PARTICLE FILTERING APPROACH TO ADAPTIVE TIME-DELAY ESTIMATION

Eric A. Lehmann

Western Australian Telecommunications Research Institute
35 Stirling Highway, Perth WA 6009, Australia
E-mail: Eric.Lehmann@watri.org.au

ABSTRACT

A particle filter algorithm is developed for the problem of online subsample time-delay estimation between noisy signals received at two spatially separated sensors. The delay is modeled as an adaptive FIR filter whose coefficients are determined by the tracker's particles, and updated on a sample-by-sample basis. Efficient tracking of the delay parameter over time is ensured with the derivation of a global system model integrating the target dynamics for both near-field and far-field operation. Experimental simulations are carried out to assess the algorithm's convergence and tracking performance, and demonstrate that the proposed method is able to efficiently track time delays with stationary signals as well as speech.

1. INTRODUCTION

The problem of estimating the time delay between signals received at two spatially separated sensors has been given substantial attention over the last few decades [1–4]. Time delay estimation represents a crucial process in several domains such as surveillance and defence systems, wireless communication networks, geophysics and biomedical engineering [4]. In order to provide robust time delay estimates (TDEs) for a variety of applications, several approaches have been proposed in the literature. Some of these methods, such as the well-known generalized cross-correlation (GCC) approach [5], derive time delay estimates on the basis of the current frame of signal data only, without accounting for previously computed TDEs. A second class of time delay estimators includes adaptive algorithms that perform some sort of tracking of the parameter of interest, such as the LMS (least mean square) method [2, 6]. The main advantage of using an adaptive technique is the ability to deal with time-varying time delays resulting from relative target–sensor motions.

By taking into account the temporal information available from previous estimation updates, Bayesian filtering provides an attractive framework for the generic problem of parameter estimation and tracking [7]. Among the different methods using this state-space approach, the concept of particle filtering (PF) appears as a promising technique able to deal with nonlinear and/or non-Gaussian problems, and has received considerable attention in the last few years [8–10].

The work presented in this paper describes the implementation of a PF-based algorithm for the problem of estimating and tracking time delays. Although the developed method can be applied to a wide range of TDE applications, the derivations will be based on the example of determining the time delay of arrival of a source signal in the context of speaker localization. The next section provides a brief overview of the various concepts involved in this specific type of application. Section 3 briefly reviews the Bayesian approach and particle filtering principles, and the proposed PF framework for time delay estimation is then developed in Section 4. The last part of this paper presents some experimental results obtained from the simulation of the proposed algorithm, and the paper finally concludes with a discussion of these results and a summary of future developments.

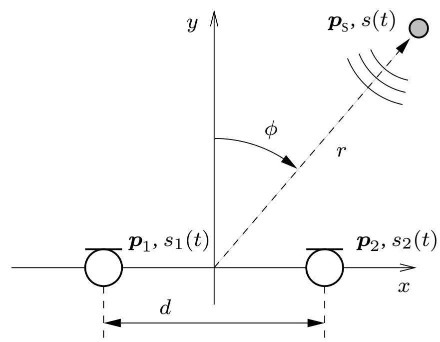

Fig. 1. General setup definition for the TDE problem.

2. TIME DELAY ESTIMATION

The time delay estimation problem can be formulated as follows, see Figure 1. A source (target) is located at the position $\boldsymbol{p}_s = [x_s \;\; y_s]^T$ and emits the unknown signal $s(t)$. Two sensors, located at $\boldsymbol{p}_i = [x_i \;\; y_i]^T, i = 1, 2$, provide the signals $s_1(t)$ and $s_2(t)$, which are commonly modeled as:

$$s_1(t) = s(t) + \eta_1(t)\,, \tag{1a}$$

$$s_2(t) = \alpha \cdot s(t - \tau) + \eta_2(t)\,, \tag{1b}$$

where $\tau \in \mathbb{R}$ is the time difference of arrival (TDOA) of the source signal with respect to each sensor, and $\alpha > 0$ represents the difference in signal amplitude observed at the sensors. The diffuse noise variables $\eta_1(\cdot)$ and $\eta_2(\cdot)$ are assumed to be independent zero-mean Gaussian processes, and uncorrelated with the source signal. The aim is to provide an estimate $\hat{\tau}$ of the (potentially time-varying) time delay on the basis of the noisy measurements $s_1(\cdot)$ and $s_2(\cdot)$.

3. STATE-SPACE APPROACH

3.1. Bayesian Filtering

Consider the following discrete-time estimation problem, where $k = 1, 2, \ldots,$ represents the time index. Let the variable X_k represent the state of the system under consideration at time k. At each time step, assume that a (noisy) observation of the state becomes available, and denote this measurement variable as Y_k. Using a Bayesian filtering approach and assuming Markovian dynamics, this system can be globally represented by means of the following two state-space equations [7]:

$$X_k = g(X_{k-1}, u_k)\,, \tag{2a}$$
$$Y_k = h(X_k, v_k)\,, \tag{2b}$$

where $g(\cdot)$ and $h(\cdot)$ are possibly nonlinear functions, and u_k and v_k are possibly non-Gaussian noise variables. Ultimately, one would like to compute the so-called posterior probability density function (PDF) $p(X_k|Y_{1:k})$, where $Y_{1:k} = \{Y_1, \ldots, Y_k\}$ represents the concatenation of all measurements up to time k. The density $p(X_k|Y_{1:k})$ contains all the statistical information available regarding the current condition of the state variable X_k, and an estimate $\widehat{X}_k$ of the state then follows, for instance, as the mean or the mode of this PDF.

The solution to this Bayesian filtering problem consists in the following two steps of prediction and update [9]. Assuming that the posterior density $p(X_{k-1}|Y_{1:k-1})$ is known at time $k-1$, the posterior PDF $p(X_k|Y_{1:k})$ for the current time step k can be computed using the following recursion equations:

$$p(X_k|Y_{1:k-1}) = \int p(X_k|X_{k-1})\, p(X_{k-1}|Y_{1:k-1})\, \mathrm{d}X_{k-1}\,,$$
$$p(X_k|Y_{1:k}) \propto p(Y_k|X_k)\, p(X_k|Y_{1:k-1})\,,$$

where $p(X_k|X_{k-1})$ is the transition density, and $p(Y_k|X_k)$ is the so-called likelihood function.

3.2. Sequential Monte Carlo Method

Particle filtering is an approximation technique that solves the Bayesian filtering problem by representing the posterior density as a set of N samples of the state space $X_k^{(n)}$ (particles) with associated weights $w_k^{(n)}, n \in \{1, \ldots, N\}$ (see e.g. [8]). Originally proposed in [9], the so-called bootstrap algorithm is an attractive PF variant due to its simplicity of implementation and low computational demands. Assuming that the set of particles and weights $\{(X_{k-1}^{(n)}, w_{k-1}^{(n)})\}_{n=1}^N$ is a discrete representation of the posterior density at time $k-1$, the bootstrap PF update can be described as follows, for $k = 1, 2, \ldots$:

1. *Prediction*: propagate the particles through the transition equation, $\widetilde{X}_k^{(n)} = g(X_{k-1}^{(n)}, u_k)$.
2. *Update*: assign each particle a likelihood weight as given by $\tilde{w}_k^{(n)} = w_{k-1}^{(n)} \cdot p(Y_k|\widetilde{X}_k^{(n)})$, then normalize the weights:
$$w_k^{(n)} = \tilde{w}_k^{(n)} \cdot \Big(\sum\nolimits_{i=1}^N \tilde{w}_k^{(i)}\Big)^{-1}. \tag{3}$$
3. *Resampling*: compute the effective sample size,
$$N_{\text{eff}} = \Big(\sum\nolimits_{n=1}^N \big(w_k^{(n)}\big)^2\Big)^{-1}.$$
If N_{eff} is above some pre-defined threshold N_{t}, simply define $X_k^{(n)} = \widetilde{X}_k^{(n)}, \forall n$. Otherwise, draw N new samples $X_k^{(n)}, n \in \{1, \ldots, N\}$, from the existing set of particles $\{\widetilde{X}_k^{(i)}\}_{i=1}^N$ according to their weights $w_k^{(i)}$, then reset the weights to uniform values: $w_k^{(n)} = N^{-1}, \forall n$.

As a result, the set of particles and weights $\{(X_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ is approximately distributed as the current posterior density $p(X_k|Y_{1:k})$. The sample set approximation of the true posterior PDF can then be obtained using:

$$p(X_k|Y_{1:k}) \approx \sum\nolimits_{n=1}^N w_k^{(n)}\, \delta(X_k - X_k^{(n)})\,,$$

where $\delta(\cdot)$ is the Dirac delta function, and an estimate $\widehat{X}_k$ of the target state for the current time step k follows as:

$$\widehat{X}_k = \int X_k \cdot p(X_k|Y_{1:k})\, \mathrm{d}X_k \approx \sum\nolimits_{n=1}^N w_k^{(n)} X_k^{(n)}. \tag{4}$$

It can be shown that the variance of the weights $w_k^{(n)}$ can only increase over time, which decreases the overall accuracy of the algorithm. This constitutes the so-called degeneracy problem, known to affect any PF implementation. The conditional resampling step in the algorithm given above is introduced as way to mitigate these effects. This resampling process can be easily implemented using a scheme based on a cumulative weight function, see e.g. [8].

4. PARTICLE FILTER FOR TDOA TRACKING

4.1. Algorithm Development

For time delay estimation, the state parameter X_k is defined as the variable of interest in (1), i.e. $X_k \triangleq \tau_k$, where $\tau_k \in \mathbb{R}$ is expressed as a multiple of the sampling period. The PF approach to TDE presented here relies on a representation of the TDOA τ as a finite impulse response (FIR) filter with $2P+1$ coefficients h_i, which is used to process one of the microphone signals [6]. Assuming that the signal data is collected in a series of overlapping frames of length $2P+1$, the following vector notation can be introduced for the signals and filter coefficients:

$$\mathbf{h}(k) = \big[h_{-P}(k)\ \cdots\ h_0(k)\ \cdots\ h_P(k)\big]^{\mathrm{T}}, \tag{5}$$
$$\mathbf{s}_i(k) = \big[s_i(k-2P)\ \cdots\ s_i(k-P)\ \cdots\ s_i(k)\big]^{\mathrm{T}},$$

for $i = 1, 2$. Ideally, the FIR coefficients are samples of a $\operatorname{sinc}(\cdot)$ function that represents a (truncated) fractional delay filter [11], implicitly compensating for the TDOA in the sensor signals. Thus, the filter weights in (5) are here constrained as follows:

$$h_i(k) = \operatorname{sinc}(i - \tau_k),\ i \in \{-P, \ldots, P\}\,. \tag{6}$$

The PF algorithm presented in Section 3.2 requires the definition of two important concepts, the likelihood function $p(Y_k|X_k)$ and the transition density $p(X_k|X_{k-1})$, which are derived in the following.

1) *Likelihood Function*. Assuming $\alpha \approx 1$, the combined signal observation Y_k for time delay estimation is formed as the error $e(k) = s_2(k) - s_1(k - \tau_k)$, which can be expressed in vector form as follows: $Y_k \triangleq s_2(k-P) - \mathbf{h}(k)^{\mathrm{T}}\mathbf{s}_1(k)$.[1] Minimization of this error function can be ensured by assuming that $e(k)$ has a distribution concentrated around zero with a small variance σ_Y^2. Using a Gaussian error PDF leads to the following likelihood function:

$$p(Y_k|X_k) = \mathcal{N}\big(s_2(k-P); \mathbf{h}(k)^{\mathrm{T}}\mathbf{s}_1(k), \sigma_Y^2\big), \tag{7}$$

where $\mathcal{N}(\cdot\,; \mu, \sigma^2)$ is the density of a Gaussian variable with mean μ and variance σ^2. Note that despite the Gaussian error assumption in (7), the likelihood function $p(Y_k|X_k)$ itself cannot be approximated with a normal distribution in practice, due to the effects of the received signals $s_1(\cdot)$ and $s_2(\cdot)$. The top graph in Figure 2 depicts a typical example of likelihood function computed for a source emitting speech, demonstrating the non-Gaussian and multi-modal character of this function.

[1] Note that the output of the particle filter has to be delayed by P samples in order to maintain causality. The number of FIR coefficients should hence remain as small as possible in a practical implementation.

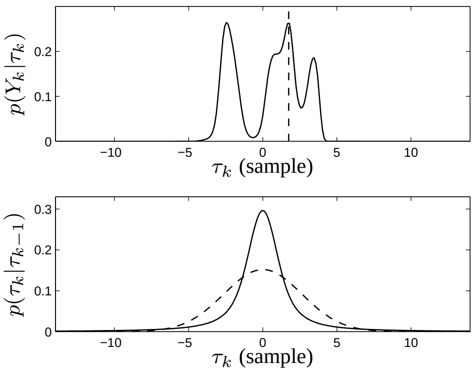

Fig. 2. Examples of PF-related PDFs. *Top plot:* likelihood function (dashed line: source's TDOA). *Bottom plot:* transition density (dashed line: Gaussian PDF with identical mean and variance).

2) *Transition Density*. The purpose of the transition equation (2a) is to provide a model of the specific dynamics of the state variable X_k over time. The development of the transition PDF for parameter τ_k will make use of a random walk process to model the dynamics of the target in Cartesian coordinates:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \mathbf{I} \cdot \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \mathbf{u}_k \,, \qquad (8)$$

where $\mathbf{u}_k \sim \mathcal{N}([0 \;\; 0]^\mathrm{T}, \sigma_X^2 \mathbf{I})$, $\mathbf{I}$ is the 2×2 identity matrix, and $\sigma_X = \bar{v} f_\mathrm{s}^{-1}$, with f_s the sampling frequency and $\bar{v}$ the maximum steady-state velocity of the considered target.

Based on the problem setup and definitions given in Figure 1, the TDOA can be computed as (plane wave assumption):

$$\tau_k = \frac{f_\mathrm{s} d}{c} \cdot \frac{x_k}{\sqrt{x_k^2 + y_k^2}} \,, \qquad (9)$$

with c the propagation speed of acoustic waves. A first-order Taylor expansion of (9) about the point $[x_{k-1} \;\; y_{k-1}]^\mathrm{T}$ can be shown to yield the following approximation formula:

$$\tau_k \approx \tau_{k-1} + \frac{f_\mathrm{s} d \, y_{k-1}^2}{c \, r_{k-1}^3} (x_k - x_{k-1}) - \frac{f_\mathrm{s} d \, x_{k-1} y_{k-1}}{c \, r_{k-1}^3} (y_k - y_{k-1}) \,,$$

with the range parameter defined as: $r_k = \sqrt{x_k^2 + y_k^2}$. Incorporating the random walk model definitions of (8) into the previous result leads to the conditional PDF:

$$p(\tau_k | \tau_{k-1}, r_{k-1}) = \mathcal{N}(\tau_{k-1}, \sigma_\tau^2) \,, \qquad (10a)$$

$$\sigma_\tau = \frac{\sigma_X}{r_{k-1}} \sqrt{\frac{f_\mathrm{s}^2 d^2}{c^2} - \tau_{k-1}^2} \,. \qquad (10b)$$

The independence assumption of (8) between the x_k and y_k motions results in τ_k and r_k also being independent. This, together with (10), finally leads to the transition density for τ_k being given as:

$$p(\tau_k | \tau_{k-1}) = \int p(\tau_k | \tau_{k-1}, r_{k-1}) \, p(r_{k-1}) \; \mathrm{d}r_{k-1} \,. \qquad (11)$$

Assuming a uniform distribution of r_k between 0 and a maximum range R, $r_k \sim \mathcal{U}(0, R)$, the bottom plot of Figure 2 depicts an example PDF $p(\tau_k | \tau_{k-1} = 0)$ computed as in (11) with $\sigma_X = 0.25\,\mathrm{m}$, $d = 0.3\,\mathrm{m}$ and $R = 5\,\mathrm{m}$. For comparison, this plot also shows a Gaussian PDF with mean and variance identical to those of $p(\tau_k | \tau_{k-1})$. The comparatively thinner mode and heavier tails of $p(\tau_k | \tau_{k-1})$ account for the different dynamics resulting for τ_k depending on whether the target is located in the near-field or far-field.

The prediction step of the PF algorithm presented in Section 3.2 implicitly requires sample values of τ_k to be generated with distribution given by (11). The main issue with generating τ_k from an existing τ_{k-1} value is related to the fact that a specific TDOA does not provide information about the associated range r_{k-1}. This problem is solved here by first generating random values of r_{k-1} according to $p(r_{k-1})$, and subsequently drawing a sample τ_k from a normal PDF with mean and variance defined by (10).

Assumption: at time $k-1$, the set of particles $X_{k-1}^{(n)} = \tau_{k-1}^{(n)}$ and weights $w_{k-1}^{(n)}, n \in \{1, \ldots, N\}$, is a discrete representation of the posterior $p(X_{k-1} | Y_{1:k-1})$.

Iteration: for each new data sample from the sensors, $k = 1, 2, \ldots$, update the particle set as follows:

1. *Prediction:* for each particle, randomly choose a value $r_{k-1} \sim \mathcal{U}(0, R)$, then propagate the particles by sampling from a normal distribution $\widetilde{X}_k^{(n)} \sim \mathcal{N}(X_{k-1}^{(n)}, \sigma_\tau^2)$, where σ_τ is computed according to (10b).
2. *Update:* compute the unnormalized weights as given by $\tilde{w}_k^{(n)} = w_{k-1}^{(n)} \cdot p(Y_k | \widetilde{X}_k^{(n)})$ using the likelihood defined in (7), then normalize the weights according to (3).
3. *Resampling:* if necessary, resample the particles $\widetilde{X}_k^{(n)}, n \in \{1, \ldots, N\}$, to form the new set of particles and weights $\{(X_k^{(n)}, w_k^{(n)})\}_{n=1}^N$, as described in Step 3 of Section 3.2.

Result: the set $\{(X_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ is approximately distributed as the posterior density $p(X_k | Y_{1:k})$, and the current time delay can be estimated according to (4).

Alg. 1. PF-TDE, PF algorithm for time delay estimation.

4.2. Proposed Algorithm

Based on the previous developments, the resulting PF algorithm for time delay estimation, denoted PF-TDE, is summarized in Algorithm 1. In order to allow the algorithm to adapt in cases where the sensor signals have a time-varying strength (due to a non-stationary source signal or a moving target), the data vectors $\mathbf{s}_i(k)$, $i = 1, 2$, are first normalized as follows:

$$\mathbf{s}_i(k) = 2(\sigma_{s_1} + \sigma_{s_2})^{-1} \cdot \big[s_i(k - 2P) \;\; \cdots \;\; s_i(k) \big]^\mathrm{T},$$

with σ_{s_i} the standard deviation of the signal values $s_i(k-j)$, $j \in \{0, \ldots, 2P\}$. This procedure allows the definition of a constant σ_Y parameter in the implementation, regardless of the statistics and dynamic range of the input signals. It also enables the tracking algorithm to automatically account for potential changes in the input signal-to-noise ratio (SNR) during adaptation.

4.3. Discussion

The developments presented in this section highlight the non-Gaussian character of the TDE problem. Tracking algorithms such as the

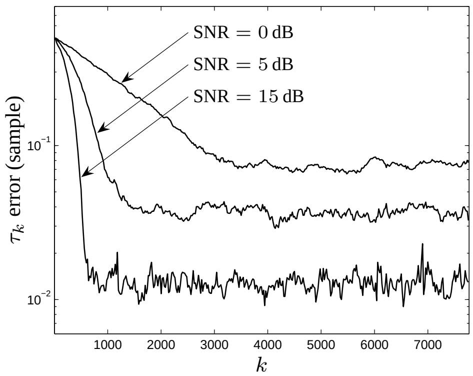

Fig. 3. Absolute time-delay estimation error with white noise signal (results averaged over 100 independent simulation runs).

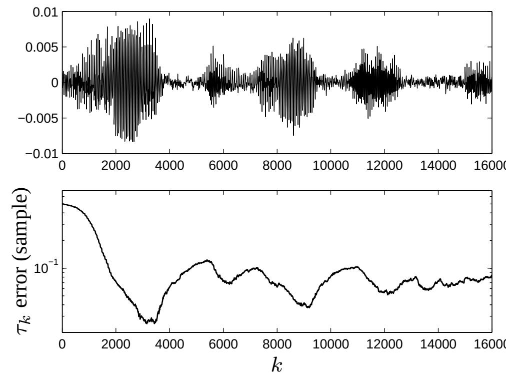

Fig. 4. Simulation results with speech signal. *Top plot:* sensor signal $s_1(k)$. *Bottom plot:* absolute τ_k error (averaged over 100 runs).

Kalman filter and its derivatives would hence lead to sub-optimal results, whereas sequential Monte Carlo methods are able to integrate this information in the development of the resulting algorithm.

In comparison to TDE methods such as the GCC function, the proposed tracker has the advantage of eliminating the search for a function maximum over a set of potential time delays. Constraining the FIR coefficients as in (6) also bypasses the need for interpolation of the filter weights, required in traditional LMS implementations to yield subsample time delays. Finally, the PF approach involves a model of the TDE "dynamics", effectively limiting the variations of the tracked parameter to a practically relevant range.

5. SIMULATION RESULTS

This section presents an overview of the tracking results obtained for the PF-TDE method. The various algorithm parameters have been optimized manually to maximize the tracking performance, while ensuring that the algorithm is able to track time-varying time delays for a variety of target trajectories and velocities. The PF-TDE method was implemented with $P = 25$, $N = 50$ particles, and a resampling threshold $N_\text{t} = 32.5$. For the model parameters, $\bar{v} = 1\,\text{m/s}$, $R = 3\,\text{m}$, and $\sigma_Y = 0.1$. Also, in order to avoid singularities during the computations, a small lower limit was used when generating random range values in the prediction step: $r_{k-1} \sim \mathcal{U}(0.1, 3)$.

Figures 3 and 4 show the performance results obtained with a stationary source with position $\boldsymbol{p}_\text{s} = [1.8\,\text{m}\;\; 3.5\,\text{m}]^\text{T}$, and emitting white noise sampled at $f_\text{s} = 16\,\text{kHz}$. The two sensor locations were defined as $\boldsymbol{p}_1 = [1.35\,\text{m}\;\; 1\,\text{m}]^\text{T}$ and $\boldsymbol{p}_2 = [1.65\,\text{m}\;\; 1\,\text{m}]^\text{T}$. The signals $s_i(k), i = 1, 2$, were generated by convolving the source data with fractional-delay filters corresponding to the time of arrival from the target to each sensor, and corrupted with additive white Gaussian noise to yield a desired SNR level.

Figure 3 gives an overview of the convergence and tracking results for several SNR values. To this purpose, PF-TDE was initialized half a sample off the true source's TDOA, i.e. $X_0^{(n)} = \tau_\text{s} + 0.5, \forall n$. Figure 4 shows a typical result of the proposed algorithm's tracking performance when speech is used as non-stationary source signal. The sensor data was generated according to the procedure described above with $15\,\text{dB}$ SNR.

6. CONCLUSION AND FUTURE WORK

In this paper, a particle filtering method was proposed for the problem of estimating the TDOA of a source signal received at two spatially separated sensors, working on a sample-by-sample basis. Simulation results show that the proposed algorithm is able to successfully track the time-delay parameter with an estimation error of less than 0.1 sample for a range of SNR values, and for stationary signals as well as speech. Future developments of the proposed PF algorithm will include: *i)* attempt to improve the proposed method by making use of additional sensors and including the attenuation parameter α in the state vector; and *ii)* performing a comparative assessment of the tracking accuracy in reverberant conditions and with moving targets, with respect to other existing TDE methods.

7. REFERENCES

[1] *IEEE Trans. ASSP, Special Issue on Time Delay Estimation*, vol. 29, no. 3, June 1981.

[2] H. C. So, "LMS-based algorithm for unbiased FIR filtering with noisy measurements," *Electron. Lett.*, vol. 37, no. 23, pp. 1418–1420, Nov. 2001.

[3] Y. Tan, "Time-varying time-delay estimation for nonlinear systems using neural networks," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 1, pp. 63–38, 2004.

[4] G. C. Carter, *Coherence and time delay estimation*, IEEE Press, Piscataway, N.J., 1993.

[5] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. ASSP*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[6] F. Reed, P. Feintuch, and N. Bershad, "Time delay estimation using the LMS adaptive filter—static behavior," *IEEE Trans. ASSP*, vol. 29, no. 3, pp. 561–571, June 1981.

[7] B. D. O. Anderson and J. B. Moore, *Optimal filtering*, Dover Publications, New York, 2005.

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

[9] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[10] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*, Springer-Verlag, New York, 2000.

[11] Y. T. Chan, J. M. Riley, and J. B. Plant, "Modeling of time delay and its application to estimation of nonstationary delays," *IEEE Trans. ASSP*, vol. 29, no. 3, pp. 577–581, June 1981.