A Discrete Nonlinear Filter for Fast Sampled Problems based on Vector Quantization

Mauricio G. Cea, Graham C. Goodwin and Arie Feuer

Abstract—The Chapman-Kolmogorov equation and Bayes’ rule provide a conceptually simple solution to the discrete nonlinear filtering problem. Unfortunately these equations involve high order multiple integrals which are, in general, computationally intractable. Here we exploit recent results on an incremental form of the discrete nonlinear filter to develop a novel algorithm which is computationally straightforward at high sample rates. We illustrate performance by two examples.

I. INTRODUCTION

Given a state-space model of a process, the aim of nonlinear filtering is to estimate the posterior probability density function (pdf) of the states of the system, based on all available information, including the set of received measurements [1]. The most commonly used filter is the discrete linear filter, also known as the Kalman filter [2]. The discrete nonlinear filter is also well known. It has been applied in many fields of science, see for example [3], [4], [5].

It is well known that a conceptual solution to the nonlinear filter is provided via the Chapman-Kolmogorov equation and Bayes’ rule [1]. However, this solution is, in general, computationally intractable due to the fact that it depends upon high order multiple integrals.

Different approximate solutions have been proposed for this problem. The most common of these approximations [6] are:

- (i) Local linearization, including the Extended Kalman Filter [2], [7]
- (ii) Monte Carlo, or Particle Filtering, methods based on discretization using random sampling techniques [8]
- (iii) Deterministic methods - based on deterministic discretization via vector quantization [9].

Here we will focus on algorithms of type (iii) since, as we argue below, they are particularly well suited to the use of fast sampling rates. We develop a new algorithm which utilizes ideas from vector quantization. We also present an example which illustrates, the efficacy of the proposed method.

The layout of the paper is as follows. In Section II we review the incremental form of the discrete-time nonlinear filter. In Section III we review Lloyd’s algorithm applied to the Vector Quantization problem. In Section IV we show how this scheme can be applied to the nonlinear filtering problem. In Section V we show how the algorithm can be made efficient when fast sampling rates are employed. In Section VI we present two examples to illustrate the performance of the algorithm.

II. REVIEW OF INCREMENTAL FORM OF DISCRETE NONLINEAR FILTER

The use of an incremental form to describe the discrete-time nonlinear filter is motivated by the fact that, at fast sampling rates, the change in the posteriori probability between successive samples is small. This idea is illustrated in Figure 1, for a particular example. (Details of this example will be given in Section IV.)

We summarize below the incremental form of the discrete nonlinear filter. For further details, see [10].

A. The model

The starting point is to express the discrete time Markov model in incremental form [11]. This can be obtained by rearranging the traditional model equations, leading to,

\[
\begin{align*}
    dx^+ & \triangleq x_{k+1} - x_k = f_i(x_k) \Delta + \omega_k \\
    dz^+ & \triangleq z_{k+1} - z_k = \Delta g_i(x_k) + \nu_k; z_0
\end{align*}
\]

The noise sequence \( \omega_k \) is assumed Gaussian with covariance \( E \{ \omega_k \omega_k^T \} = Q_i(x_k) \Delta \). The measured noise sequence \( \nu_k \) has covariance \( E \{ \nu_k \nu_k^T \} = R_i \Delta \). We assume \( p_{x_0}(x_0) \) is given and the sampling period is denoted by \( \Delta \).

B. The incremental form of the discrete nonlinear filter

In the sequel\(^1\), we use \( p_{x_k}(a|b) \) to denote the conditional probability density for the state at time \( k \) evaluated at ‘\( a \)’, given ‘\( b \)’. We will also use \( Z_k \) to denote \( \{ z_0, z_1, \ldots, z_k \} \).

Subject to the incremental form model (1), (2), we have, for arbitrary \( \Delta \).

1) Observation Update:

\[
p = p_{x_{k+1}|Z_{k+1}} \frac{\hat{g}(x_{k+1}) - E\{ \hat{\gamma} \}}{1 + E\{ \hat{\gamma} \}}
\]

where

\[
p \triangleq p_{x_{k+1}|Z_{k+1}} - p_{x_{k+1}}(x_{k+1}|Z_{k})
\]

\(^1\)Here, and in the sequel, we assume that \( R_i \) does not depend upon \( x_k \).

The development can be extended to cover the latter case with additional terms appearing.

\(^2\)In the equations presented below, an overtilde, i.e. \( \tilde{\cdot} \), denotes an incremental value, i.e. a term which goes to zero as the sample period goes to zero.
Here, $E\{\cdot\}$ denotes expected value using $p_{x_{k+1}}(x_{k+1}|Z_k)$, and
\[
\tilde{\gamma}(x_{k+1}) = \sum_{k=1}^{\infty} \frac{\beta(x_{k+1})^k}{k!} \quad (5)
\]
\[
\tilde{\beta}(x_{k+1}) = \frac{-1}{2\Delta} (dz^+ - h(x_{k+1})\Delta)^T R^{-1} \cdot (dz^+ - h(x_{k+1})\Delta) \quad (6)
\]
2) State Update:
\[
dp^+ = \int N_{x_k} \left\{ \hat{p} + \hat{S} + \hat{Q} + \hat{Q}\hat{S} \right\} \cdot p_{x_k}(x_k|Z_k) dx_k \quad (7)
\]
where
\[
dp^+ = p_{x_{k+1}}(x_{k+1}|Z_k) - p_{x_k}(x_{k+1}|Z_k) \quad (8)
\]
Here, $N_{x_k}$ is a gaussian distribution function with mean $x_{k+1} = f_1(x_{k+1})\Delta$ and covariance $Q_1(x_{k+1})\Delta$,
\[
\tilde{p} = p_{x_k}(x_k|Z_k) - p_{x_k}(x_{k+1}|Z_k) \quad (9)
\]
\[
\tilde{Q} = \frac{\det\{Q(x_{k+1})\}^{1/2} - \det\{Q(x_k)\}^{1/2}}{\det\{Q(x_k)\}^{1/2}} \quad (10)
\]
\[
\hat{S}(x_k, x_{k+1}) = \sum_{k=1}^{\infty} \frac{\bar{\alpha}^k}{k!} \quad (11)
\]
\[
\bar{\alpha} = -\frac{1}{2\Delta}(x_{k+1} - x_k - f(x_k)\Delta)^T Q(x_k)^{-1} \cdot (x_{k+1} - x_k - f(x_k)\Delta)
\]
\[
+ \frac{1}{2\Delta}(x_k - x_{k+1} + f(x_{k+1})\Delta)^T Q(x_{k+1})^{-1} \cdot (x_k - x_{k+1} + f(x_{k+1})\Delta) \quad (12)
\]
Proof: See [10]

Remark 1: Note equations (3) and (7) are inherently incremental since they describe the change in the posterior probability from sample to sample. A key conclusion arising from this incremental form is that the change in the posterior probability between samples is small when fast sampling is deployed. This is illustrated in Figure 1. This idea will be exploited in the sequel to develop a novel nonlinear filtering algorithm for use at fast sampling rates.

III. REVIEW OF LLOYD ALGORITHM FOR VECTOR QUANTIZATION

The core idea of the Lloyd algorithm [12], [13] is that, for a given probability density function $p(x)$, one defines a quantizer $q$.
\[
q : \mathbb{R}^k \rightarrow \{\alpha_1, \ldots, \alpha_{N_X}\}; \quad \alpha_i \in \mathbb{R}^k \quad (13)
\]
\[
q(x) = \alpha_i \quad i f x \in S_i \setminus \cup_{j<i} S_j \quad (14)
\]
The measure of the sets $S_i$, which we denote $\tilde{p}_i$, gives a discrete approximation to the probability density function:
\[
p(x) \simeq \sum_{i=1}^{N_x} \tilde{p}_i \delta(x - \alpha_i) \quad (15)
\]
where $\delta$ is the Kronecker delta function.

We denote the quantizer by the pair $(A; M_A) = \{\alpha_1, \ldots, \alpha_{N_X}; S_1, \ldots, S_{N_x}\}$.

To design a quantizer it is usual to optimize an expected distortion measure. We will utilize a quadratic distortion measure $d(x, \alpha) = \|x - \alpha\|^2$. Let $f_X$ be a density function for a $k$-dimensional random vector $X$ having finite second moment. Then, the expected distortion of the quantizer is then
\[
D(A, M) = E(d(X, q(X)))
\]
\[
= \int_{\mathbb{R}^k} d(X, q(X)) f_X(x) dx \quad (16)
\]
In the special case when $S_i = \{x \in \mathbb{R}^k | d(x, \alpha_i) \leq d(x, \alpha_j) \forall j\}$ we say [13] that $M_A = (S_1, \ldots, S_{N_x})$ is the set of Voronoi cells of $A$.

A well known result is that $D(A, M_A) \leq D(A, M)$ for all feasible $M$, see [13]. Hence the problem of optimal quantizer design is reduced to finding the code-book $A^*$ which minimizes $D(A, M_A)$. On the other hand, it is also known [13], that for a given feasible $M$ and quadratic distortion, the optimal codebook satisfies
\[
\alpha_i = E\{X|X \in S_i\} = \frac{\int_{S_i} x f_X(x) dx}{\int_{S_i} f_X(x) dx}; \quad i = 1, \ldots, N_x \quad (17)
\]
\[
\|\alpha_{i+1} - \alpha_i\| \leq \epsilon \quad \forall i \quad (19)
\]
The above condition is commonly called the centroid condition [13].

There are various associated technical issues, e.g., ensuring that $\int_{S_i} f_X(x) dx \neq 0$. We will not discuss these issues here, see [13].

A well known approach for finding an optimal vector quantizer is Lloyd’s algorithm [12], which iterates between (17), (18) until
\[
|\alpha_{i+1} - \alpha_i| \leq \epsilon \quad \forall i \quad (19)
\]
where $|\cdot|$ denotes a suitable norm and $\epsilon$ can be chosen arbitrarily small at iteration $k$.

IV. REVIEW OF APPLICATION OF LLOYD ALGORITHM TO DISCRETE NONLINEAR FILTERING

A key step in obtaining an approximate algorithm is to form a grid for the random variable $x$. One can utilize Lloyd’s algorithm to compute a suitable gridding. Several algorithms of this type are described in [9]. A related algorithm based on the use of on-line gridding is described in [14]. Our contribution in the current paper is to exploit fast sampling to obtain an efficient algorithm.
V. A DISCRETE NONLINEAR FILTER AT FAST SAMPLING

A difficulty when applying Lloyd’s algorithm to the nonlinear filtering problem is the fact that the expected distortion measure will, in general, be a nonconvex function of the quantizer [13]. However, using the ideas described in Section II, we see that the change in \( p(x_k|Z_k) \) from sample to sample is small when fast sampling rates are deployed\(^3\). We can exploit this observation to obtain an efficient algorithm. Specifically, if we have a good approximation to \( p_{x_k}(x_k|Z_k) \), then since \( p_{x_{k+1}}(x_{k+1}|Z_{k+1}) \) is close, then the quantizer for \( p_{x_k}(x_k|Z_k) \) will be a good starting point for a quantizer for \( p_{x_{k+1}}(x_{k+1}|Z_{k+1}) \).

For the sake of the explanation, we use the term sample to refer to events related to time evolution and we use the term iteration to refer to recursive computations at a given sample time.

The key idea of our algorithm is to use only one iteration to compute the new centroids of the Voronoi cells using, as starting point, the centroids at the previous sample time. We also use a quantized approximation to \( p(x_{k+1}|x_k) \) with \( N_w \) points. Note that, when the process noise covariance \( Q(x_k)\Delta \) is independent of \( x_k \), then this quantizer can be easily designed off-line. Otherwise, since the distribution is gaussian one can scale a quantizer for \( N(0,1) \) without involving excessive on-line computation. (For simplicity we will assume \( Q \) is independent of \( x \) when describing the algorithm).

In summary the algorithm is:

**Offline:**
1) Quantize \( p(x_0) \) on \( N_x \) grid points.
2) Quantize \( p(x_{k+1}|x_k) \) on \( N_w \) grid points.

**Online:**
1) Set \( k = 0 \).
2) Compute the Time Update using the Quantized state vector (centroids of the Voronoi cells) and the \( N_w \) point approximation of \( p(x_{k+1}|x_k) \). This gives a discrete approximation to \( p_{x_{k+1}}(x_{k+1}|Z_k) \) on \( N_xN_w \) points.
3) Compute the Observation Update, at the \( N_xN_w \) grid points.
4) Compute the new centroids of the existing Voronoi cells, thus requantizing to \( N_x \) grid points.
5) Update the Voronoi cells.
6) Go to step 2.

Note that steps 4 and 5 correspond to one iteration of Lloyd’s algorithm at each sample to produce an approximation to \( p_{x_{k+1}}(x_{k+1}|Z_{k+1}) \) on \( N_x \) grid points.

VI. EXAMPLES

We present two examples to illustrate different aspects of the algorithm.

**A. Example 1**

Consider the following Markovian system:

\[
\begin{align*}
 dx^+ &= x_{k+1} - x_k = -1x_k\Delta + \omega_k \\
 dz^+ &= (x_k)^2\Delta + \nu_k,
\end{align*}
\]

where \( E\{\omega_k\omega_k^T\} = 10\Delta \), \( E\{\nu_k\nu_k^T\} = 10\Delta \) and \( \Delta \) is the sampling period.

We will assume fast sampling\(^4\) with \( \Delta = 0.1 \).

In general, it is impossible to compute \( p_{x_k}(x_k|Z_k) \) exactly. However, because this is a simple example, we can obtain a very good approximation by using a fine grid. We will utilize this to give a “benchmark” against which we can compare the performance of our algorithm. In particular we use 1001 points to grid \( x \) between \(-10 \) and \(+10 \).

![Fig. 1. Time evolution of the probability density function at fast sampling, \( \Delta = 0.1 \)](image1)

![Fig. 2. Grid points corresponding to algorithm (i)](image2)

We will compare the performance of 3 algorithms.

(i) We evaluate the “true” evolution of \( p_{x_k}(x_k|Z_k) \). Then, so as to form a comparison with the other algorithms, we produce a quantized form of this “true” distribution at each sample by running many iterations of Lloyd’s algorithm. (This result will be used as a benchmark against which to compare the other algorithms.)

\(^3\)Actually, we will see in example 1 presented in section VI that “fast sampling” can be consistent with sampling rates used in practice.

\(^4\)Note that \( \Delta = 0.1 \) could be considered a moderate sampling rate. However, it is \( 1/10^{th} \) the response time of the system and this is sufficiently “fast” in the context of this example.
that any algorithm of the Extended Kalman Filter type for estimating the posterior mean will be incapable of crossing “various” boundaries in the state space, i.e. it will “lock up”.

The system in continuous time is:

\[
dx &= [x(t) - x(t)^3]dt + d\omega \tag{22}
\]
\[
dz &= [x(t)^2 - 0.5x(t)]dt + d\nu \tag{23}
\]

where \(d\omega\) and \(d\nu\) have incremental covariance \(Qdt = Rd\nu = 0.1dt\).

We consider a fast sampled version of this system, i.e.:

\[
dx^+ &= [x_k - x_k^3]\Delta + \omega_k \tag{24}
\]
\[
dz^+ &= [x_k^2 - 0.5x_k]\Delta + \nu_k \tag{25}
\]

where \(\omega\) and \(\nu\) have incremental covariance \(Q\Delta = R\Delta = 0.1\Delta\).

(ii) We run the algorithm of Section V. i.e with 1 iteration of the Lloyd algorithm per sample.

(iii) We run the algorithm of section V save we iterate the Lloyd algorithm at each sample until convergence occurs.

So as to provide a common basis for comparing the three algorithms, we will plot the resulting grid points at each sample time. The results are shown in Figures 2, 3, 4 respectively for algorithms (i), (ii), (iii). Comparison of Figures 2 and 4 shows that the multi-step Lloyd algorithm gives excellent results for this example. Comparison of Figures 3 and 4 shows some deterioration when only one iteration of the Lloyd algorithm is used per sample. However, key features of the posterior distribution are captured. This is further illustrated in Figure 5 which shows the cumulative probability distribution i.e. \(\int_{-\infty}^{x_k} p_{x_k}(x_k|z_k)dx_k\) for the three algorithms at sample 10 together with the cumulative distribution corresponding to the true posteriori density (calculated with a fine grid). We see that the new algorithm gives a very good approximation to the probability distribution for this example.

B. Example 2

This problem was suggested in [15]. The system has several inherit difficulties. For example, it is argued in [15]...
We compare algorithms (i) and (ii) as discussed in relation to example 1. Figures 7, 8, 9, 10 show the cumulative distribution at \( k = 1, 100, 1000, 6000 \) respectively. For clarity we show only the results for the “true” algorithm and the algorithm suggested in Section V. We see from these figures that the algorithm of Section V tracks the evolution of the “true” distribution albeit with noticeable errors at certain times.

VII. CONCLUSIONS

We have developed a novel discrete nonlinear filter based on vector quantization via Lloyd’s algorithm. The key observation is that, when fast sampling is used, the posterior probability changes slowly between samples. This observation is exploited in our algorithm by running only one iteration of the Lloyd’s algorithm at each sample. Two examples have been used to illustrated the performance of the algorithm. It has been shown that the proposed algorithm works well, although the second example is particularly challenging and warrants further study.

REFERENCES