Wideband target tracking by using SVR-based sequential Monte Carlo method

Nihat Kabaoğlu a,*, Hakan A. Çirpan b

a Department of Electronics Engineering, Maltepe University, Marmara Eğitim Köyü, Maltepe, 34857 Istanbul, Turkey
b Department of Electrical and Electronics Engineering, Istanbul University, Avcılar, 34850 Istanbul, Turkey

ARTICLE INFO

Article history:
Received 29 August 2007
Received in revised form 10 May 2008
Accepted 9 June 2008
Available online 13 June 2008

Keywords:
Wideband array processing
Target tracking
Particle filters
Sequential Monte Carlo
Support vector machines
Support vector regression

ABSTRACT

In this work, a support vector regression (SVR) based sequential Monte Carlo method is presented to track wideband moving sources using a linear and passive sensor array for a signal model based on buffered data. The SVR method is employed together with a particle filter (PF) method to improve the PF tracker performance when a small sample set is available. SVR is used as a sample producing scheme for the current state vector. To provide a good approximation of the posterior density by means of improving the sample diversity, samples (particles) are drawn from an importance density function whose mean and covariance are calculated by using the pre-estimating state vector and the state vector’s previous estimate. Thus, a better posterior density than the classical one can be obtained. Simulation results show that the method proposed in this work performs better than the classical one when a small sample set is available. Moreover, the results also show that a modified signal model that utilizes buffering data is superior to the signal model in Ng et al. [Application of particle filters for tracking moving receivers in wireless communication systems, in: IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Rome, Italy, June 2003, pp. 575–579].

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The major problems in array processing are the detection of the number of sources, the estimation of the parameters (like the direction of arrival (DOA) or intersensor delay (ISD) of the sources), and the recovery of the incident source signals through the exploitation of data collected with an antenna array. In the literature, there are many methods proposed by different researchers under narrowband and wideband scenarios to estimate the DOA or the ISD. For wideband scenarios, common approaches (like [1–5]) sample the spectrum of incoming signals at each sensor to form an array of narrowband signals. Then, traditional array processing algorithms developed for the narrowband case can be applied to the transformed signals. However, this approach presents some difficulties. These difficulties arise mainly during the transformation of data into the frequency domain, which requires one to work with complex numbers. In order to avoid the difficulties that arise from working with complex numbers, the problem may be worked on in the time domain rather than in the frequency domain. In the time domain, it is possible to use a model where a bandlimited wideband source signal is expressed in terms of its discrete samples utilizing an interpolation function. Similar methods of expression are given in detail in [6–10]. Moreover, in [12], a method based on this was utilized to solve these problems. The presented signal model in [12] is based on utilizing a reconstruction (interpolation) matrix depending on the ISD instead of on an array steering matrix.

* This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 104E130.
* Corresponding author.
E-mail address: nihatk@maltepe.edu.tr (N. Kabaoğlu).

0165-1684/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2008.06.006
The target tracking problem is of great importance in many communication systems, like radar, sonar, wireless personal security, etc. The main concern of many of the applications in these areas is proposing an algorithm that is able to track the targets. In the literature, many methods have been proposed for tracking moving target(s). When subspace-type methods are excepted, the rest of these methods are collected into two main groups: those methods based on the principle of maximum likelihood (ML) (or its iterative solutions such as recursive expectation-maximization (REM) method [13], space alternating generalized expectation-maximization (SAGE) method [14], and other gradient- or Newton-type algorithms [15–17]), and methods based on Bayesian philosophy or its numerical solutions, such as PF methods like Rao-Blackwellization [18], unscented Kalman filter [19,20], Markov chain Monte Carlo (MCMC) PF [12,21], etc. All of these use each observation once. Accordingly, they are suitable for on-line processing. Moreover, since PF-based methods are adequate for on-line processing they have recently been applied to target tracking problems successfully.

In the PF algorithm, the posterior probability density is represented with an equivalent histogram by a set of weighted particles (samples). To properly achieve this, a large number of particles is required to be used. However, increasing the sample size increases the computational complexity, thus decreasing the efficiency of the PF algorithm. When the PF algorithm is used with a small sample size, the effectiveness and diversity among the samples is reduced due to the inaccuracy of the sample weights. Utilization of resampling to solve the degeneracy problem, which is the most common problem with PF methods, aggravates effectiveness and diversity problems. One of the existing methods proposed in [22] to overcome this problem is the use of a forward filter backward smoothing technology. Since the PF algorithm needs a state-space model, the conventional strategy is to add a random walk to the parameters, and then augment the state space within the parameters for joint estimation. However, the use of a random walk results in an increase in the covariance of parameters, thereby diffusing the posterior densities more than the actual ones. Kernel smoothing [23] is an effective approach to improve the covariance issue. (A comprehensive description of PF methods can be found in [24].)

The SVM-based method proposed as a tracker is also fast enough to run in real time similar to the PF in [25,26]. The SVM is motivated by Vapnik’s statistical learning theory (as presented in [27,28]), which was primarily constructed to address binary classification problems. The SVM is actually a maximal margin algorithm that seeks to place a hyperplane between classes of points such that the distance between the closest points are maximized. In this framework, the SVM is organized under the structural risk minimization principle instead of the traditional empirical risk minimization principle. The field this work focuses on is a function approximation using SVM, commonly called SVM for regression, or SVR. A key element of SVM for regression is the cost function which comprises the sum of loss functions describing the fitting error of training data and a stabilizer controlling the smoothness of the desired regression function. However, with SVM for regression, different loss functions (e.g., linear epsilon-insensitive, quadratic, and Huber’s functions) are adopted to reduce the number of support vectors. Thus the desired sparse representation is obtained by reducing the complexity of the resulting approximation function. Using the SVM methodology, an effective, sample producing scheme is obtained.

In this study, a PF algorithm based on SVR is proposed as a tracker within the sequential Monte Carlo framework in order to obtain sufficient performance of the PF algorithm with a small sample set. A PF algorithm using SVR was proposed for the visual tracking problem in [29] where an algorithm is presented for tracking an object in a video sequence. In the algorithm in [29], SVR is utilized to re-weight the particles sampled by one of the classic re-sampling methods. Since this method focuses directly on the particles and their weights, SVR may destroy the convergence of the particles. However, in our work, before each iteration step, the state vector at the current time step is pre-estimated by SVR. The particles are then drawn from an importance density function having a mean that is the same as that of the pre-estimated state vector and a covariance matrix obtained through the use of the pre-estimated state vector and the estimate of the state vector at the previous time step. Later, sample weights are calculated with current measurement data to obtain an estimate of the state vector. After the re-sampling process, an updated set of particles and a set of weights associated with those particles are determined. All of these steps are processed for each time step \( n \). Consequently, a smoothed posterior density of the state for each time step \( n \) can be better obtained than the one arrived at with classical PF when a small sample set is available. Furthermore, this work also contributes to improving the signal model presented in [11] in a small way through the modified signal model requiring buffered data. It is presumed that the signal model in this work eliminates the error caused by the approximation used in the signal model in [11] and then improves the estimation performance.

This paper is organized as follows. Section 2 presents the data model under the wideband assumption. The state-space model and derivation of the joint posterior density function to be used for constituting the methods and algorithms used in this work, as well as the derivation of the importance density function for both the classical PF and the proposed PF algorithm are presented in Section 3. Section 4 presents the simulation results for different scenarios. After an analysis of these results, a discussion of the proposed method is presented in Section 5.

2. Data model for moving wideband sources

In this paper, a modified version of the data model proposed in [11] is used. The models in [11,12] are based on the estimation error between the sensor outputs and their approximation constituted by a linear combination of \( L - 1 \) past samples, but the model used in this work is based solely on snapshots. This means that the data model
used in this work requires the buffering of extra samples. However, this improves the estimator performance. Due to the similarity between the signal model used in this work and the ones in [11,12], the following description of the signal model is based on the one presented in those papers.

A scenario is considered in which \( K \) wideband signals, \( s_k(t) \), for \( k = 0, \ldots, K - 1 \), from distinct unknown time-varying directions \( \theta_k(t) \) are received onto an uniform linear array of \( M \) identical sensors. The number of sources \( K \) is assumed to be known, but if desired, using the buffered data, it can be estimated by means of various methods (e.g., the Akaike information criterion, the minimum description length, the eigenvalue decomposition, the transformation on the correlation matrices, the hypothesis testing and the MCM).

The frequency response of each wideband signal is band limited to

\[
|f| \in [f_{k}^{L}, f_{k}^{U}], \quad f_{k}^{L} = f_{k}^{0} + \Delta f_{k},
\]

where \( f_{k}^{0} \) and \( f_{k}^{U} \) are the lower and upper frequencies, and \( \Delta f_{k} \) is the bandwidth of the \( k \)th signal. The maximum time delay \( T_{\text{max}} \) experienced by an incident source wave as it transverses between two adjacent sensors can be defined

\[
T_{\text{max}} = \min_{k = 0, \ldots, K - 1} \left( \frac{1}{2f_{k}^{U}} \right).
\]

The time-varying direction vector \( \Theta(t) \) and corresponding ISD vector \( \tau(t) \) are defined, respectively, as

\[
\Theta(t) = [\theta_{0}(t), \theta_{1}(t), \ldots, \theta_{K-1}(t)]^{T},
\]

\[
\tau(t) = [\tau_{0}(t), \tau_{1}(t), \ldots, \tau_{K-1}(t)]^{T}.
\]

The relationship between the ISD of the \( k \)th source by \( \tau_{k}(t) \) and \( \theta_{k}(t) \) at time \( t \) is as follows:

\[
\tau_{k}(t) = \frac{\lambda}{C} \sin \theta_{k}(t),
\]

where \( \lambda \) is the interspacing of the sensors and \( C \) is the speed of propagation, such that

\[
|\tau_{k}(t)| \leq \frac{1}{2f_{k}^{U}} = \min_{k = 0, \ldots, K - 1} \left( \frac{1}{2f_{k}^{U}} \right) = T_{\text{max}}.
\]

The signal undergoes a delay as it progresses along the array. In contrast to the narrowband case, the delayed signal \( s_k(t - \tau_k(t)) \), due to the \( k \)th incident source, cannot be determined by a corresponding phase shift. Moreover, since the time delay is seldom an integer multiple of the sampling frequency, interpolation is required to estimate the time-shifted version \( s_k(t - \tau_k(t)) \) from samples of the received signal. A modified version of the data model proposed in [11] requires buffering of the received data and uses signal interpolation to estimate the time-shifted versions of the wideband signals as they transverse a linear array.

A snapshot vector \( y(t) \in \mathbb{R}^{M \times 1} \) at time \( t \) is denoted, and, according to [12], this snapshot vector represents the data received by a linear array of sensors as follows:

\[
y(t) = \sum_{k=0}^{K-1} s_k(t) + \sigma_n w(t),
\]

where the quantities \( s_k(t) \), the \( k \)th signal, and \( s_k(n) \), the corresponding discrete-time version, are defined as

\[
s_k(t) = [s_k(t), \ldots, s_k(t - (M - 1)\tau_k(t))]^{T}
\]

and

\[
s_k(n) = [s_k(n), \ldots, s_k(n - (L - 1))]^{T}.
\]

In (8), the interpolation matrix \( \bar{R} (\tau_k(t)) \) interpolates the discrete-time sequences \( s_k(n) \) to give the desired sequences \( s_k(t) \) which correspond to the signals from the \( k \)th source at time \( t \), and \( w(t) \in \mathbb{R}^{M \times 1} \) is an independently identically distributed (\( iid \)) Gaussian random variable which is uncorrelated with the signal, defined as

\[
w(t) \sim \mathcal{N}(\mathbf{0}, \Sigma_{\text{noise}}),
\]

where \( \mathcal{N}(\mathbf{0}, \Sigma) \) denotes the multivariate normal distribution with mean \( \mathbf{0} \) and covariance \( \Sigma \) and \( \sigma_{\text{noise}} \) is the observation noise variance, and \( \bar{R} (\tau_k(t)) \in \mathbb{R}^{M \times L} \) represents an \( L \)th-order interpolation matrix for the sources [12]. It is defined as

\[
\bar{R} (\tau_k(t)) = \left\{ \begin{array}{cl}
\mathbf{R} (\tau_k(t)) & \tau_k(t) > 0, \\
\mathbf{E}_{\mathbf{M}} \mathbf{R} (\tau_k(t)) & \tau_k(t) < 0.
\end{array} \right.
\]

We now define a new set of interpolation matrices \( \bar{R}_l (\tau(t)) \in \mathbb{R}^{M \times L} \), for \( l = 0, \ldots, L - 1 \), which is a collection of all the \( l \)th columns of the individual interpolation matrices of \( \bar{R} (\tau(t)) \) from (8) for \( k = 0, \ldots, K - 1 \), i.e.,

\[
\bar{R}_l (\tau(t)) = [\bar{R}_l (\tau_0(t)), \bar{R}_l (\tau_1(t)), \ldots, \bar{R}_l (\tau_{K-1}(t))].
\]

where \( \bar{R}_l (\tau(t)) \in \mathbb{R}^{M \times 1} \) is the \( l \)th column of \( \bar{R} (\tau(t)) \). After rearranging (8) by using (14) and (15), the snapshot vector \( y(n) \) at \( n = nT_s \) (where \( T_s \) is the sampling period and is normalized to unity for notational convenience from this

\[1\] A half-sinc approach is used for interpolating the waveforms instead of a full-sinc approach in order to obtain a causal signal model. This causes an error, but use of a window function makes this error small. A detailed description of the signal model can be found in [12].
3. Derivation of importance density functions

3.1. Derivation of the joint posterior density function

The amplitude vector is chosen to be a distribution with zero mean and a covariance matrix corresponding to maximum entropy prior, as in [30]. $\mathbf{x}(n) \sim N(0, \delta^2 \sigma_w^2 [\mathcal{A}(\tau(n))\mathcal{A}^T(\tau(n))]^{-1})$ where $\mathcal{A}(\tau(n)) = \sum_{l=0}^{L-1} \mathbf{R}_l(\tau(n))$, and the ISD is assumed to change according to

$$\tau(n) = \tau(n-1) + \sigma_w \mathbf{v}(n),$$

where $\delta^2$ is a hyperparameter closely related to the signal-to-noise ratio, and the noise vector $\mathbf{v}(n) \in \mathbb{R}^K$ is an iid Gaussian random variable with zero mean and covariance matrix $\mathbf{I}_K$. In (17), $\sigma_w^2$ indicates the state noise variance.

In the proposed system of equations, the noise variances $\sigma_w^2$ and $\sigma_v^2$ are assumed to be unknown and constant over time. The unknown vectors of amplitudes $\mathbf{x}(n)$ are assumed to be iid among snapshots.

The vector of all parameters $\mathbf{\Psi}$ describing the received signal model can be defined as

$$\mathbf{\Psi} \triangleq [\tau_{1:n}, \mathbf{x}_{1:n}, \sigma_w^2, \sigma_v^2]^T,$$

where $\cdot|_{1:n}$ indicates all the elements from time 1 to time $n$.

The distribution of interest is $p(\mathbf{\Psi}_{1:n}|\mathbf{y}_{1:n})$, which can then be expanded using appropriately selected prior distributions of parameters, according to Bayes' theorem thus

$$p(\mathbf{\Psi}_{1:n}|\mathbf{y}_{1:n}) \propto p(\mathbf{y}_{1:n}|\mathbf{\Psi}_{1:n}, \mathbf{x}_{1:n}, \sigma_w^2, \sigma_v^2) p(\mathbf{x}_{1:n}|\tau_{1:n}, \sigma_v^2) p(\tau_{1:n}|\sigma_v^2) p(\sigma_v^2) p(\sigma_w^2),$$

where $p(\mathbf{y}_{1:n}|\mathbf{\Psi}_{1:n}, \mathbf{x}_{1:n}, \sigma_w^2, \sigma_v^2)$ is the likelihood term, and the other distributions compose the joint prior distribution for the parameter vector $\mathbf{\Psi}$. Given these states, the observations are assumed to be iid, together with the state conditional update likelihood. Distributions for each of the terms in (19) can be assigned as

$$p(\mathbf{y}_{1:n}|\mathbf{\Psi}_{1:n}, \mathbf{x}_{1:n}, \sigma_w^2, \sigma_v^2) = \prod_{j=1}^{n} \mathcal{N}(\mathbf{y}_j; \mathbf{R}_j(\tau_j) \mathbf{x}_j, \sigma_w^2),$$

$$p(\tau_{1:n}|\sigma_v^2) = \prod_{j=1}^{n} \mathcal{N}(\tau_j; \tau_{j-1}, \sigma_v^2),$$

$$p(\mathbf{x}_{1:n}|\tau_{1:n}, \sigma_v^2) = \prod_{j=1}^{n} \mathcal{N}(\mathbf{x}_j; \mathbf{R}_j(\tau_j) \mathbf{x}_j, \sigma_v^2) \mathcal{N}(\mathbf{x}_j; \mathbf{0}, \sigma_v^2),$$

where

$$\mathbf{\Sigma}(\tau_j) = \begin{bmatrix} \frac{1}{\delta^2} \mathcal{A}(\tau_j) \mathcal{A}^T(\tau_j) + \mathbf{R}_0^2(\tau_j) \mathbf{R}_0(\tau_j) \\ \mathbf{R}_0(\tau_j) \end{bmatrix},$$

$$\mathbf{m}_x = \mathbf{\Sigma}^{-1}(\tau_j) \mathbf{R}_0(\tau_j) \mathbf{y}_j - \sum_{l=1}^{L-1} \mathbf{R}_l(\tau_l) \mathbf{x}_l.$$
analytically integrated out over $x_n$ which yields

$$p(\tau_{1:n}, \sigma^2_w, \sigma^2_w | y_{1:n}) \propto \prod_{j=1}^{n} \left\{ \left( \frac{1}{2\pi\sigma_w^2} \right)^{1/2} \exp \left\{ -\frac{1}{2\sigma_w^2} (y_j - \sum_{l=0}^{L-1} \tilde{R}_i(\tau_i)x_{j-l})^T \right\} \right. \times \exp \left\{ -\frac{1}{2\sigma^2} (\tau_i - \tau_{j-1})^T (\tau_i - \tau_{j-1}) \right\} \left. \right\}$$

(28)

The MAP estimators of the variances can also be obtained as follows:

$$\hat{\sigma}_w^{2_{\text{MAP}}} = \frac{\sum_{j=1}^{n} (y_j - \sum_{l=0}^{L-1} \tilde{R}_i(\tau_i)x_{j-l})^T (y_j - \sum_{l=0}^{L-1} \tilde{R}_i(\tau_i)x_{j-l})}{Mn + 2}.$$  

(29)

$$\hat{\sigma}_w^{2_{\text{MAP}}} = \frac{\sum_{j=1}^{n} (\tau_i - \tau_{j-1})^T (\tau_i - \tau_{j-1})}{Kn + 2}.$$  

(30)

Since the amplitudes and noise variances can be estimated by using their MAP procedure, the parameter vector $\Psi_{1:n}$ can be reduced to $\tau_{1:n}$ to be sampled with the PF, as $\Psi_{1:n} \equiv \tau_{1:n}$. Furthermore, the posterior density in (28) cannot be directly sampled, an importance density function should be obtained. In the next section, derivation of the importance density function is determined to be the main concern when a small sample set is available.

3.2. Derivation of the importance density function according to [31]

The descriptive idea of the PF algorithm is to use a set of random samples with a set of associated weights to approximate the posterior probability density function. The PF algorithm exploits the sequential Monte Carlo method for on-line inference within a Bayesian framework. For each time sample, prediction and update processes are executed consecutively. If a large sample set is used, the approximation of the posterior probability density function can successfully be done. When a large sample set is considered, the PF algorithm’s well-known problem of degeneracy is encountered. To overcome the degeneracy problem, the re-sampling method is used. The generic PF algorithm for the problem focused on in this work is given in Table 1.

The function $p(\Psi_{n} | \Psi_{n-1}, y_n)$ is known as the importance function or the importance density. The main issue related to the importance density is how to choose it. One approach is to use the prior distribution $p(\Psi_{n} | \Psi_{n-1})$, which is the most common choice. In this case, the particle weights can be determined by the likelihood function only. Thus, it is clear that the algorithm will exhibit a poor performance since the importance density is independent of the current measurement. Therefore, a more reliable importance density should be used. One choice of an optimal importance density is given in [31], though it is computationally complex. At each observation time $n$, particles are generated from an optimal importance function which, according to [31], is given as

$$L(\Psi_n) = L_p(\Psi_n) + L_q(\Psi_n),$$

(31)

where $L_p(\Psi_n) = \log p(y_n | \Psi_n)$ and $L_q(\Psi_n) = \log p(\Psi_{n} | \Psi_{n-1})$, $p(y_n | \Psi_n)$ and $p(\Psi_{n} | \Psi_{n-1})$ can be obtained, respectively, by using (16) and (17). So,

$$\pi(\Psi_{n} | \Psi_{n-1}, y_n) \sim \mathcal{N}(m_n, C_n),$$

(32)

which has the following covariance and mean:

$$C_n = -\left( \nabla^2 L(\Psi_n) \right)^{-1},$$

(33)

$$m_n = \Psi_{n-1} + C_n \nabla L(\Psi_n),$$

(34)

where $\nabla L(\Psi_n)$ and $\nabla^2 L(\Psi_n)$ represent the gradient and the Hessian matrix of $L(\Psi_n)$, respectively. This importance function is a Gaussian approximation of the optimal importance function, and as such it can be easily sampled.

3.3. Derivation of the importance density function based on SVR

While the PF algorithm presents a way in which to study non-linear/non-Gaussian models, it requires that the particles must constitute a large sample set to get the
desired approximation of the posterior density due to the inaccuracy of the sample weights. A large sample set naturally induces a computational complexity and the problem of sample degeneracy. To tackle these problems, a small sample set and re-sampling should be used. While a small sample set decreases the performance of the PF algorithm, re-sampling activates a sample deprivation that causes a loss of effectiveness and diversity among the samples. If the diversity of the particles is preserved, the satisfactory approximation of the posterior density can be obtained, which allows sample deprivation problem to be avoided. Both to prevent the sample degeneracy problem and to represent a better posterior density than that represented in a classical PF algorithm, non-linear SVR is a suitable model.

Non-linear SVR maps the sample points onto a higher dimensional space using non-linear mapping (which uses a map \( \phi : \mathbb{R}^f \rightarrow \mathbb{R}^h \) where \( h \) is greater than \( f \)). Searching a separating hyperplane in the higher dimensional space is equivalent to searching a non-linear separating surface in \( \mathbb{R}^f \). While working in the higher dimensional space, mapped data \( \phi(y_n) \) is used, where \( y_n \) and \( \phi(y_n) \) represent two different data points. In order to decrease the computational complexity, a kernel function \( \text{K}_r(y_n, y_m) = \phi(y_n) \phi(y_m) \) is used. This means that the kernel function can be used instead of \( y_n, y_m \) everywhere in the optimization problem without ever needing to know definitely what \( \phi \) is.

When a new data sample is received, the output of the SVR can be found by using the following regression function [32]:

\[
f(y_n) = \sum_{v=1}^{N_t} \left( \beta_v - \beta_v^* \right) \Psi_v \phi(y_n) + b
\]

\[
= \sum_{v=1}^{N_t} \left( \beta_v - \beta_v^* \right) \Psi_v \text{K}_r(y_n, y_v) + b.
\]

(35)

where \( N_t \) denotes the number of support vectors, \( \beta_v \) and \( \beta_v^* \) are the Lagrange multipliers, \( \Psi_v \) is the corresponding output to \( y_n \) in the training phase, and \( b \) is the bias term.

The most used kernel functions are the polynomial kernel \( \text{K}_p(y_n, y_m) = (y_n y_m + 1)^{2 \delta} \), the sigmoid kernel \( \text{K}_s(y_n, y_m) = \tanh(\kappa y_n y_m - \delta) \) and the Gaussian radial basis function kernel \( \text{K}_g(y_n, y_m) = e^{\|y_n - y_m\|^2/2\sigma^2} \).

In order to find a map that fits the training input set into the given form of the output set, let us consider the problem of approximating the set of data \( D_r \):

\[
D_r = \{ D_j; j = 1, \ldots, K \},
\]

(36)

where \( D_j = \{ (y_{n-1}, \hat{\Psi}_{n-1}^{(j)}), (y_{n-1}+1, \hat{\Psi}_{n-1}^{(j)}), \ldots, (y_{n-1}, \hat{\Psi}_{n-1}^{(j)}) \} \), and \( y_{n-1} \), and \( \hat{\Psi}_{n-1}^{(j)} \), respectively, denote the input data vector and the corresponding output value for \( l = 1, \ldots, l \) and \( l \) indicates the number of the elements in the training set, which has the same length as that of the buffered data. Thus, we can construct the regression function \( f_j \) over \( D_r \):

\[
f_j : \Psi^M \rightarrow \mathbb{R},
\]

(37)

where \( M \) is the dimension of the array output data. Thus, SVR is used to determine \( f_j \) each time a new data sample is received:

\[
f_j(y_n) = \sum_{v=1}^{l} (\beta_v - \beta_v^*) \Psi_v \phi(y_n) + b
\]

\[
= \sum_{v=1}^{l} (\beta_v - \beta_v^*) \Psi_v \text{K}_r(y_n, y_v) + b.
\]

(38)

Eq. (38) implies that the training phase consists of \( L \) data points. Eq. (38) can be expressed in a more compact form as

\[
\hat{\Psi}_n^{(j)} = [1, \hat{\Psi}_n^{(j-Ln-1)} \odot \text{K}_r(y_{n-Ln-1}, y_n)] p_n,
\]

(39)

where \( \hat{\Psi}_n^{(j-Ln-1)} \in \Psi^k, \text{K}_r(.) \in \Psi^k, \odot \) denotes component-wise multiplication and \( p_n \in \Psi^{l+1} \) has its first entry equal to \( b_n \) and remaining entries equal to \( (\beta_v - \beta_v^*) \).

Using \( L \) number of buffered measurement vectors \( \{ y_{n-v}, v = 1, \ldots, L \} \) and their parameter vectors \( \{ \Psi_{n-v}, v = 1, \ldots, L \} \), training sets \( D = \{ D_j; j = 1, \ldots, K \} \) are constructed (here, it is assumed that these parameter vectors were estimated by the classical PF algorithm). Then, they are employed to train an SVR for each element of the parameter vector. Thus, \( K \) number of SVRs are trained simultaneously. At the end of a training phase, an SVR is thus determined for each element of the parameter vector, which means that a set of parameters \( \{ (\beta_v, b_v), v = 1, \ldots, L \} \) belonging to each SVR is obtained. The usage of the SVR in a training phase is shown in Fig. 1. When a new measurement vector is received at instant \( n \), a new \( L \) number of measurement vectors \( \{ y_{n-v+1}, v = 1, \ldots, L \} \) are applied to SVRs as inputs. Finally, a pre-estimation for each element of the parameter vector at instant \( n \) is obtained by using Eq. (39). Each of them is an element of the pre-estimation of the state vector \( \Psi_n \). The usage of the SVR in a test phase is shown in Fig. 2. Once the pre-estimation of the parameter vector \( \Psi_n \) is obtained, sample particles \( \{ \Psi_{n-1}^{(i)} \}; i = 1, \ldots, P \) are drawn from a Gaussian distribution function \( p(\Psi_{n-1}^{(i)} | \Psi_{n-1}, y_{n-Ln}) \) with the following mean and covariance:

\[
\mu_n = \Psi_n,
\]

(40)

\[
C_x = E[(\Psi - \mu_x)(\Psi - \mu_x)^T],
\]

(41)

where \( \Psi \) stands for the estimates of \( \Psi \) obtained at the previous time steps. In practice, computation of \( E[.] \) is evaluated through taking average of available estimates from the previous time steps. However, its computation through the use of only one previous sample does not effect the performance of the proposed algorithm significantly. In the light of this explanation, the covariance matrix can be computed based only on the previous estimate ignoring the rest, so (41) can be modified as follows:

\[
C_x = \frac{1}{K} (\Psi_{n-1} - \mu_x)((\Psi_{n-1} - \mu_x)^T,
\]

(42)

where \( \Psi_{n-1} \) stands for the estimate of \( \Psi_n \) obtained at the previous time step. Later, the classical re-sampling method can be applied to these particles if effective sample size is below a certain limit. All these processes are executed for each time step \( n \). To construct new training sets at the current time step, \( L \) number of measurement vectors used in the test phase of the SVR at the previous time step, \( \{ y_{n-v+1}, v = 1, \ldots, L \} \), and the
estimates of the parameter vector obtained at the previous time steps, \( \{ \hat{\Psi}_{n-L}^{(v)}, ..., \hat{\Psi}_{n-L+1}^{(v)} \} \), are used. In order to obtain a pre-estimate of the state vector at the current time step, the current measurement vector is included in the previous training set and the first one is excluded from this set. Thus, the amount of data in a new training set is fixed at each time step. These training points constitute a moving window over the data. Consequently, a smoothed and well-approximated posterior density of the state for each time step \( n \) can be obtained by using this importance density function when a small sample set is available. The proposed PF algorithm is given in Table 2.

4. Simulations

In this section, some simulation results are presented in order to demonstrate the superiority of the PF algorithm integrated with SVR method versus the classical PF algorithm and the proposed signal model versus the model given in [11]. In the following simulations, a uniform linear array composed of \( M = 10 \) omnidirectional antenna elements with a half-wavelength spacing of the elements is used. In these experiments, \( K = 2 \) Gaussian processes are generated for the moving sources that have a zero mean with variance \( \sigma_n^2 \). These wideband source signals are bandlimited to the frequency values \( f \in [100 \text{ Hz}, 600 \text{ Hz}] \), thus the bandwidth of the signals is 500 Hz. The spectrum of these signals is illustrated in Fig. 3. All necessary data required to train the SVM is obtained by using the classical PF algorithm. To train and to test the SVM a Gaussian radial basis kernel function is used.

4.1. Tracking of ISDs and signal recovery

In these simulations, we demonstrate the ability of the proposed algorithm in tracking the ISDs \( \tau(n) \) of the sources\(^2\) and recovering the source waveforms \( x(n) \) for the wideband scenario. The ISDs of the \( K = 2 \) sources are generated as autoregressive (AR) processes whose coefficients represent a 9th order low-pass Butterworth filter, with normalized cutoff frequency 0.1. The variance of the

\(^2\) Once the ISDs for the sources are available, the DOA of the sources, \( \theta(n) \), are determined through (5).
Table 2
Algorithm of PF with SVR

1. Initial phase of the algorithm (for \( n \leq L \)):
   - Use the classical PF algorithm in order to obtain the estimates of state vectors \( \Psi_n \), \( i = n \) to \( n - L \).
   - Construct the training sets \( D = \{ D_j, j = 1, \ldots, K \} \) for each SVR, where \( D_j = \{ (y_{n-L}, \Psi_n^{(i)}), (y_{n-L+2}, \Psi_n^{(i)}), \ldots, y_n, \Psi_n^{(i)} \} \).
   - Train the SVRs and then obtain a set of parameters \( (\beta_n, b_n) \) for each SVR.

2. Predict and update (for \( n > L \)):
   - Find a pre-estimation of the state vector \( \Psi_n \) by using the equation in (39).
   - For \( i = 1, \ldots, P \), draw samples \( \Psi_n^{(i)} \) from \( \pi(\Psi_n | x_{n-L}) \), where \( x_{n-L} \) is obtained from Eqs. (40) and (42), respectively.
   - For \( i = 1, \ldots, P \), evaluate un-normalized weights:
     \[ \tilde{w}_{n}^{(i)} = \frac{p(y_n | \Psi_n^{(i)})(\Psi_n^{(i)} | x_{n-L})}{\pi(\Psi_n | x_{n-L})} \]
   - For \( i = 1, \ldots, P \), normalize weights:
     \[ w_{n}^{(i)} = \frac{\tilde{w}_{n}^{(i)}}{\sum_{i=1}^{P} \tilde{w}_{n}^{(i)}} \]

3. Re-sample particles if effective sample size \( N_{\text{eff}} = \frac{1}{\sum_{i=1}^{P} w_{n}^{2}} \) is below a certain threshold:
   - For \( i = 1, \ldots, P \), sample an index \( j \) distributed according to \( P(j = p) = w_{n}^{(i)} \) for \( p = 1, \ldots, P \), then set \( \Psi_n^{(i)} = \Psi_n^{(j)} \) and \( w_{n}^{(i)} = 1/P \)

4. Preparation of SVRs for the next time step:
   - Construct new training sets \( D = \{ D_j, j = 1, \ldots, K \} \) for each SVR, where \( D_j = \{ (y_{n-L}, \Psi_n^{(i)}), (y_{n-L+2}, \Psi_n^{(i)}), \ldots, y_n, \Psi_n^{(i)} \} \).
   - Train the SVRs, and then obtain a set of parameters \( (\beta_n, b_n) \) for each SVR.

AR process is \( \sigma_0^2 \). Likewise, the source waveforms are also generated AR processes, corresponding to a 10th order low-pass Butterworth filter with normalized cutoff frequency 0.3 and variance \( \delta^2 \).

The initial values for the ISDs are considered to be \( \tau(0) = [-75.67 \mu s, 75.67 \mu s] \). The time-varying ISDs evolve according to (17). The number of particles, number of time samples, buffered data length and SNR are chosen as \( P = 150, N_t = 100, L = 4 \) and \( \text{SNR} = 10 \text{dB} \), respectively. Fig. 4 presents the sequential estimates of the ISDs estimated using both the PF algorithm integrated with the SVR method and the classical PF algorithm for the proposed signal model. The sequential estimates of the ISDs estimated using both the PF algorithm integrated with the SVR method and the classical PF algorithm for the proposed signal model and the signal model in [11] are shown in Fig. 5, and the sequential estimates of the signal amplitudes are presented in Fig. 6. According to Fig. 4, the proposed PF algorithm integrated with the SVR method is superior to the classical PF algorithm. Furthermore, Fig. 5 shows that the proposed method for the proposed signal model estimates both ISDs better than the proposed method for the signal model given in [11]. If desired, noise variances \( \sigma_n^2 \) and \( \sigma_r^2 \) can also be sequentially estimated by using their MAP procedure in (29) and (30) as well signal amplitudes in (27).

4.2. Performance evaluation

A comparison between the estimate variances of \( \tau(n) \) for both signal models and posterior Cramer–Rao bounds (PCRBs) is given in Fig. 7. One can observe from Fig. 7 that the variances of the estimates are closer to the PCRBs.
To exhibit the performance of the proposed estimator, the same scenario is used for different SNR values. Variances of the estimates obtained using the method presented in this work versus different SNR values are shown in Fig. 8. From this figure, it can easily be seen that the variances of the estimates are reduced as SNR value increases.

A particular time sample at $n = 30$ for the evolution of $\tau(n)$ has been selected to present the comparison between the PCRB and the performance of the proposed PF.
algorithm for the signal model in [11] and the proposed signal model in terms of estimated variances of \( s(n) \) as a function of SNR. The algorithm is run for 100 independent trials over a range of SNR, from \(-4\) to \(12\) dB. Each trial uses 100 observations. In Fig. 9, which shows this comparison, it can be observed that the proposed method integrated with SVR method for the proposed signal model performs better than that for the signal model in [11]. The reason why the variances of the proposed method for the proposed signal model come closer to theoretical PCRB...
than those of the signal model in [11] is that the proposed signal model uses the buffered data.

5. Conclusion

A signal model different from the classical one used for DOA estimation problems is utilized in this work. This signal model, which is an extended version of the model in [12], is based on an interpolation function and past signal samples. The main contribution of this work is to improve the performance of the PF algorithm when only a small sample set is available. To this end, the SVR is integrated into the sequential Monte Carlo method. The other is to improve the signal model presented in [11] by using buffered data. The proposed signal model in this

![Fig. 8. Averaged error of the proposed PF algorithm with SVR estimates versus the classical PF estimates.](image8.png)

![Fig. 9. Variances of estimates versus SNR for the proposed signal model and the one in [11].](image9.png)
work eliminates the error caused by the approximation used by the signal model in [11]. Thus, the estimator performance is improved.

To properly approximate the posterior density, the PF algorithm needs a large sample set. When a large sample set is considered, the computational cost results in a degradation of the efficiency of the PF algorithm. In this framework, an SVR-based particle producing scheme improves the effectiveness of the PF algorithm, and maintains the diversity of samples. The efficiency and the performance of the presented method are demonstrated through ISDs' tracking of the moving multiple sources problem by comparing it with the classical PF algorithm. Besides the tracking of ISDs, signal amplitudes are estimated. Furthermore, the proposed method for tracking the ISDs of multiple moving sources is compared with the classical PF algorithm and PCRBs in terms of performance. The simulation results show that the proposed PF algorithm is more effective than the classical PF algorithm in the consideration of a small sample set, and the signal model presented in this work performs better than the one in [11].

The most important problem of the proposed estimation method is to appropriately choose the kernel function and the other parameters of the SVM, which has long been a known problem in SVM-based regression algorithms. A new algorithm which estimates the kernels is planned as a future work. Another problem of the proposed estimation method is the buffered data requirement constituting the signal model and the training data set. This problem is unavoidable due to inherent structure of the SVM which needs a training phase, but it provides an advantage on learning the target dynamics for a prediction-based algorithm. For the many recursive algorithms, this advantage also eliminates the necessity of the assumption on the system having Markov dynamics. In the proposed algorithm, an estimation of the target dynamics can be obtained by using a set of pilot data determined with the help of any other estimation method which does not require knowing the system dynamics. Thus, a new algorithm for the system with unknown dynamics can be developed. In the future work, we also intend to devise this type of algorithms.

The PF algorithm and the signal model proposed in this work demonstrate reliable tracking of the ISDs of moving targets in a white noise environment with a uniform linear array of sensors.

Acknowledgment

We would like to thank Rana Ortac Kabaoglu for her help and fruitful discussions on support vector machines.

References


