)})$ and $q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n)$.

3. Calculate the unnormalized weights $W_n^{(i)}$ as
$$W_n^{(i)} = W_{n-1}^{(i)} \frac{p(\mathbf{z}_n|\mathbf{x}_n^{(i)}) p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n)}.$$

4. Obtain normalized weights $\mathcal{W}_n^{(i)}$.

5. Estimate posterior distribution as $p(\mathbf{x}_n|\mathbf{z}_{0:n}) \approx \sum_{i=1}^{N} \mathcal{W}_n^{(i)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(i)})$.

5.6. Degeneracy Problem

In practice, iteration of steps 1 and 3 in the SIS filter algorithm (defined in Section 5.5) may lead to problems of degeneracy where only a few particles will have significant weights. *Degeneracy* is typically measured by effective sample size N_e defined as

$$N_e := \frac{1}{\sum_{i=1}^{N} (\mathcal{W}_n^{(i)})^2}. \tag{35}$$

A large degeneracy implies a small value of N_e. A small N_e implies that we are spending too much computational effort to update particles that contribute very small to the posterior approximation.

An example of degeneracy If we set $q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$. Then

$$W_n^{(i)} = W_{n-1}^{(i)} \cdot p(\mathbf{z}_n|\mathbf{x}_n^{(i)}).$$

In this case, if the prior $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$ is a much broader distribution than the likelihood $p(\mathbf{z}_n|\mathbf{x}_n^{(i)})$, or if both the densities are not "aligned," then only a few particles will have significant weights, and a large number of particles will have weights close to zero. Thus, in this situation, we are effectively using a very small number of particles to represent the posterior distribution.

5.7. Resampling

If N_e goes below some preset threshold, we then need to resample. The motivation behind resampling is to retain particles with large weights and discard those with smaller weights. Resampling process involves re-selecting a new set of N particles from the current set of N particles, where a new particle is selected with probability equal to its normalized weight. This process is carried out "with replacement," which means that a particle that has been selected before can be reselected in future. Thus, particles with higher normalized weights are likely to be selected more than once, whereas particles with lower weights might not be reselected at all after resampling is complete. After resampling, each particle is assigned a uniform weight of $\frac{1}{N}$.

We next look at a common algorithm used for resampling the particles.

5.7.1. Systematic Resampling Algorithm

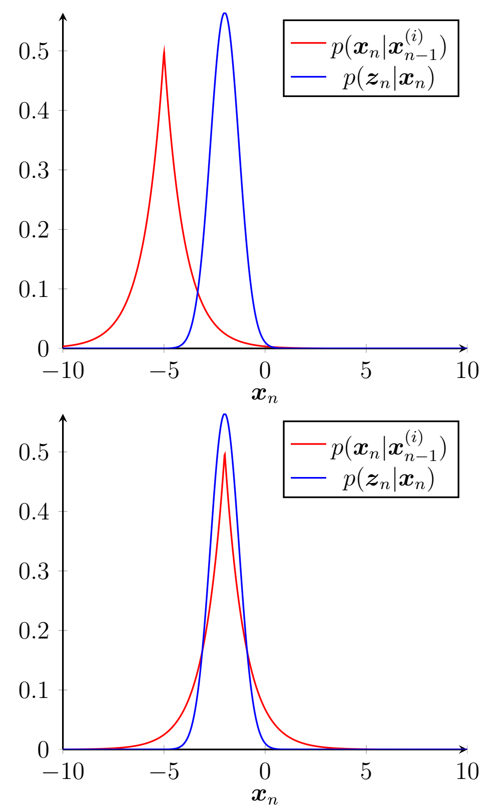
1. Given a set of particles and weights $\{\mathbf{x}_n^{(i)}, \mathcal{W}_n^{(i)}, i = 1, \ldots, N.\}$
2. Select a new sample from the set $\left\{\mathbf{x}_n^{(i)}\right\}_{i=1}^N$ with probability $\mathcal{W}_n^{(i)}$ and call it $\hat{\mathbf{x}}_n^{(i)}$. The new sample is assigned the weight $\hat{\mathcal{W}}_n^{(i)} = \frac{1}{N}$.
3. Repeat the resampling process until N number of new particles have been sampled.

5.7.2. A Common Version of Sequential Importance Resampling (SIR) Filter

One common version of SIR filter chooses the prior distribution as a proposal: $q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$. Then

$$W_n^{(i)} = p(\mathbf{z}_n|\mathbf{x}_n^{(i)}), \tag{36}$$

and these updates are followed by resampling at each step. Thus the likelihood $p(\mathbf{z}_n|\mathbf{x}_n^{(i)})$ solely decides the weight of the particle i at the nth stage. It is worth noting that there is no $W_{n-1}^{(i)}$



Two different scenarios for weight propagation in particle filters. In the top figure most particles will have negligible weight: in the other scenario most particles will have significant weight as the two densities are "aligned."

Figure 2. For particles to have any meaningful weights, there should be a significant overlap between the prior and likelihood density functions.

term in weight update equation (36) since after resampling at time $n-1$, the weights are set equal to $\frac{1}{N}$.

There is a potential problem with this version of SIR filter. This has to do with the independence of the proposal or prior distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$ from observations $\mathbf{z}_n$. Since the particles are sampled from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$ and then weighted by the likelihood $p(\mathbf{z}_n|\mathbf{x}_n^{(i)})$, only those particles $\mathbf{x}_n^{(i)}$ will have considerable weight that are sampled from the region of significant overlap of the two densities (please see Fig. 2 in this regard). Thus, it is more beneficial to update states by taking observations or the likelihood function into account at each stage of time n.

5.7.3. Problems with Resampling: Sample Impoverishment or loss of diversity

Particles with large weights are likely to be redrawn multiple times. In extreme case, all particles might "collapse" into a single particle. In this scenario, we are approximating the posterior with just one distinct sample which poorly characterizes the true posterior density. This phenomenon is known as *sample impoverishment* or *diversity loss* of samples or particles. The solution to this problem is to use Bootstrap or SIR filter with Markov Chain Monte Carlo (MCMC) steps, which is the subject of discussion in the next topic.

5.8. Bootstrap Particle Filter with Markov Chain Monte Carlo (MCMC) steps

We can use the SIR filter in conjunction with MCMC steps to alleviate sample impoverishment issues. After the particles have been resampled, they are *perturbed* about their locations using MCMC algorithm in order to distribute them to regions of higher posterior probabilities. This is one of the most common techniques to introduce "diversity" in resampled particles [27].

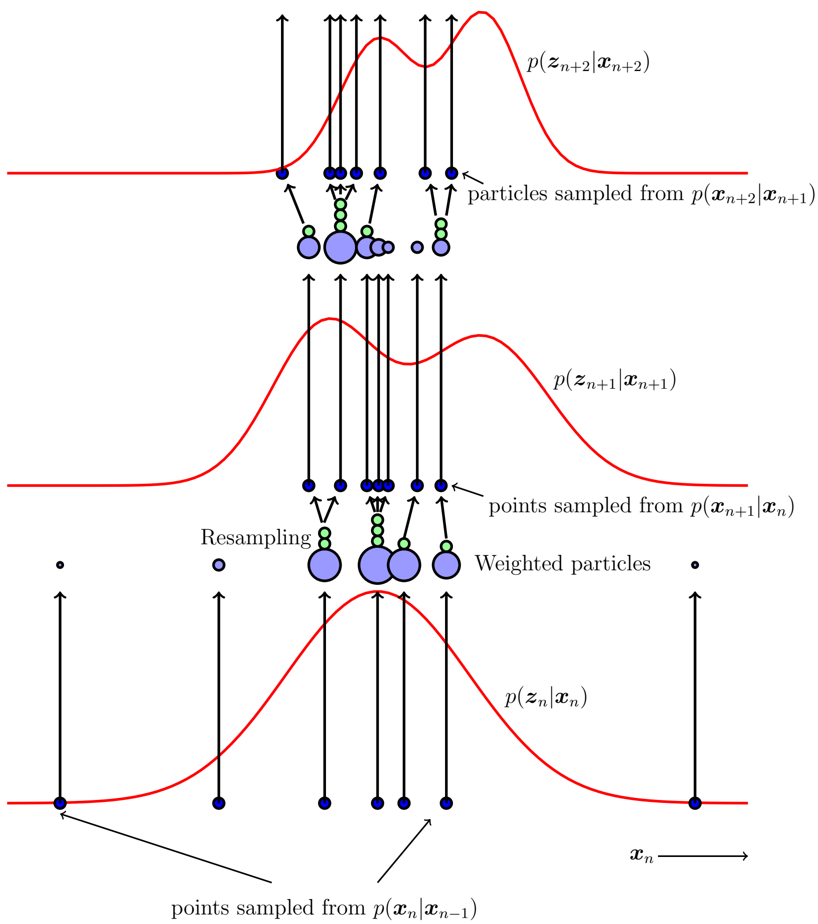
After obtaining resampled particles $\hat{\mathbf{x}}_n^{(i)}$, we obtain a new set of particles $\tilde{\mathbf{x}}_n^{(i)}$ as follows

$$\tilde{\mathbf{x}}_n^{(i)} = \hat{\mathbf{x}}_n^{(i)} + \boldsymbol{\epsilon}^{(i)}, \tag{37}$$

where $\boldsymbol{\epsilon}^{(i)} \sim p(\boldsymbol{\epsilon}) = \mathcal{N}(0, C_{\epsilon\epsilon})$, where $C_{\epsilon\epsilon}$ is a diagonal matrix. The corresponding acceptance probability for the Metropolis-Hastings approach of MCMC is given by [27]

$$A(\tilde{\mathbf{x}}_n^{(i)}, \hat{\mathbf{x}}_n^{(i)}) = \min\left\{\frac{p(\mathbf{z}_n|\tilde{\mathbf{x}}_n^{(i)})p(\tilde{\mathbf{x}}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{p(\mathbf{z}_n|\hat{\mathbf{x}}_n^{(i)})p(\hat{\mathbf{x}}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}, 1\right\}. \tag{38}$$

This acceptance probability $A(\tilde{\mathbf{x}}_n^{(i)}, \hat{\mathbf{x}}_n^{(i)})$ is then compared to a value that is sampled from a uniform distribution between 0 and 1. If the acceptance probability is greater than the sampled value, we replace $\hat{\mathbf{x}}_n^{(i)}$ with $\tilde{\mathbf{x}}_n^{(i)}$. Otherwise, we retain $\hat{\mathbf{x}}_n^{(i)}$ as our final particle.



Weight propagation of particles at different instants of time.

Figure 3. Weight propagation in SIR filter

Since the posterior is proportional to $p(\mathbf{z}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})$, we note that the acceptance probability compares the fraction of two posterior densities (one corresponding to perturbed particle $\tilde{\mathbf{x}}_n^{(i)}$, and the other corresponding to resampled particle $\hat{\mathbf{x}}_n^{(i)}$) with 1. This means that the larger the value of posterior corresponding to the perturbed particle, the greater is the probability of it replacing the resampled particle in the final set. In this sense, we only choose a given perturbed particle to represent the posterior if it maximizes the posterior density.

We can summarize the basic algorithm in the next section.

5.9. Bootstrap Particle Filter with MCMC steps algorithm

1 **Initialize:** Draw $\mathbf{x}_0^{(i)}$ from $p(\mathbf{x}_0)$ for $i = 1, 2, \ldots, N$. Set $W_0^{(i)} = \frac{1}{N}$.

2 **Importance Sampling:** Draw $\mathbf{x}_n^{(i)}$ from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})$.

3 **Weight update:** $W_n^{(i)} = p(\mathbf{z}_n|\mathbf{x}_n^{(i)})$.

4 **Weight normalization:** $\mathcal{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{i=1}^N W_n^{(i)}}$.

5 **Resampling decision:** Compute $N_{\text{eff}} = \frac{1}{\sum_{i=1}^N \mathcal{W}_n^{(i)2}}$.

If $N_{\text{eff}} \leq N_{\text{threshold}}$, then resample. Otherwise compute the approximate distribution as follows,

$$p(\mathbf{x}_n|\mathbf{z}_{1:n}) \approx \sum_{i=1}^N \mathcal{W}_n^{(i)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}),$$

and go to step 2.

6 **Resampling:** Resample $\hat{\mathbf{x}}_n^{(i)}$ from $\mathbf{x}_n^{(i)}$ using *Systematic Resampling* algorithm.

7 **Diversify:**

- Draw $\boldsymbol{\epsilon}^{(i)}$ from $\mathcal{N}(0, C_{\epsilon\epsilon})$.
- Compute $\tilde{\mathbf{x}}_n^{(i)} = \hat{\mathbf{x}}_n^{(i)} + \boldsymbol{\epsilon}^{(i)}$.
- Compute *Diversification Acceptance Probability* as follows:

$$A(\tilde{\mathbf{x}}_n^{(i)}, \hat{\mathbf{x}}_n^{(i)}) = \min\left\{\frac{p(\mathbf{z}_n|\tilde{\mathbf{x}}_n^{(i)})p(\tilde{\mathbf{x}}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{p(\mathbf{z}_n|\hat{\mathbf{x}}_n^{(i)})p(\hat{\mathbf{x}}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}, 1\right\}$$

- Draw u_k from $\mathcal{U}(0, 1)$.

– Set $\mathbf{x}_n^{(i)} = \tilde{\mathbf{x}}_n^{(i)}$ if $u_k < A(\tilde{\mathbf{x}}_n^{(i)}, \hat{\mathbf{x}}_n^{(i)})$. Else set $\mathbf{x}_n^{(i)} = \hat{\mathbf{x}}_n^{(i)}$.

8 **Distribution:** Compute distribution as follows:

$$p(\mathbf{x}_n|\mathbf{z}_{0:n}) \approx \sum_{i=1}^{N} \mathcal{W}_n^{(i)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}).$$

References

[1] L. Afeef, M. Karabacak, and H. Arslan, "Joint tracking of multiple beams in beamspace mimo systems," 2020.

[2] R. Deng, S. Chen, S. Zhou, Z. Niu, and W. Zhang, "Channel fingerprint based beam tracking for millimeter wave communications," *IEEE Communications Letters*, vol. 24, no. 3, pp. 639–643, 2020.

[3] J. Palacios, D. De Donno, and J. Widmer, "Tracking mm-wave channel dynamics: Fast beam training strategies under mobility," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[4] J. Li, Y. Sun, L. Xiao, S. Zhou, and C. E. Koksal, "Fast analog beam tracking in phased antenna arrays: Theory and performance," 2019.

[5] A. Shahmansoori, B. Uguen, G. Destino, G. Seco-Granados, and H. Wymeersch, "Tracking position and orientation through millimeter wave lens mimo in 5g systems," *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1222–1226, 2019.

[6] S. Shaham, M. Ding, M. Kokshoorn, Z. Lin, S. Dang, and R. Abbas, "Fast channel estimation and beam tracking for millimeter wave vehicular communications," *IEEE Access*, vol. 7, pp. 141 104–141 118, 2019.

[7] Y. J. Kim and Y. S. Cho, "Beam-tracking technique for millimeter-wave cellular systems using subarray structures," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7806–7810, 2018.

[8] J. Lim, H.-M. Park, and D. Hong, "Beam tracking under highly nonlinear mobile millimeter-wave channel," *IEEE Communications Letters*, vol. 23, no. 3, pp. 450–453, 2019.

[9] S. Shahsavari, M. A. Amir Khojastepour, and E. Erkip, "Robust beam tracking and data communication in millimeter wave mobile networks," in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2019, pp. 1–8.

[10] P. Li, D. Liu, X. Hou, and J. Wang, "Trajectory prediction and channel monitoring aided fast beam tracking scheme at unlicensed mmwave bands," *Electronics*, vol. 9, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/5/747

[11] V. Va, H. Vikalo, and R. W. Heath, "Beam tracking for mobile millimeter wave communication systems," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016, pp. 743–747.

[12] J. Bae, S. H. Lim, J. H. Yoo, and J. W. Choi, "New beam tracking technique for millimeter wave-band communications," 2017.

[13] C. Han, S. Zhou, and L. Xiao, "A new multipath beam tracking algorithm for mmwave mobile communications," in *Proceedings of the 2019 International Symposium on Signal Processing Systems*, ser. SSPS 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 61–66. [Online]. Available: https://doi.org/10.1145/3364908.3364919

[14] J. Li, Y. Sun, L. Xiao, S. Zhou, and A. Sabharwal, "How to mobilize mmwave: A joint beam and channel tracking approach," 2018.

[15] M. S. Bashir and M. R. Bell, "Optical beam position estimation in free-space optical communication," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, December 2016.

[16] ——, "Optical beam position tracking in free-space optical communication systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 20, no. 2, April 2018.

[17] M. S. Bashir and M. R. Bell, "The impact of optical beam position estimation on the probability of error in free-space optical communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 3, pp. 1319–1333, June 2019.

[18] M. S. Bashir and S. S. Muhammad, "Time synchronization in photon-limited deep space optical communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 30–40, 2020.

[19] M. S. Bashir, "Free-space optical communications with detector arrays: A mathematical analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1420–1429, 2020.

[20] M. S. Bashir and M.-S. Alouini, "Signal acquisition with photon-counting detector arrays in free-space optical communications," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2181–2195, 2020.

[21] M. S. Bashir and M.-S. Alouini, "Beam tracking with photon-counting detector arrays in free-space optical communications," 2020, (available on arXiv at https://arxiv.org/pdf/2001.04007.pdf).

[22] M. S. Bashir and M. S. Alouini, "Free-space optical MISO communications with an array of detectors," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1765–1780, 2020.

[23] M.-C. and Tsai, M. S. Bashir, and M.-S. Alouini, "A single detector versus an array of detectors receiver in free-space optical communications: A performance comparison," Jan 2021. [Online]. Available: engrxiv.org/fds4a

[24] M. S. Bashir and M.-S. Alouini, "Optimal power allocation between beam tracking and symbol detection channels in a free-space optical communications receiver," Dec 2020. [Online]. Available: https://www.techrxiv.org/articles/preprint/Optimal_Power_Allocation_Between_Beam_Tracking_and_Symbol_Detection_Channels_in_a_Free-Space_Optical_Communications_Receiver/13296563/1

[25] M. S. Bashir, M.-C. Tsai, and M.-S. Alouini, "Cramér–rao bounds for beam tracking with photon counting detector arrays in free-space optical communications," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1065–1081, 2021.

[26] M. S. Bashir and M.-S. Alouini, "Adaptive acquisition schemes for photon-limited free-space optical communications," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 416–428, 2021.

[27] *Particle-Based Bayesian State–Space Processors*. John Wiley Sons, Ltd, 2009, ch. 7, pp. 237–298. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470430583.ch7