we must also have

$$s_* = \operatorname*{argmin}_{s[n]} \varepsilon[N; s[0], s[1], \cdots, s[N-1], x].$$

Hence, it remains only to show

$$s_* = \operatorname*{argmin}_{s[n]} \varepsilon[n+1; s[0], s[1], \cdots, s[n], x] = \operatorname{sgn} \sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{s[n-1],\cdots,s[k]}(y[k]) \quad (42)$$

to initiate the induction.

But since

$$\varepsilon[n+1; s[0], s[1], \cdots, s[n], x] = \varepsilon[n; s[0], s[1], \cdots, s[n-1], F^{-1}_{s[n]}(x)] + (y[n+1] - x)^2$$

we have

$$s_* = \operatorname*{argmin}_{s[n]} \varepsilon[n+1; s[0], s[1], \ldots, s[n], x] = \operatorname*{argmin}_{s[n]} \sum_{k=0}^{n} \beta^{2(k-n)} (F^{(n-k)}_{s[n-1],\cdots,s[k]}(y[k]) - F^{-1}_{s[n]}(x))^2 \quad (43)$$

Expanding the quadratic terms in (43) and noting from (6) that, for $x \in (-1, \beta - 1)$

$$|F^{-1}_{s[n]}(x)| = \frac{\beta - 1 - x}{\beta} \quad (44)$$

is independent of $s[n]$, we get

$$s_* = \operatorname*{argmax}_{s[n]} \left\{ F^{-1}_{s[n]}(x) \sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{s[n-1],\cdots,s[k]}(y[k]) \right\}. \quad (45)$$

In turn, since

$$s[n] = \operatorname{sgn} F^{-1}_{s[n]}(x)$$

for $x \in (-1, \beta - 1)$ in accordance with (6), (45) implies s_* must be given by the right-hand side of (42).

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewer for several helpful comments and suggestions, including those on connections to the extended Kalman filter.

REFERENCES

[1] M. D. Richard, "Probabilistic state estimation with discrete-time chaotic systems," RLE Tech. Rep. 571, MIT, Cambridge, MA, Mar. 1992.

[2] C. Myers, A. Singer, B. Shin, and E. Church, "Modeling chaotic systems with hidden Markov models," in *Proc. Int. Conf. on Acoustics Speech, and Signal Processing*, 1992.

[3] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, "State space reconstruction in the presence of noise," *Physica D*, vol. 51, pp. 52–98, 1991.

[4] A. Lasota and M. C. Mackey, *Probabilistic Properties of Deterministic Systems*. Cambridge, UK: Cambridge Univ. Press, 1985.

[5] A. V. Holden, Ed., *Chaos*. Princeton, NJ: Princeton Univ. Press, 1986.

[6] C. Myers, S. Kay, and M. Richard, "Signal separation for nonlinear dynamical systems," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1992.

[7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering* Englewood Cliffs, NJ: Prentice-Hall, 1979.

Remarks on Linear and Nonlinear Filtering

Bernard Delyon

***Abstract*— This communication tries to give some insight into relationships existing between Viterbi and the Forward–Backward algorithm (used in the context of Hidden Markov Models) on one hand and Kalman filtering and Rauch–Tung–Striebel smoothing on the other. We give a unifying view which shows how those algorithms are related and give an example of a nonlinear hybrid system that can be filtered through a mixed algorithm.**

***Index Terms*—Nonlinear filtering, Viterbi algorithm.**

I. INTRODUCTION

In this communication, we consider estimation of the state of semi-Markov processes. These processes arise in two quite different fields: Hidden Markov Models (widely used for speech recognition, [1]) and Kalman–Bucy filtering; in the first case, the state-space is discrete (generally finite) while in the second one it is the Euclidean space. However, algorithms which are used have considerable similarities. Inspection of these similarities will lead us first to a generalization of Kalman–Bucy filtering in a particular extension to nonlinear systems and secondly to extend this model to a state-space which is mixed continuous-discrete.

A. Model

Semi-Markov processes (Y_n) are defined through their state-space representation (X_n, Y_n) (in some measurable space $\mathcal{X} \times \mathcal{Y}$) where Y is the observation and X is the hidden state; (X_n, Y_n) is a Markov chain and the assumption is that the transition from (X_n, Y_n) to (X_{n+1}, Y_{n+1}) does not depend on Y_n, which implies that (X_n) is itself a Markov chain. This process is thus characterized by its transition function $\Pi(x, x', y')$.

- In the case of discrete spaces
$$\Pi(x, x', y') = P(X_{n+1} = x', Y_{n+1} = y' | X_n = x)$$
- for continuous spaces
$$\Pi(x, x', y')dx'\,dy' = P(X_{n+1} \in dx', Y_{n+1} \in dy' | X_n = x)$$
and analogous formulas for mixed discrete/continuous spaces. An initial distribution is also given for X_0. The assumption imply that (Y_n) "depends" only on (X_{n-1}, X_n) in the sense that
$$P(Y_n | X_0, \cdots, X_N) = P(Y_n | X_{n-1}, X_n) = \Pi(X_{n-1}, X_n, Y_n)/p(X_{n-1}, X_n)$$
where $p(x, x')$ is the transition function of the chain (X_n)
$$p(x, x') = \int \Pi(x, x', y')\, dy'.$$
Thus the distribution of the process may also be realized by first running the Markov chain (X_n) with its own law and then drawing the random variables (Y_n) with their distribution conditioned on (X_{n-1}, X_n).

Manuscript received May 14, 1993; revised February 9, 1994.

The author is with IRISA-INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France.

IEEE Log Number 9407107.

B. Examples

The linear-Gaussian model has the following representation:

$$X_n = FX_{n-1} + w_{n-1}, \qquad w_n \simeq \mathcal{N}(0, Q)$$
$$Y_n = HX_n + v_n, \qquad v_n \simeq \mathcal{N}(0, R) \tag{1}$$

(for the sake of simplicity, we assume that matrices F, H, Q, R are not time-dependent and v and w are two independent sequences of independent variables). We have (see bottom of page)

In the case of nonlinear filtering with discrete state space (Hidden Markov Models), transition probabilities are generally presented in the form (cf [1])

$$\Pi(i, j, y) = a_{ij}b_{ij}(y)$$

where

$$a_{ij} = \int \Pi(i, j, y)\, dy$$

is the transition matrix of the chain (X_n) and b_{ij} is the distribution of Y_n conditional to the transition from i to j.

C. Problems

Two problems are traditionally addressed:

- Smoothing: what can be said of the sequence $X_0, X_1, \cdots, X_N$ once we are given an observation set $Y_1, Y_2, \cdots, Y_N$?
- Filtering: how to estimate recursively X_q from the observation of Y_q and the previous estimate $\hat{X}_{q-1}$?

In any case, the problem is strongly connected to the maximization over $X_0, X_2, \cdots, X_n$ of the log-likelihood functional

$$\mathcal{L}(X_0, X_1, \cdots, X_n) = f(X_0, X_1, Y_1) + f(X_1, X_2, Y_2) + \cdots + f(X_{n-1}, X_n, Y_n). \tag{2}$$

The usual filtering algorithms when the model is linear-Gaussian (Kalman–Bucy, Rauch–Tung–Striebel) or when the state-space is finite (Viterbi) consist only in fast *exact* maximization of such a functional by taking maximum advantage of this particular form. The most popular strategy, for nonlinear systems in continuous state-space, is to use the Extended Kalman Filter, which consist of some kind of linearization of the above functional; this procedure is not guaranteed to lead to a stable algorithm. Another possible choice would be to search for an approximate solution of this equation by fast numerical methods (which are frequently very efficient); this last aspect does not seem to have been really explored. In this communication, we will try to extend as far as possible the *exact* maximization procedures to some nonlinear models.

In Section II, we will show how the solutions for Viterbi and Kalman–Bucy are related; in Section III, we study the case of nonlinear filtering in continuous state-space; in Section IV we give an example where the state-space is a mixed discrete and continuous space.

II. Forward–Backward and Viterbi Algorithms

We compare here two algorithms: the first estimates the present state X_n by maximizing its probability conditionally to the observations, while the second maximizes the probability of the whole trajectory $X_0, X_1, \cdots, X_n$ conditionally to the observations. In the case of filtering, the observations we consider for the estimation X_n are $Y_1, Y_2, \cdots, Y_n$ while in the case of smoothing it is the whole observation set $(Y_1, Y_2, \cdots, Y_N)$.

A. Forward–Backward Algorithm

This algorithm is designed for recursive estimation of the probability for the current state X_n to be x once we have observed $Y_1, \cdots, Y_n$, that is, $P(Y_n = x|Y^n)$ (we put $Y^n = (Y_1, Y_2, \cdots, Y_n)$); it is actually simpler to calculate the unnormalized probability $\alpha_n(x) = P(X_n = x, Y^n)$. Using Markov property and Bayes formula we obtain

$$\begin{aligned}\alpha_n(x) &= \sum_u P(X_n = x, Y_n, X_{n-1} = u, Y^{n-1}) \\ &= \sum_u P(X_n = x, Y_n|X_{n-1} = u, Y^{n-1})\alpha_{n-1}(u) \\ &= \sum_u \Pi(u, x, Y_n)\alpha_{n-1}(u).\end{aligned} \tag{3}$$

Using this formula we can estimate recursively, at each time n, $\alpha_n(x)$ for all values of x.

In the same way, in continuous state-space, we obtain for the unnormalized probability density the equation

$$\alpha_n(x) = \int \alpha_{n-1}(u)\Pi(u, x, Y_n)\, du. \tag{4}$$

In the case of Kalman–Bucy filtering, $\alpha_n(x)$ is a Gaussian density and the last formula leads directly to Kalman–Bucy filter equation, expressing the reestimation of mean and variance. In the same way, after time N, we can compute recursively the backward variables

$$\beta_n(x) = P(Y_{n+1}^N|X_n = x)$$

with

$$\beta_n(x) \sum_u \cdot \beta_{n+1}(u)\Pi(x, u, Y_{n+1}).$$

The estimated filtered state at time n will be

$$\hat{x}_n = \arg\max \alpha_n(x).$$

The Markov property (independence of the past and future conditioned on the present) implies that $P(X_n = x, Y^N) = \alpha_n(x)\beta_n(x)$ and the estimated smoothed state at time n will be

$$x_n^{\#} = \arg\max \alpha_n(x)\beta_n(x).$$

In the Gaussian case, this algorithm coincides with the two-filter or Mayne–Fraser smoother ($x_n^{\#}$ is the weighted mean of $E[X_n|Y^n]$ and $E[X_n|Y_{n+1}^N]$; cf [3], eq. (5.1–12)]); this comes simply from the fact that (4) implies that α_n has the form

$$\alpha_n(x) = c_n \exp\{-(x - E[X_n|Y^n])^T R_n^{-1}(x - E[X_n|Y^n])/2\}$$

and a similar expression holds for β_n.

$$\Pi(u, x, y) = \frac{\exp\{-(x - Fu)^T Q^{-1}(x - Fu)/2 - (y - Hx)^T R^{-1}(y - Hx)/2\}}{\det(2\pi Q)^{1/2} \det(2\pi R)^{1/2}}.$$

B. Viterbi Algorithm

The Viterbi algorithm is designed for recursive estimation, for each state x_n, of the most likely path ending at this state, say $\mathcal{C}(x_n) = (x_0(x_n), \cdots x_{n-1}(x_n), x_n)$. Like before we have

$$\begin{aligned}\phi_n(x_n) &\triangleq P(\mathcal{C}(x_n), Y^n) \\ &= \sup_{x_0,\cdots,x_{n-1}} P(x_0, \cdots, x_n, Y^n) \\ &= \sup_{x_0,\cdots,x_{n-1}} P(Y_n, x_n | x_0, \cdots, x_{n-1}, Y^{n-1}) \\ &\quad \cdot P(x_0, \cdots, x_{n-1}, Y^{n-1}) \\ &= \sup_{x_0,\cdots,x_{n-1}} P(Y_n, x_n | x_{n-1}) P(x_0, \cdots, x_{n-1}, Y^{n-1}) \\ &= \sup_{x_{n-1}} \Pi(x_{n-1}, x_n, Y_n)\phi_{n-1}(x_{n-1}). \end{aligned} \tag{5}$$

At the same time we memorize the function

$$\xi_{n-1}(x_n) = \arg \sup_{x_{n-1}} \Pi(x_{n-1}, x_n, Y_n)\phi_{n-1}(x_{n-1}). \tag{6}$$

When the state-space is discrete, this function, is a state pointer which represents the value of x_{n-1} on the most likely path $\mathcal{C}(x_n) = (x_0, \cdots, x_{n-1}, x_n)$ when $(x_n, Y_1, \cdots, Y_n)$ are given; in the case of Gaussian linear smoothing it will be a linear function (see below). The filtered estimate at time n knowing Y^n is

$$\check{x}_n = \arg\max_x \phi_n(x). \tag{7}$$

Smoothed estimates over the interval $[0, N]$ are given by the equations

$$\begin{aligned} x_N^* &= \check{x}_N \\ x_{n-1}^* &= \xi_{n-1}(x_n^*). \end{aligned} \tag{8}$$

Those equations are well known in dynamic programming (cf [10]).

C. Comments

The forward-filtered estimate $\hat{x}_n$ maximizes the *a posteriori* probability $P(X_n = x|Y^n)$ with respect to x, while Viterbi-filtered estimate $\check{x}_n$ maximizes $\phi_n(x)$ which is the *a posteriori* probability of the whole path; those two estimates are generally different. However, as we shall see in next section, they are identical in the case of Gaussian linear filtering.

In the following sections we consider smoothing only under Viterbi aspects for the following reason: nothing guarantees that the sequence $x_n^{\#} = \arg\max_x P(X_n = x|Y^N)$, $n = 0, \cdots, N$, has a nonzero probability for the Markov chain; this sequence is fundamentally different from (x_n^*) (for instance, in speech recognition, the identified sequence of phonemes has to be globally meaningful (constitute a word), which means that it is necessary to obtain a sequence with nonzero probability). However, in the case of Gaussian linear smoothing, those two sequences are identical.

III. Fast Filtering in Continuous State-Space

A. Result

In the case of continuous state-spaces, the problem in the application of previous formulas is that we have to memorize the functions ϕ_n and ξ_n, which is impossible unless those functions are parametrized by a vector, say $\theta \in \boldsymbol{R}^d$. This is what happens in Kalman–Bucy (filtering) and Rauch–Tung–Striebel (smoothing) algorithms where these functions are Gaussian densities.

We explore here a more general setting where ϕ_n and ξ_n can still be parametrized. We study only the stationary case (i.e., corresponding in the Kalman–Bucy context to the case where the variance of the first state is such that no matrix has to be reestimated during the algorithm). The assumption on the model is constituted by a constraint on the form of the transition and observation probability.

We assume that the transition and observation probability may be expressed as

$$\begin{aligned}\log P(X_n = x', Y_n = y'|X_{n-1} = x) &= -U(x - Ax') - V(x') \\ &\quad + V(x) + \theta(y')^T x' - Z(y') \end{aligned} \tag{9}$$

where U and V are convex functions, θ and Z are arbitrary functions, and A is a matrix with all eigenvalues inside the unit circle.

Theorem 1: We assume that the *a priori* probability for X_0 is proportional to $\exp(-V(x_0) + \theta_0^T x_0)$; in that case, functions ϕ_n and ξ_n may be parametrized with a sequence θ_n and the filtering and smoothing equations are

$$\theta_n = A^T\theta_{n-1} + \theta(Y_n) \tag{10}$$

$$\xi_n(x) = Ax + \nabla g(\theta_n) \tag{11}$$

$$\check{x}_n = \nabla h(\theta_n) \tag{12}$$

where g and h are Legendre transforms of functions U and V

$$g(\theta) = \sup_x \theta^T x - U(x) \tag{13}$$

$$h(\theta) = \sup_x \theta^T x - V(x). \tag{14}$$

Equations (8) are used for smoothing.

Proof: The proof is elementary if one sues (5); it consists in verifying by induction that

$$\log(\phi_n(x)) = \theta_n^T x - V(x) + \sum_{i=1}^{n} g(\theta_{i-1}) - Z(Y_i).$$

Equations (12) and (11) come from (7) and (6) and from the fact that the x which realizes the supremum of (13) (resp. (14)) is $\nabla g(\theta)$ (resp., $\nabla h(\theta)$). ■

Comments: • The function U is not necessarily finite; for instance, in two dimensions, one can have $U(x) = +\infty$ if the second coordinate of x is nonzero; if the matrix A has a second row $(1, 0)$, this will mean that the second coordinate of X_n coincides with the first one of X_{n+1}; one can easily verify that for such a function U, $g(x)$ does not depend on the second coordinate of x.

• The corresponding with Kalman–Bucy filtering and Rauch–Tung–Strie-bel smoothing (with the notation of (1) and of [3]) is

$$\begin{aligned} A &= P_+ F^T P_-^{-1} \\ U(x) &= x^T(F^T Q^{-1} F + P_+^{-1})x/2 \\ V(x) &= x^T P_+^{-1} x/2 \\ \theta(y) &= H^T R^{-1} y \\ Z(y) &= y^T R^{-1} y/2 + \text{const} \\ \theta_n &= P_+^{-1}\hat{x}_n^+ \\ A^T\theta_n &= P_-^{-1}\hat{x}_{n+1}^- \end{aligned} \tag{15}$$

where P_+ and P_- are variance of prediction errors of

$$\hat{x}_n^+ = E[x_n|Y^n] = \check{x}_n \quad \text{and} \quad \hat{x}_n^- = E[x_n|Y^{n-1}] = F\check{x}_{n-1}.$$

They are solution of the system

$$P_- = FP_+F^T + Q$$
$$P_+^{-1} = P_-^{-1} + H^T R^{-1} H \tag{16}$$

(cf [3], tables 4.2.1 and 5.2.2]). Checking this correspondence is standard matrix algebra, starting with the identification of (9) with

$$-(x' - Fx)^T Q^{-1}(x' - Fx)/2 \\ -(y' - Hx')^T R^{-1}(y' - Hx')/2 + \text{const}$$

and using (16).

- One has

$$\theta_n = \nabla U(x_n^* - Ax_{n+1}^*)$$

where (x_n^*) is the optimal sequence. This can be checked instantaneously by differentiation with respect to x_n of the global likelihood of the sequence:

$$\log P(x_1, \cdots, z_N, y_1, \cdots, y_N) \\ = V(x_0) - V(x_N) \\ - U(x_0 - Ax_1) \cdots - U(x_{N-1} - Ax_N) \\ + \theta(y_1)^T x_1, \cdots, +\theta(y_N)^T x_N - Z(y_1), \cdots, -Z(y_N) \tag{17}$$

where the three terms appear after differentiation led to (10).

B. Examples

Starting from a Linear Model: Replacement, in (15), of $U(x)$ by $f(U(x))$ for some function f leads to new models. For instance, if $f(x) = x^\alpha$, $1/2 < \alpha$, we obtain, after a simple calculation

$$\nabla g(\theta) = (\theta^T M\theta/2)^{(1-\alpha)/(2\alpha-1)} M\theta$$

with

$$M = (F^T Q^{-1} F + P_+^{-1})^{-1}.$$

This transformation may be interpreted as a spreading of the noise distribution if $\alpha < 1$, and a peaking if $\alpha > 1$; note that only the backward smoothing equation is affected by this modification.

A solution for assuming the noise bounded (in the sense that $|U(x - Ax')| \leq \alpha$) would be to choose, for example, $f(x) = x$ if $|x| < \alpha$ and $+\infty$ elsewhere. In this case

$$\nabla g(\theta) = \min\left(\frac{\alpha}{|M\theta|}, 1\right) M\theta.$$

Another approach: It is not easy to design functions U and V such that (9) represents a transition probability; if (A, U, θ, Z) is given, $\exp(-V)$ should be an eigenvector of the operator

$$f \to Tf(x) = \int f(x') \exp\{-U(x - Ax') \\ +\theta(y')^T x' - Z(y')\}\, dx'\, dy'$$

since the operator is positive, the existence of V is equivalent to the existence of a nonnegative function W such that $TW < cW$ for some constant c.

On the other hand, note that for smoothing purposes, an approximation of V may be sufficient because this function is utilized only for obtaining the filtered estimate of the state at the last time N. In other words, a replacement of V by $\tilde{V}$ has the same effect as replacing the likelihood of the sequence $(x_0, \cdots, x_N)$ by

$$P(x_0, \cdots, x_N) e^{V(x_N) - \tilde{V}(x_N)}$$

(cf (17)) which has a small influence on the estimates except close to the end.

It is thus reasonable to define the model by giving the functions (A, U, θ, Z), and calculate V later. This has an interesting probabilistic interpretation since the probability given the past and the future is

$$\log P(x', y' | x, x'') \simeq -U(x - Ax') \\ -U(x' - Ax'') + \theta(y')^T x' - Z(y'). \tag{18}$$

The Markov chain is thus represented as a reciprocal process (cf [4]). In the linear case, the theory of reciprocal processes has been worked out completely in [5], [6]; the nonlinear case, which is much more complex, is studied in [7], [8]. It is unfortunately very difficult to find a form similar to (1) (i.e., $X_{n+1} = f(X_n, w_{n+1})$) for such a process; on the other hand, (18) contains the only terms of global likelihood involving x' and y'; thus a direct minimization over x' and y' leads to the equations satisfied by the most likely (noiseless) trajectories: for three successive points (x, x', x'')

$$A^T \nabla U(x - Ax') - \nabla U(x' - Ax'') + \theta(y') = 0 \tag{19}$$

$$\nabla\theta(y')^T x' - \nabla Z(y') = 0 \tag{20}$$

and (18) can be considered as the distribution of a noisy version of a process satisfying (19) and (20).

We now particularize and explore the kind of dynamical systems which may be represented by those equations. If Z is convex with Legendre transform k and θ has the form $\theta(y) = H^T y$ for same matrix H, (20) leads of $y' = \nabla k(Hx')$ and (19) becomes

$$A^T \nabla U(x - Ax') - \nabla U(x' - Ax'') + H^T \nabla k(Hx') = 0.$$

If we set $\theta_n = \nabla U(x_n + Ax_{n+1})$, this equation can be rewritten as

$$\theta_n = A^T \theta_{n-1} + H^T y_n$$
$$x_n = Ax_{n+1} + \nabla g(\theta_n)$$
$$y_n = \nabla k(Hx_n)$$

(remember that $\nabla g(\nabla U(x)) = x$) which is actually the noiseless version of (10). We do not know any simple way of solving this system in discrete time; we will interpret it as the discretization of the continuous-time system

$$\dot{\theta} = B^T \theta + H^T y \tag{21}$$
$$\dot{x} = -Bx - \nabla g(\theta) \tag{22}$$
$$y = \nabla k(Hx) \tag{23}$$

where

$$A = \exp(\epsilon B) \tag{24}$$

and ϵ is the sampling period. Since A has all its eigenvalues inside the unit circle, B is stable. This system is explosive if we start with arbitrary initial conditions; searching for stable solutions of the form $x = \phi(\theta)$, we obtain the equation for ϕ

$$g(\theta) = -\theta^T B\phi(\theta) - k(H\phi(\theta)) \tag{25}$$

and the noiseless state x and observation y satisfy

$$\dot{\theta} = B^T \theta + H^T y$$
$$x = \phi(\theta)$$
$$y = \nabla k(Hz).$$

This is the model (B, H, k, ϕ) of the noiseless system; the conditions on the model are that B is stable, k is convex, and the function g given by (25) is convex. Filtering is done by feeding (21) with the observed process y_t and smoothing is performed by running (22) backwards in time with the trajectory θ_t obtained in the filtering stage. A natural initial value of x_T is $\phi(\theta_T)$.

We particularize again further. For instance, performing the usual trick for translating high-order systems into systems of order one, we can consider examples of the form

$$\begin{aligned} \theta &= (z, \dot{z}, \cdots, z^{(p-1)}) \\ B^T_{ij} &= \delta_{i+1,j}, \qquad i < p \\ B^T_{pj} &= -b_j \\ H &= (0, 0, \cdots, 0, 1) \\ \phi(\theta) &= (b_2, b_3, \cdots, b_p, 1)^T \theta_1 \end{aligned}$$

corresponding to the one-dimensional representation of the noiseless process

$$\text{noiseless system} \begin{cases} z^{(p)} + \sum_{j=0}^{p-1} b_{j+1} z^{(j)} = \lambda(z) \\ x = (b_2, b_3, \cdots, b_p, 1)^T z \\ y = \lambda(z) \end{cases} \begin{matrix} \\ \} \\ \} \end{matrix} \begin{matrix} \text{state} \\ \text{observation} \end{matrix}$$

where $\lambda(z) = k'(z)$. The condition on B is that the roots of

$$\sum b_j z^j$$

have a negative real part; because of the special form of ϕ, function g can be rewritten as

$$g(\theta) = b_1 \theta_1^2 - k(\theta_1). \tag{26}$$

We have finally the conditions (convexity of g and k)

$$\text{assumptions} \begin{cases} \text{the roots of } \sum b_j z^j \text{ have negative real part} \\ 0 \le \lambda'(z) \le 2b_1. \end{cases}$$

The continuous-time filtering and smoothing equations become

$$\begin{matrix} \text{filtering,} \\ \text{smoothing} \end{matrix} \begin{cases} \dot{\theta}_1 = B^t \theta_t + H^T y_t & \text{forward} \\ \dot{x}_t = B x_t - (2b_1 \theta_{t1} - \lambda(\theta_{t1})) H^T & \text{backward} \end{cases}$$

the first equation is solved on $[0, T]$ and the second one, backwards in time with the initial condition of $x(T) = (b_2, b_3, \cdots, b_p, 1)^T \theta_1(T)$.

If for instance, a function λ such as

$$\lambda(z) = \begin{cases} \alpha z & |z| \le Z_0, \\ \alpha Z_0 & |z| > 0, \end{cases} \qquad 0 \le \alpha \le 2b_1,\ Z_0 > 0$$

satisfies the assumptions. In those cases, the evolution equation of the model with $p = 1$ has two stable points with opposite values and 0 is unstable; this is a major difference with what happens with linear systems. The amount of noise assumed in the noisy system is difficult to evaluate; it is clearly related to invariant transformations of the noiseless system such as $\lambda(z) \to \lambda(z) + cz$ and $b_1 \to b_1 + c$ which do not change the noiseless system while they modify the noise (we could also have put $\phi(\theta) = (b_2, b_3, \cdots, b_p, 1)^T \psi(\theta_1)$ for some increasing function ψ without changing the system).

Thus the noise in the stochastic system is determined by the particular form of equations chosen for describing the noiseless one (i.e., (19 and 20)).

Obviously many questions remain open, particularly the problem of the nature of the noise (some interesting insights may be found in [6]).

IV. A Hybrid Model

This section extends the idea of Viterbi and Kalman filtering to a mixed continuous and discrete state (x, e), $e \in \{1, 2, \cdots, n\}$.

A similar situation is considered in [9, p. 182] where e_n is a finite-state Markov chain and x_n is a linear process whose matrices F, H, Q, R (cf. (1)) depend on e_n; as explained [9, p. 186], filtering equations lead to an infinite set of equations to be solved at each instant (this is basically due to the fact that x_n is not Gaussian any more, so that the propagation of its conditional mean implies the propagation of its whole conditional distribution). We will explore this case in the following looking this time at $\check{x}$ and x^*.

Theorem 2: We assume that the transition and observation probability may be expressed as

$$\begin{aligned} \log P(x', e', y'|x, e) = &-U_{ee'}(x - Ax') - V_{e'}(x') \\ &+ V_3(x) + \theta(y')^T x' - Z_{ee'}(y') \end{aligned} \tag{27}$$

where the functions $U_{ee'}$ and V_e are convex and the functions θ_e are arbitrary. We assume that the *a priori* probability of (x_0, e_0) is proportional to $p_0(e_0) \exp(-V(x_0) + \theta_0^T x_0)$; filtering and smoothing may be performing by estimating the continuous and discrete states through the equations

$$\eta_0 = \log(p_0(e)) \tag{28}$$

$$\theta_n = A^T \theta_{n-1} + \theta(Y_n) \tag{29}$$

$$\eta_n(e) = \eta_{n-1}(\epsilon_{n-1}(e)) + g_{\epsilon_{n-1}(e)e}(\theta_{n-1}) - Z_{\epsilon_{n-1}(e)e}(Y_n) \tag{30}$$

$$\epsilon_n(e) = \arg\max_{e_n} \eta_n(e_n) + g_{e_n, e}(\theta_n) - Z_{e_n, e}(Y_{n+1}) \tag{31}$$

$$\xi_n(x, e) = Ax + \nabla g_{\epsilon_n(e)e}(\theta_n) \tag{32}$$

$$\check{e}_n = \arg\max \{\eta_n(e) + h_e(\theta_n)\} \tag{33}$$

$$\check{x}_n = \nabla h_{\check{e}_n}(\theta_n) \tag{34}$$

where $g_{ee'}$ and h_e are Legendre transforms of functions $U_{ee'}$ and V_e. $\epsilon_n(e)$ is the most likely discrete state at time n knowing $e_{n+1} = e$ (it is independent of x_{n+1}).

Proof: By using (5) we shall show by induction on n that

$$\log \phi_n(x_n, e_n) = \eta_n(e_n) + \theta_n^T x_n - V_{e_n}(x_n) \tag{35}$$

where

$$\eta_n(e_n) = \max_{e_0, \cdots, e_{n-1}} \log(p_0(e_0)) + \sum_{i=1}^{n} g_{e_{i-1}e_i}(\theta_{i-1}) - Z_{e_{i-1}e_i}(Y_i).$$

Note that the above formula for the function η_n corresponds to (28) and (30). Using (5), we check (35) at time $n + 1$

$$\begin{aligned} \log \phi_{n+1}(x, e) &= \max_{x_n e_n} \{\log \phi_n(x_n, e_n) \\ &\quad - U_{e_n e}(x_n - Ax) - V_e(x) + V_{e_n}(x_n) \\ &\quad + \theta(Y_{n+1})^T x - Z_{e_n e}(Y_{n+1})\} \\ &= max_{x_n, e_n} \{\eta_n(e_n) + \pi_n^T x_n - U_{e_n e}(x_n - Ax) \\ &\quad - V_e(x) + \theta(Y_{n+1})^T x - Z_{e_n e'}(Y_{n+1})\}. \end{aligned}$$

Taking the supremum over x_n, we obtain

$$x_n = Ax + \nabla g_{e_n e}(\theta_n) \tag{36}$$

and

$$\begin{aligned}\log \phi_{n+1}(x, e) &= \max_{e_n}\{\eta_n(e_n) + \theta_n^T Ax + g_{e_n e}(\theta_n) \\ &\quad - V_e(x) + \theta(Y_{n+1})^T x - Z_{e_n e}(Y_{n+1})\} \\ &= \max_{e_n}\{\eta_n(e_n) + g_{e_n e}(\theta_n) - V_e(x) \\ &\quad + \theta_{n+1}^T x - Z_{e_n e}(Y_{n+1})\}.\end{aligned}$$

The $e_n = \epsilon(e)$ which realizes the optimum leads to (31), and then (36) leads to (32).

Taking the supremum over x_n, in (35), we obtain $\hat{x}_n = \nabla h_{e_n}(\theta_n)$ and the probability of the best path arriving at e_n at time n is

$$\log \phi_n(e_n) = \eta_n(e_n) + h_{e_n}(\theta_n)$$

which leads to (33). ■

Comments: • The storage requirements of this algorithm are still reduced: for smoothing it will be N state pointers (as in Viterbi algorithm) and N vectors.

• We can model a signal whose law is a mixture of linear models: each element of the mixture will be indexed by a pair (e, e') and $U_{ee'}(u)$ will be (with some abuse of notation)

$$U_{ee'}(x) = -\log(p(e, e')) + x^T U_{ee'} x$$

where $p(e, e')$ is the probability of the transition from e to e'. As before, functions $V_e(x)$ will be quadratic forms $x^T V_e x$ and $\theta(y)$ is a matrix product Θy. If we drop indices in the (e, e')−dependent linear model $(F_{ee'}, Q_{ee'}, H_{ee'}, R_{ee'})$, we have to identify the likelihood (27) to

$$-(x' - Fx)^T Q^{-1}\Big((x' - Fx)/2 - (y' - Hx')^T R^{-1}(y' - Hx')/2$$

and this leads to

$$H^T R^{-1} H + Q^{-1}\Big) = 2A^T U_{ee'} A + 2V_{e'}$$

$$F^T Q^{-1} F = 2U_{ee'} - 2V_e$$

$$F^T Q^{-1} + Q^{-1} F = U_{ee'} A + A^T U_{ee'}$$

$$R^{-1} = \Theta.$$

With those notation the process may be described in the following way: starting with a discrete state (e, x), the process jumps to another one e' with probability $p(e, e')$, and a new state x' is chosen with the dynamics of $(F_{ee'}, Q_{ee'})$ and an observation y' is then produced with x' and $(H_{ee'}, R_{ee'})$ (cf (1)). Setting $e = e'$ in the equations above (steady state), we see the principal restriction of this mode: the matrices $R^{-1}H(= \Theta)$ and $P_+ F^T P_-^{-1}(= A)$ are independent of e.

References

[1] L. R. Rabiner and B. H. Juang, "Introduction to hidden Markov models," *IEEE ASSP Mag.*, Jan. 1986.

[2] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 2, Mar. 1974.

[3] A. Gelb, *Applied Optimal Estimation.* Boston, MA: M.I.T. Press, 1989.

[4] B. Jamison, "Reciprocal processes," *Zeit. Wahrsch.*, vol. 30, pp. 65–86, 1974.

[5] A. J. Krener, R. Frezza, and B. C. Levy, "Gaussian reciprocal processes and self-adjoint differential equations of second order," *Stochastics and Stochastic Reps.*, vol. 34, pp. 29–56, 1991.

[6] B. C. Levy, R. Frezza, and A. J. Krener, "Modeling and estimation of discrete-time Gaussian reciprocal processes," *IEEE Trans. Automat. Contr.*, vol. 35, no. 5, pp. 1013–1023, 1991.

[7] A. J. Krener, "Reciprocal diffusions and stochastic differential equations of second order," *Stochastics*, vol. 24, pp. 393–424, 1988.

[8] B. C. Levy, A. J. Krener, "Dynamics and Kinematics of Reciprocal Diffusions," *J. Math. Phys.*, vol. 34, pp. 1846–1975, May 1993.

[9] M. Mariton, *Jump Linear Systems in Automatic Control.* New York: Marcel Dekker, 1990.

[10] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, Mar. 1973.

Echo Canceler Performance Analysis in Four-Wire Loop Systems with Correlated AR Subscriber Signals

John Homer, *Member, IEEE*, and Iven M. Y. Mareels, *Senior Member, IEEE*

***Abstract*—By using a simple example we illustrate the effect of auto-correlation and cross correlation of subscriber signals on the achievable performance of adaptive echo cancelers in a four-wire telephone network.**

***Index Terms*—LMS adaptation, FIR filter, bias, averaging, convergence rate.**

I. Introduction

Adaptive echo cancelers are used in four-wire loop telephone networks to suppress the effects of echoes. The commonly used double echo-canceler (DEC) system with an LMS adaptive FIR echo canceler placed at each end of the four-wire loop—Fig. 1—typically performs well. However, poor performance, such as slow or inadequate echo cancelation, and bad behavior, such as signal bursting, has been observed [1]. Such behavior has been linked to the presence of correlation within and between the subscriber signals [1], [2]. One approach is to quantify this link so that line-coding schemes can be used confidently to control the subscriber signal correlation levels and enhance the performance of the DEC system [3].

In this light, various authors [2], [3] have determined bounds on the correlation levels of the subscriber signals within which good echo cancelation is guaranteed. The bounds, however, only identify sufficient conditions for good performance. The aim of this communication is to derive an explicit equation relating asymptotic performance of the DEC system to the cross-correlation and auto-correlation levels of the subscriber signals. Although a number of simplifying assumptions are made, the results indicate the necessity of reducing signal correlation levels if good echo cancelation is to be achieved.

Consider the DEC system of Fig. 1, with subscribers 1 and 2, at either end of the network, sending signals s_1 and s_2 and receiving

Manuscript received October 27, 1993; revised May 20, 1994. The activities of the Cooperative Research Centre for Robust and Adaptive Systems (CRCRASys) are funded by the Australian Commonwealth Government under the Cooperative Research Centres Program.

J. Homer is with the Department of Electrical and Computer Engineering, University of Queensland, St. Lucia, Qld. 4067, Australia.

I. M. Y. Mareels is with CRCRASys and the Department of Systems Engineering, Research School of Information Sciences and Engineering, Australian National University, Canberra, ACT 0200, Australia.

IEEE Log Number 9406686.