ABSTRACT

This work presents the application of an equalization method for mobile station terminals using particle filters. To use particle filters for equalization, a mathematical model is presented which allows to represent the transmitted symbols as the state of a stochastic system which can be estimated by particle filters. In this paper an equalizer structure with particle filters is proposed. Several improvement strategies, which help to obtain better estimation results with a lower number of particles, are discussed. Additionally, a performance evaluation of the particle filter equalizer for GSM/EDGE is presented.

1. INTRODUCTION

Particle filters are statistical estimation methods which gained popularity in the last decade. Since the pioneering work of Gordon, Salmond and Smith [1] particle filters are used in many fields, for example image recognition or positioning systems. In the last years the application of particle filters to problems in communications engineering appeared in various publications, such as [2]. For application in baseband algorithms of a mobile station receiver, three main areas for particle filters are supposable.

- symbol estimation
- channel estimation
- combined symbol and channel estimation: blind symbol estimation

In a mobile communication system, usually the channel is estimated first, then, by using this channel information, a symbol estimation is done. Blind symbol estimation means to estimate the transmitted symbols without doing the channel estimation first. This estimation technique is mostly applied where no training data are available to estimate the channel parameters. According to [3] blind symbol estimation with particle filters is only suggestive if no training symbols are used in a communication system. Because of the existence of a Training Sequence Code (TSC) in GSM, it was decided not to apply particle filters for blind symbol estimation. Channel estimation algorithms are well investigated for GSM. Commonly, least mean square algorithms (LMS) are used which obtain good estimation results [4]. For this work those channel estimation algorithms were used to estimate the channel. Then this channel estimation is then used as an input parameter for the presented particle filter symbol estimation algorithm.

2. MATHEMATICAL MODEL

For many communication systems, e.g. GSM/EDGE, the mobile radio channel with the effects of multipath propagation and noise can be described approximately with an Finite impulse response (FIR) Filter and an additional random variable $v_k$ usually assumed to be sampled from a gaussian distribution, as shown in Fig. 1. This model can be represented in the following state space form:

$$x_k = Fx_{k-1} + bs_k \tag{1}$$

$$y_k = h^T x_k + v_k \tag{2}$$
where
\[
\begin{align*}
\mathbf{h}_k &= \begin{bmatrix} h_{k1} \\ h_{k2} \\ \vdots \\ h_{kL} \end{bmatrix}; \quad \mathbf{x}_k = \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-L+1} \end{bmatrix} \\
\end{align*}
\]
(3)

and
\[
\begin{align*}
\mathbf{F} &= \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \\
\end{align*}
\]
(4)

The matrix \( \mathbf{F} \) is used to perform the shifting operation of the input values inside the FIR filter. \( \mathbf{h}_k^T \) represents a vector containing the \( L \) filter coefficients of the multipath channel FIR model (the attenuation factors on certain multipath propagation paths) and \( \mathbf{x}_k \) represents a vector of the last \( L \) data symbols transmitted through the channel. Generally, as pointed out by the index \( k \) the vector of the modelled attenuation coefficients \( \mathbf{h}_k^T \) is time variant. For the practical application of the methods presented in this work the vector \( \mathbf{h}_k^T \) is assumed to be constant for one GSM burst. The idea of using particle filters for equalization was to use them to estimate this state vector \( \mathbf{x}_k \). For symbol estimation, the desired symbol is then extracted out of the most probable state vector.

3. PARTICLE FILTERS

Particle filters can be used to estimate the state \( \mathbf{x}_k \) at the time step \( k \) of a system whose state trajectory can be described by the equations Eqn. 1 and Eqn. 2. Note, that the system model need not to be linear for applying particle filters. Particle filters are then used to estimate the pdf \( p(\mathbf{x}_k|y_k) \) at the time step \( k \). For state estimation the expectation value for a continuous pdf or the state with the maximum probability for a discrete pdf, as it is the case in this work, can be used.

3.1. Foundation of Particle Filters

The desired pdf \( p(\mathbf{x}_k|y_k) \) can be principally computed according to Bayes Theorem:
\[
p(\mathbf{x}_k|y_k) = \frac{p(y_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(y_k)} \\
\]
(5)

Because only the maximum probability is to be found for symbol estimation, \( p(y_k) \) can be omitted:
\[
p(\mathbf{x}_k|y_k) \propto p(y_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\
\]
(6)

If the probability density functions of Eqn. 6 are known \( p(\mathbf{x}_k|y_k) \) could be calculated for every state \( \mathbf{x}_k \). For practical systems with a discrete state space the effort to do this in real time is too high. For example a typical state space of the channel model for 8PSK modulation in an EDGE receiver has the size of the order of \( 10^7 \) elements (typical channel length of 8 with 8 possible values per symbol). Additionally, the equalization problem cannot be solved analytically. That means for exact calculation of the maximum aposteriori probability the required pdf has to be evaluated for every state of the state space. Thus for practical application \( p(\mathbf{x}_k|y_k) \) is approximated. This approximation is done by importance sampling.

3.2. Importance Sampling

If the desired pdf \( p(\mathbf{x}_k|y_k) \) would be known, it could be sampled to obtain a set of \( N \) so called particles:
\[
\chi = \{ \mathbf{x}_k^{(n)} \}_{n=1}^{N}. \\
\]
(7)

For \( N \to \infty \) it can be shown that [6]
\[
p(\mathbf{x}_k|y_k) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}). \\
\]
(8)

That means that if a large number of random experiments according to the probability function \( p(\mathbf{x}_k|y_k) \) can be done, the number of outcomes of \( \mathbf{x}_k \) during these experiments can be used as an approximation of \( p(\mathbf{x}_k|y_k) \). Practically, this pdf is to be estimated so it cannot be sampled directly. However, if samples can be generated from a different pdf \( q(\mathbf{x}_k|\mathbf{x}_{k-1}, y_k) \), which is called *importance sampling function*, with the same support than \( p(\mathbf{x}_k|y_k) \), which means [7] that
\[
p(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0 \quad \text{for all } \mathbf{x} \in \mathbb{C}^{n_x} \\
\]
with \( n_x \) as the dimension of the state vector \( \mathbf{x} \), then a correct weighting of these particles makes an estimation of \( p(\mathbf{x}_k|y_k) \) possible. This weighting factor is called *importance weight* \( w_k^{(n)} \)
\[
w_k^{(n)} \propto \frac{p(\mathbf{x}_k|y_k)}{q(\mathbf{x}_k|\mathbf{x}_{k-1}, y_k)}. \\
\]
(9)

That means an estimation value
\[
\hat{p}(\mathbf{x}_k|y_k) = \sum_{n=1}^{N} w_k^{(n)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}) \\
\]
(10)
could be calculated. This estimation \( \hat{p}(\mathbf{x}_k|y_k) \) of the posterior probability function is equal to \( p(\mathbf{x}_k|y_k) \) if \( N \) tends to infinity [8]:
\[
\hat{p}(\mathbf{x}_k|y_k) \xrightarrow{N \to \infty} p(\mathbf{x}_k|y_k). \\
\]
(11)

The choice of a suitable importance sampling function is of great importance to obtain good performance results of
a particle filter. In this work the prior importance function
\( q(x_k^{(n)} | x_{k-1}^{(n)}, y_k) = p(x_k^{(n)} | x_{k-1}^{(n)}) \) which results in
\[
 w_k^{(n)} \propto w_{k-1}^{(n)} p(y_k | x_k^{(n)})
\] (13)
was used. By using this importance sampling function, only the two probability density functions, \( p(x_k^{(n)} | x_{k-1}^{(n)}) \) and \( p(y_k | x_k^{(n)}) \), have to be known to perform the state estimation.

4. PARTICLE FILTERS FOR EQUALIZATION

The probability \( p(x_k | x_{k-1}) \) can be derived directly from the state transition equation shown in Eqn. 1. The function \( p(x_k | x_{k-1}) \) consists of a deterministic part \( Fx_{k-1} \) and the stochastic part \( bs_k \) because the input \( s_k \) is unknown at the receiver. It is of great use that the sent symbol \( s_k \) can only be selected out of a known alphabet \( S \). This alphabet consists of two symbols for GMSK and eight symbols for 8PSK modulation. All symbols of this alphabet \( S \) are assumed to have the same probability to occur. This leads to the definition of \( p(x_k | x_{k-1}) \) shown in Eqn. 14.

\[
p(x_k | x_{k-1}) = \begin{cases} 
\frac{1}{|S|} & \text{for } x_k = Fx_{k-1} + bs_k; s_k \in S \\
0 & \text{else} 
\end{cases}
\] (14)

For practical application this means only elements out of the symbol space are used in the particle filter algorithm. Because the noise is assumed to be sampled from a normal distribution \( p(y_k | x_k) \) can be calculated the following way:

\[
p(y_k | x_k) = \frac{1}{\sqrt{\pi \sigma^2}} e^{-\frac{|y_k - y_{k-1}|^2}{2\sigma^2}}. \tag{15}
\]

Using these probability density functions an algorithm for equalization with can be formulated as shown in Tab. 1. In this algorithm new particles are drawn using the state transition equation Eqn. 1. For creating new particles only values out of the set of valid symbols, \( S_{\text{GMSK}} \) or \( S_{\text{8PSK}} \), for GMSK or 8PSK modulation, respectively, are used. After that the importance of the particles is calculated. The steps 2 to 6 are often called sequential importance sampling (SIS) in literature [7, 8]. In this algorithm an additional resampling step was used. The main principle of resampling shown schematically in Fig. 2 is the following. Particles in one iteration of the particle filtering algorithm are generated from particles of the iteration before. The assumption when using resampling is that particles with a high probability create child particles with also a high probability. For this reason resampling only duplicates particles with a high probability and suppresses particles with a low probability. The higher the probability of a particle the more often a particle is resampled. This method can dramatically improves the performance of the particle filter algorithm. It prevents the algorithm from degeneration [7, 8] and has the additional advantage that the importance weight of a particle from a prior computation step must no be taken into consideration explicitly as in Eqn. 13. To satisfy the assumption that particles with a high probability also have child particles with a high probability for equalization a prefilter [5] was applied to the channel coefficients to ensure declining coefficients. With this filter the prefiltered channel coefficients justify this assumption. In Fig. 2 the resulting effect of resampling is shown. Particles generated by the particle filter algorithm with a high probability are duplicated more often than particles with a low probability leading to the behaviour that for the next iteration particles are “moved” to areas of the state space with a high probability. That means that the number of particles used for the algorithm, clearly the dominating parameter in terms of performance and computation effort, is spent more efficiently than without resampling.

5. IMPLEMENTATION

The algorithm shown in Tab. 1 is highly parallelizable. It can be implemented in a structure as shown in Fig. 3. Most of the operations applied to a particle can be done independently of other particles. Only the resampling process cannot be implemented that easily in parallel than the other operations. This leads to a very flexible structure for realization in hardware. The SIS part of the algorithm can be done with particle filtering units (PF units) consisting of a particle memory and an operational unit of the SIS algorithm. The number of these units can be defined as a trade-off between area usage and computation speed. For hardware realization additional improvements of the algorithm were implemented. In the SIS part of the algorithm the generation of a random symbol sampled from a uniform distribution is needed. Mostly, random numbers are generated with a linear congruential generator [9]. This number
FOR every time step \( k \) do

FOR every particle index \( n \) from 1, \ldots, \( N \) do

3 Draw new particle \( x_k^{(n)} = F x_{k-1}^{(n)} + b s_k^{(n)} \)
with \( s_k^{(n)} \) uniformly sampled out of \( \mathbb{S}_{\text{PSK}} \) or \( \mathbb{S}_{\text{GMSK}} \)

4 Compute importance weights \( \tilde{w}_k^{(n)} = e^{-\frac{|y_k - h^T x_k^{(n)}|^2}{\sigma^2}} \)

5 Normalize importance weights \( w_k^{(n)} = \frac{\tilde{w}_k^{(n)}}{\sum_{m=1}^{N} \tilde{w}_k^{(m)}} \)

6 END FOR

7 Resample particles

8 Select the state \( \tilde{x}_k \) which occurs most times after resampling as most likely state

9 END FOR

10 END FOR

Table 1. Particle filter algorithm for equalization

was compared to a state of the art reduced state viterbi algorithm [10].

As an additional performance comparison the particle filter equalizer was compared to a state of the art estimation algorithm: the Kalman filter. Simulation showed that with comparable computation efforts, particle filters outperform Kalman filter algorithms for symbol estimation, as shown in Fig. 5 for example for 8PSK modulation.

As shown in in Fig. 4 and Fig. 5 the bit error rate of the particle filter algorithm is scalable by the number of particles used for estimation. For GMSK satisfying results can be achieved with about 50 particles. For 8PSK approximately 100 particles are needed to obtain good results. So it is supposable that the amount particles needed for good
estimation results also depends on the number values of a modulation symbol.

7. CONCLUSION

This work shows the application of particle filter algorithms for symbol estimation, exemplary for the GSM mobile communication system. It was shown that particle filters can outperform existing algorithms. The particle filter equalizer performs better than a commonly used reduced state viterbi algorithm, however, its complexity is much higher than the reduced state viterbi method. Interestingly, the computational effort for particle filters depends only linearly on the number of channel coefficients. That gives reason that particle filter equalizers could be a promising choice with manageable complexity for extremely broadband communication systems with long channel impulse responses.

8. REFERENCES


