SRC TR 87-185

A Bound Approach To Asymptotic Optimality In Nonlinear Filtering Of Diffusion Processes

by

L. Saydy and G.L. Blankenship

A BOUND APPROACH TO ASYMPTOTIC OPTIMALITY IN NONLINEAR FILTERING OF DIFFUSION PROCESSES

By

L. Saydy [1] and G.L. Blankenship [2]

ABSTRACT

The asymptotic behavior as a small parameter $\epsilon \to 0$ is investigated for one dimensional nonlinear filtering problems. Both weakly nonlinear systems (WNL) and systems measured through a low noise channel are considered. Upper and lower bounds on the optimal mean square error combined with perturbation methods are used to show that, in the case of WNL, the Kalman filter formally designed for the underlying linear systems is asymptotically optimal in some sense. In the case of systems wiyh low measurement noise, three asymptotically optimal filters are provided, one of which is linear. Examples with simulation results are provided.

KEYWORDS : Nonlinear filtering, Kalman filtering, bounds on the optimal mean square error, asymptotic methods.

[1] Electrical Engineering Department and Systems Research Center, University Of Maryland, College Park, MD, 20742. This work supported in part by NSF Grant CDR-85-00108.

[2] Electrical Engineering Department and Systems Research Center, University Of Maryland, College Park, MD, 20742.

1 INTRODUCTION :

We consider the Ito stochastic model:

$$\begin{aligned} dx_t &= g(t, x_t)\, dt + \sigma(t)\, dw_t \\ dy_t &= h(t, x_t)\, dt + \rho(t)\, dv_t \\ x(0) &= x_0 \quad ; \quad 0 \le t \le T \end{aligned} \tag{1}$$

where g, h, α and ρ are smooth functions of their arguments, $\{v_t\}$, $\{w_t\}$ are independent Wiener processes, x_0 a random variable independent of $\{v_t\}$, $\{w_t\}$. Given this model one is interested in computing least squares estimates of functions of the signal x_t given $\sigma\{y_s, 0 \le s \le t\}$, the σ-algebra generated by the observations, i.e., quantities of the form $E[\phi(x_t) \mid \sigma\{y_s, 0 \le s \le t\}]$. In many applications this computation must be done recursively. This involves the conditional probability density $p^y(t, x)$ which satisfies a nonlinear stochastic partial differential equation, the Kushner-Stratonovich equation [1]. By considering an unnormalized version of p^y, the above problem can be reduced to the study of the Duncan-Mortenson-Zakai (DMZ) equation which is linear ([6]).

The filtering problem was completely solved in the context of finite dimensional linear Gaussian systems by Kalman and Bucy [2], [3] in 1960-61, and the resulting Kalman filter (KF) has been widely applied. Apart from a few special cases [4], [5] the nonlinear case is far more complicated; the evolution of the conditional statistics is, in general, an infinite dimensional system.

Although progress has been made using the DMZ equation, optimal algorithms are not generally available. The performance of suboptimal designs, however derived, may be based on lower and upper bounds on the optimal mean square error (MS-error)) $p(t)$ ([7]). This approach is used here to investigate the asymptotic behavior of a class of nonlinear filtering problems, namely weakly nonlinear systems ([8]) and systems with low measurement noise level ([9]-[12]). Systems of the first type are modeled as:

$$\begin{aligned} dx_t &= a(t)x_t\,dt + \epsilon f(t, x_t)dt + \sigma(t)dw_t \\ dy_t &= c(t)x_t\,dt + \rho(t)dv_t \end{aligned} \tag{2}$$

while those of the second type are:

$$\begin{aligned} dx_t &= g(t, x_t)dt + \sigma(t)dw_t \\ dy_t &= h(t, x_t)dt + \epsilon\, dv_t \end{aligned} \tag{3}$$

It is well known that for filtering problems of this type there may be no finite set of equations which propagate the conditional mean. We are interested in (one dimensional) suboptimal filters which are asymptotically optimal in the sense that the corresponding a priori mean square error (MSE) is identical, up to some power of ϵ, to the optimal one.

Weakly nonlinear systems have been studied in [12],[16],[17]. In [12], Brockett showed that in the general case, even to be optimal in the asymptotic sense, such filters must evolve in higher dimensional spaces than x_t does. One question of particular interest is to study the effect of the weak nonlinearity on the filtering performance. In other words the question is whether the Kalman filter (" KF"), formally designed for the underlying linear system and driven by the observation $\{y_t\}$ in (2) is asymptotically optimal for small ϵ. (Notice that these are observations from a nonlinear system).

In section 3, it is shown that for a particular class of nonlinearities f (those with bounded derivatives), the " KF" and the so-called bound optimal filter (BOF, section 2), both of which are one dimensional filters with precomputable (non random) gains, are asymptotically optimal as $\epsilon \to 0$.

Next, the low measurement noise case, first studied in [9]-[12], is treated in section 4 where the BOF and a constant gain version of it are shown to be asymptotically optimal, in addition, an even simpler (not involving the drift and linear) asymptotically optimal filter is obtained. Some of these results have been obtained in [9], [12] by a different approach (e.g. an elaborate WKB procedure applied directly to the DMZ equation in Fisk-Statonovich form [9]), while here, basic bounds on the a priori optimal MS-error and perturbation methods are used. Examples with simulation results are provided in section 5.

2 LOWER AND UPPER BOUNDS ON THE OPTIMAL MS-ERROR :

Let us consider the one dimensional version of (1) where x_0 is for simplicity assumed to be $N(0,\sigma_0^2)$; g and h are such that (1) has a unique solution ([18]), differentiable with continuous partial derivatives and satisfying the following hypotheses:

$H_1 \;:\; | g_x(t,x) - \alpha(t) | < \Delta\alpha(t)$

$H_2 \;:\; | h_x(t,x) - \beta(t) | < \Delta\beta(t) \quad ; \quad \underline{\beta}(t) := \beta(t) - \Delta\beta(t) \geq 0$

which we denote by

$$g \in < [\, \alpha(t) \,,\, \Delta\alpha(t) \,] \quad ; \quad h \in < [\, \beta(t) \,,\, \Delta\beta(t) \,]$$

define

$$\overline{\alpha}(t) := \alpha(t) + \Delta\alpha(t) \quad ; \quad \underline{\alpha}(t) := \alpha(t) - \Delta\alpha(t) \tag{4}$$

$$\overline{\beta}(t) := \beta(t) + \Delta\beta(t) \quad ; \quad \underline{\beta}(t) := \beta(t) - \Delta\beta(t) \tag{5}$$

$$p(t) := E\,(\, x_t - E(x_t \mid y_0^t)\,)^2 \tag{6}$$

$$p^*(t) := E\,(\, x_t - x_t^*)^2 \tag{7}$$

where x_t^* is the BOF and is given by

$$\begin{aligned} dx_t^* &= g(t, x_t^*)\, dt + \frac{\underline{\beta}^2(t)}{\rho^2(t)}\, u(t)\, [\, dy_t - h(t, x_t^*) dt \,] \\ x_0^* &= 0 \end{aligned} \tag{8}$$

$$\dot{u}(t) = \sigma^2(t) + 2\overline{\alpha}(t) u(t) - \frac{\underline{\beta}^2(t)}{\rho^2(t)}\, u^2(t) \quad ; \quad u(0) = \sigma_0^2. \tag{9}$$

The stochastic process satisfying the above nonlinear SDE is referred to as the bound optimal filter (BOF). Clearly the BOF is readily implementable with precomputable (non random) gain and it coincides with the Kalman filter when f and g are linear. Moreover, the BOF is " bound optimal" in the sense that, among all nonlinear filters given by (8) but with arbitrary non random, continuous gains $k(t)$, the choice $k^*(t) := \frac{\underline{\beta}(t)}{\rho^2}\, u(t)$ yields a nonlinear filter (the BOF) that has the tightest upper bound on the corresponding MS-error. Furthermore, this upper bound is precisely $u(t)$ (see [7],[13],[14]).

The following result, proved in [7]([13]), provides explicit lower and upper bounds on the (unknown) optimal MS-error $p(t)$.

Theorem 2-1:

Let $p(t)$, $p^*(t)$ and $u(t)$ be as in (6),(7) and (9) respectively. Then:

$$0 \leq l(t) \leq p(t) \leq p^*(t) \leq u(t)$$

where

$$\begin{aligned} \dot{l}(t) &= \sigma^2(t) + 2\,\underline{\alpha}(t)\, l(t) - \frac{1}{\rho^2}\left[\, \overline{\beta}^2(t) + 4\,\frac{\rho^2(t)}{\sigma^2(t)}\,(\Delta\alpha(t))^2 \,\right] l^2(t) \\ l(0) &= \sigma_0^2 \end{aligned} \tag{10}$$

Remark:

Since $l(t)$ and $u(t)$ both satisfy ode's of the Riccati type, the Theorem says that the optimal MS-error $p(t)$ in the nonlinear filtering problem is bounded by those in two corresponding Kalman filtering problems, the coefficients of which are obvious from (9) and (10).

Definition:

Let $\{x_t^s\}$ be any suboptimal filter, $p^s(t,\epsilon) := E\,(\,x_t - x_t^s\,)^2$ and $p(t,\epsilon) := E\,[\,x_t - E(x_t \mid \mathcal{Y}_0^t\,)\,]^2$. Then $\{x_t^s\}$ is said to be asymptotically optimal if $p(t,\epsilon)$ and $p^s(t,\epsilon)$ agree up to some power of ϵ in a nontrivial way.

Proof of asymptotic optimality for a given suboptimal filter $\{x_t^s\}$ uses the argument that if one can bound $p(t,\epsilon)$, $p^s(t,\epsilon)$ as in

$$0 \leq l^s(t) \leq p(t,\epsilon) \leq p^s(t,\epsilon) \leq u^s(t,\epsilon)$$

for some tractable bounds l^s and u^s, then it suffices to show that the first terms in the corresponding asymptotic expansions are identical.

3 WEAKLY NONLINEAR SYSTEMS :

Let x_t and y_t be given by

$$\begin{aligned} dx_t &= g(t,x_t)dt + \epsilon f(t,x_t) + \sigma(t)dw_t \quad , \quad 0 \le t \le T \\ dy_t &= h(t,x_t)dt + \rho(t)dv_t \end{aligned} \tag{1}$$

where x_0 is $N(0,\sigma_0^2)$, $\{w_t\}$, $\{v_t\}$ are Brownian motions independent of x_0; f , g , and h have enough smoothness to guarantee the well posedness of (1).

In the case $\epsilon > 0$ is a small parameter, g and h are linear, this type of systems are referred to as weakly nonlinear systems (WNL). WNL systems were studied in [8] where it was shown that if, e.g., $f(t,x) = x^3$, then there does not exist a reduced order (i.e. one dimensional) filter which has the optimal asymptotic performance. Our goal here is to exhibit one dimensional filters that are always asymptotically optimal for a restricted class of nonlinearities f , namely those with bounded derivatives.

In the next two subsections upper and lower bounds on $p(t) := E(x_t - E(x_t \mid \mathcal{Y}_0^t))^2$, $p^*(t) := E(x_t - x_t^*)^2$ and $p^k(t) := E(x_t - x_t^k)^2$ (x_t^*, x_t^k being the BOF and " KF" estimators respectively) are used to establish that in the weakly nonlinear case, that is in the case g and h are linear, both filters are asymptotically optimal in the sense that p , p^* and p^k are the same up to first order in ϵ.

3-1 Asymptotic optimality of the BOF:

Let x_t and y_t by (1) and assume that:

- $g \in < [\, a(t)\, , \Delta a(t)\,]$; $f \in < [\, \mu(t)\, , \Delta\mu(t)\,]$
- $h \in < [\, c(t)\, , \Delta c(t)\,]$
- $\underline{c}(t) := c(t) - \Delta c(t) > 0$; $t \ge 0$

We recall that here the BOF x_t^* is given by:

$$dx_t^* = g(t,x_t^*)dt + \epsilon f(t,x_t^*)dt + \frac{c(t)}{\rho^2(t)} u(t)[dy_t - h(t,x_t^*)dt]$$
$$x^*(0) = 0 \tag{2}$$
$$\dot{u} = \sigma^2(t) + 2(\bar{a}(t) + \epsilon\bar{\mu}(t))\, u(t) - \frac{c^2(t)}{\rho^2(t)} u^2 \quad ; \quad u(0) = \sigma_0^2$$

Proposition 3-1 :

If $\Delta a(t) = \Delta c(t) = 0$ and $c(t) > 0$, then the BOF is asymptotically optimal as $\epsilon \to 0$, i.e.

$$p^*(t) \sim p(t) = r(t) + O(\epsilon) \quad , \quad 0 \le t \le T \tag{3}$$

where

$$\dot{r} = \sigma^2(t) + 2a(t)r(t) - \frac{c^2(t)}{\rho^2(t)} r^2 \quad ; \quad r(0) = r_0^2 \tag{4}$$

Remark :

If furthermore, the system is time invariant then

$$p^*(t) = p(t) = r(t) + 2\,\epsilon\,\mu \int_0^t \phi(t,s) r(s) ds + O(\epsilon, \Delta\mu) \tag{5}$$

where

$$r(t) = \frac{\rho^2}{c^2}\left\{ a + \delta \frac{1-Ae^{-2\delta t}}{1+Ae^{-2\delta t}} \right\} \tag{6}$$

$$\delta = \sqrt{a^2 + \frac{\sigma^2}{\rho^2} c^2} \quad ; \quad A = \frac{\frac{c^2}{\rho^2}(a+\delta) - \sigma_0^2}{\sigma_0^2 - \frac{c^2}{\rho^2}(a-\delta)} \tag{7}$$

$$\phi(t,s) = e^{2a(t-s)} \exp\left\{ -2\frac{c^2}{\rho^2} \int_s^t r(\tau) d\tau \right\} \tag{8}$$

here $O(x,y)$ means order of each one of the arguments separately.

Proof :

It readily follows from the above assumptions that $(g + \epsilon f) \in < [\, a(t) + \epsilon\, \mu(t)\, , \Delta a(t) + \epsilon\, \Delta\mu(t)\,]$.

From Theorem 2-1 we get

$$0 \leq l(t) \leq p(t) \leq p^*(t) \leq u(t) \tag{9}$$

where:

$$\begin{aligned} \dot{u} &= \sigma^2(t) + 2(\,\overline{a}(t) + \epsilon\overline{\mu}(t)\,)u - \frac{\underline{c}^2(t)}{\rho^2(t)}\,u^2 \\ u(0) &= \sigma_0^2 \end{aligned} \tag{10}$$

$$\begin{aligned} \dot{l} &= \sigma^2(t)+2(\underline{a}(t)+\epsilon\underline{\mu}(t))l - \frac{1}{\rho^2(t)}\,[\overline{c}^2(t)+4\frac{\rho^2(t)}{\sigma^2(t)}(\Delta a(t)+\epsilon\Delta\mu(t))^2]l^2 \\ l(0) &= \sigma_0^2 \end{aligned} \tag{11}$$

expanding $u(t)$ in the form:

$$u(t) \sim \sum_{i=0}^{\infty} u_i(t)\,\epsilon^i \tag{12}$$

gives :

$$\begin{aligned} u^2(t) &\sim \sum_{k=0}^{\infty} c_k\,\epsilon^k \\ c_k &= \sum_{j=0}^{n} u_j(t)u_{n-j}(t) \end{aligned} \tag{13}$$

Substituting (12) and (13) in (10) and equating powers of ϵ yields:

$$\dot{u}_0 = \sigma^2(t) + 2\overline{a}(t)u_0 - \frac{\underline{c}^2(t)}{\rho^2(t)}\,u_0^2 \quad , \quad u_0(0) = \sigma_0^2 \tag{14}$$

$$\dot{u}_1 = 2[\,\overline{a}(t) - \frac{\underline{c}^2(t)}{\rho^2(t)}\,u_0(t)\,]u_1 + 2\overline{\mu}(t)u_0(t) \quad , \quad u_1(0) = 0 \tag{15}$$

Proceeding similarly for $l(t)$, one obtains:

$$\begin{aligned} \dot{l}_0 &= \sigma^2(t) + 2\underline{a}(t)l_0 - \frac{1}{\rho^2}\,[\,\overline{c}^2(t) + 4\frac{\rho^2(t)}{\sigma^2(t)}\,\Delta a^2(t)\,]\,l_0^2 \\ l_0(0) &= \sigma_0^2 \end{aligned} \tag{16}$$

$$\begin{aligned} \dot{l}_1 &= 2\,[\,\underline{a}(t) - \frac{1}{\rho^2(t)}\,(\overline{c}^2(t)+4\frac{\rho^2(t)}{\sigma^2(t)}\,\Delta a^2(t)\,)\,l_0\,]\,l_1 + 2\underline{\mu}(t)\,l_0 - 8\frac{\Delta a\,\Delta\mu}{\sigma^2}\,l_0^2 \\ l_1(0) &= 0 \end{aligned} \tag{17}$$

(here $\Delta a^2 := (\Delta a\,)^2$)

It is clear from (14) and (16) that $u_0(t)$ and $l_0(t)$ are different in the general case but coincide with $r(t)$ if $\Delta a = \Delta c = 0$ that is:

$$g(t,x) = a(t)x \quad \text{and} \quad h(t,x) = c(t)x$$

This completes the proof.

Now if the system is time invariant, i.e.,

$$a(t) = a \quad ; \quad \mu(t) = \mu \quad ; \quad c(t) = c \quad ; \quad \sigma(t) = \sigma \quad \text{and} \quad \rho(t) = \rho$$

then one easily gets the results in the remark above by using the Riccati transformation $r = \frac{\rho^2}{c^2} \frac{\dot{w}}{w}$ to solve (4) and the variation of constants formula in (15) and (17).

3-2 Asymptotic optimality of the KF :

The question considered here is whether one could, in the case of weakly nonlinear systems, ignore the nonlinear part in the drift, use the Kalman filter designed for the underlying linear system (driven by y_t) and be able to achieve asymptotic optimality as $\epsilon \to 0$. It is important, however, to notice that eventhough this scheme is being referred to as the " KF", it has little to do with the regular Kalman filter, the reason being that the " KF" is driven by observations from a nonlinear system.

Accordingly, Let $g(t,x) = a(t)x$, $h(t,x) = c(t)x$ and assume that $f \in < [\, \mu(t) \,,\, \Delta\mu(t) \,]$, $c(t) > 0$; then the " KF" is given by:

$$dx_t^k = a(t)x_t^k dt + \frac{c(t)}{\rho^2(t)} r(t) [\, dy_t - c(t)x_t^k dt \,] \quad ; \quad x^k(0) = 0 \tag{18}$$

where $r(t)$ is as in (4).

Proposition 3-2 :

Under the above assumption, the " KF" is asymptotically optimal as $\epsilon \to 0$ in the sense that:

$$p^k(t) \sim p(t) = r(t) + O(\epsilon) \qquad 0 \le t \le T$$

Proof :

We first derive an upper bound on $p^k(t) := E(x_t - x_t^k)^2$ where x_t^k is given by (18).

Let $\overline{x}_t := x_t - x_t^k$; then

$$d\overline{x}_t = [\,\overline{g}_t - c(t)G(t)\overline{x}_t\,]dt + \sigma(t)\,dw_t - \rho(t)\,G(t)\,dv_t \tag{19}$$

where $G(t) := \dfrac{c(t)}{\rho^2} r(t)$ and $\overline{g}_t = a(t)\overline{x}_t + \epsilon f(t,x_t)$. Applying *Itô's* chain rule ([1]) gives

$$d\overline{x}_t^{\,2} = [\,\sigma^2 + \rho^2 G^2\,]\,dt + 2\overline{x}_t\,d\overline{x}_t \tag{20}$$

Taking expectations on both sides yields :

$$\frac{d}{dt}E\,\overline{x}_t^{\,2} = \dot{p}^k(t) = \sigma^2 + \rho^2 G^2 + 2E\overline{x}_t\,[\,\overline{g}_t - cG\overline{x}_t\,]$$

$$\dot{p}^k(t) = \sigma^2 + \rho^2 G^2 + 2\,E\,\overline{x}_t\,\overline{g}_t - 2\,c\,G\,E\,\overline{x}_t^{\,2}\quad ,\quad p^k(0) = \sigma_0^2$$

$$\dot{p}^k = \sigma^2 + \rho^2 G^2 + 2(a - cG)p^k + 2\epsilon E\,\overline{x}_t\,f(t,x_t) \tag{21}$$

Clearly

$$2E\,\overline{x}_t\,f(t,x_t) \le E\,\overline{x}_t^{\,2} + E\,f^2(t,x_t) = p^k(t) + E\,f^2(t,x_t) \tag{22}$$

By the comparison theorem (see Appendix): $p^k(t) \le q(t)$; $q(0) = \sigma_0^2$ where

$$\begin{aligned}\dot{q}(t) &= \sigma^2 + \rho^2 G^2 + 2(a - cG)q + \epsilon(q + Ef^2)\\ &= \sigma^2 + \rho^2 G^2 + \epsilon E f^2 + [\,2(a - cG) + \epsilon\,]\,q\end{aligned} \tag{23}$$

which we rewrite as

$$\begin{aligned}\dot{q} &= i(t) + j(t)q \quad , \quad q(0) = \sigma_0^2\\ i(t) &= \sigma^2 + \frac{c^2}{\rho^2} r^2 + \epsilon\,E\,f^2(t,x_t)\\ j(t) &= \epsilon + 2\,[\,a - \frac{c^2}{\rho^2} r(t)\,]\end{aligned} \tag{24}$$

We therefore have the following bounds:

$$l(t) \le p(t) \le p^k(t) \le q(t) \tag{25}$$

where:

$$\begin{aligned} \dot{l} &= \sigma^2 + 2(a + \epsilon \underline{\mu}) l - \frac{1}{\rho^2} [c^2 + 4 \frac{\rho^2}{\sigma^2} \Delta\mu^2 \epsilon^2] l^2 \\ l(0) &= \sigma_0^2 \end{aligned} \tag{26}$$

Expanding $q(t)$ in the form:

$$q(t) \sim \sum_{i=0}^{\infty} q_i(t) \epsilon^i$$

and equating powers of ϵ yields:

$$\dot{q}_0 = \sigma^2(t) + \frac{c^2(t)}{\rho^2(t)} r^2(t) + 2[a(t) - \frac{c^2(t)}{\rho^2(t)} r(t)] q_0 \quad ; \quad q_0(0) = \sigma_0^2$$

Let $w := q_0(t) - l_0(t)$. Then from the previous section it follows by making $\Delta a = 0$ in (16) that $w(t) = q_0(t) - r(t)$. By differentiating we get

$$\dot{w}(t) = \frac{c^2(t)}{\rho^2(t)} r^2(t) + 2[a(t) - \frac{c^2(t)}{\rho^2(t)} r(t)] q_0^{\epsilon} - 2a(t) r(t) + \frac{c^2(t)}{\rho^2(t)} r^2(t)$$

This in turn easily becomes:

$$\dot{w} = 2 [a(t) - \frac{c^2(t)}{\rho^2(t)} r(t)] w \quad ; \quad w(0) = 0$$

The solution of which clearly is $w(t) = 0$ which implies $q_0 = r$.

The proof is complete.

4 LOW MEASUREMENT NOISE LEVEL :

Consider the system :

$$\begin{aligned} dx_t &= g(t, x_t) dt + \sigma(t) dw_t \\ dy_t &= h(t, x_t) dt + \epsilon \, dv_t \end{aligned} \tag{27}$$

where $\quad g \in < [a(t), \Delta a(t)]$

$h \in < [c(t), \Delta c(t)] \quad ; \quad \underline{c}(t) \geq 0 \quad ; \quad t \geq 0$

and $\epsilon > 0$ is a small parameter (this is the case in many practical situations [10], [12]).

The optimal a priori MS-error is bounded from above and below; perturbation methods for the bounds are used to show that the upper bound approaches the lower one as ϵ becomes smaller.

The result is quoted for h linear but holds for nonlinearities h which tend asymptotically to a linear function, i.e., Δc is small (see remark 2). This type of (almost linear) nonlinearities arise in practice and are usually modeled as being linear [14].

Proposition 4-1 :

Assume that $\Delta c = 0$ (i.e. h is linear) and $c(t) > 0$, then the optimal MS-error $p(t)$ satisfies the following

$$p(t) = \frac{\sigma(t)}{c(t)} \epsilon + O(\epsilon) = E(x_t - x_t^F)^2 \tag{28}$$

where $\lim_{\epsilon \to 0} \frac{O(\epsilon)}{\epsilon} = 0$ and x_t^F denotes any one of the three asymptotically optimal filters listed below.

(F_1) The BOF :

$$dx_t^* = g(t, x_t^*)dt + \frac{c(t)}{\epsilon^2} u(t) [dy_t - c(t) x_t^* dt] \quad , \quad x^*(0) = 0 \tag{29}$$

$$\dot{u}(t) = \sigma^2(t) + 2\,\bar{a}(t)\,u(t) - \frac{c^2(t)}{\epsilon^2} u^2(t) \quad ; \quad u(0) = \sigma_0^2 \tag{30}$$

(F_2) The constant gain BOF (CGBOF) :

$$dx_t^c = g(t, x_t^c)dt + \frac{\sigma(t)}{\epsilon} [dy_t - c x_t^c dt] \quad ; \quad x_t^c(0) = 0 \tag{31}$$

(F_3) The linear (first approximation) BOF :

$$dx_t^L = \frac{\sigma(t)}{\epsilon} [dy_t - c(t)\, x_t^L\, dt] \quad ; \quad x^L(0) = 0 \tag{32}$$

(28) is proven for each case separately.

Proof of ($\mathbf{F}_1$):

From Theorem 2-1 we get :

$$l(t) \le p(t) \le p^*(t) = E(x_t - x_t^*)^2 \le u(t) \tag{33}$$

$$\dot{u} = \sigma^2(t) + 2\bar{a}(t)u - \frac{\underline{c}^2(t)}{\epsilon^2}u^2 \quad ; \quad u(0) = \sigma_0^2 \tag{34}$$

$$\begin{aligned} \dot{l} &= \sigma^2(t) + 2\underline{a}(t)l - \frac{1}{\epsilon^2}[\bar{c}^2(t) + 4\frac{\epsilon^2}{\sigma^2(t)}(\Delta a)^2]l^2 \\ l(0) &= \sigma_0^2 \end{aligned} \tag{35}$$

It can be easily seen by inspection of (34) and (35) that $u(t)$ and $l(t)$ are of different order in ϵ if Δc is nonzero. Let's show this explicitly.

Expanding $u(t)$ as

$$u(t) \sim \sum_{n=0}^{\infty} u_n(t)\,\epsilon^n \tag{36}$$

yields

$$\begin{aligned} u^2(t) &\sim \sum_{n=0}^{\infty} d_n\,\epsilon^n \\ d_n(t) &= \sum_{j=0}^{n} u_j(t)u_{n-j}(t) \end{aligned} \tag{37}$$

e.g.

$$\begin{aligned} d_0(t) &= u_0^2(t) \\ d_1(t) &= 2u_0(t)u_1(t) \\ d_2(t) &= 2u_0u_2 + u_1^2 \end{aligned}$$

Substituting (36) and (37) in (34) gives:

$$\sum_{n=0}^{\infty} \dot{u}_n\,\epsilon^n = \sigma^2(t) + 2\bar{a}\sum_{0}^{\infty} u_n\,\epsilon^n - \frac{\underline{c}^2}{\epsilon^2}\sum_{0}^{\infty} d_n\,\epsilon^n \tag{38}$$

Equating powers of ϵ, starting with ϵ^{-2}, yields $d_0 = 0$, i.e., $u_0(t) = 0$. This in turn implies that $d_1 = 0$.

Similarly $\sigma^2 - \underline{c}^2 d_2 = 0$. But since $d_2 = u_1^2$, it follows that $u_1(t) = \dfrac{\sigma(t)}{\underline{c}(t)}$, i.e.,

$$u(t) = \frac{\sigma(t)}{\underline{c}(t)}\epsilon + O(\epsilon^2) \qquad \text{for every} \quad 0 \le t \le T \tag{39}$$

By a similar procedure we get $l_0 = 0$ and $l_1 = \dfrac{\sigma(t)}{\bar{c}(t)}$; that is

$$l(t) = \frac{\sigma(t)}{\bar{c}(t)}\epsilon + O(\epsilon^2) \qquad 0 \le t \le T \tag{40}$$

We conclude from (39) and (40) that if $\Delta c = 0$, i.e., $h(t,x) = c(t)x$ then:

$$u(t) \sim l(t) = \frac{\sigma(t)}{c(t)}\epsilon + O(\epsilon^2) \qquad 0 \le t \le T \tag{41}$$

which establishes the asymptotic optimality of the BOF as $\epsilon \to 0$.

This completes the proof of (F_1).

Note: These approximations are obviously not valid in the immediate vicinity of $t = 0$ where $u(0) = l(0) = \sigma_0^2$. This (boundary layer) problem is negligible. It can indeed be easily shown that the duration of the transient regime for this type of ode's is $O(\epsilon)$ (also see figure 2).

This suggests the following:

(i) Since $u(t) = \epsilon u_1(t) + O(\epsilon^2)$, one can replace $u(t)$ in (22) by $\epsilon u_1 = \epsilon \frac{\sigma(t)}{c(t)}$ and attempt to achieve asymptotic optimality as well. The new filter clearly would have the advantage that the gain $k(t) = \frac{\sigma(t)}{\epsilon}$, thus avoiding solving a Riccati equation, resulting in faster computations.

(ii) If the answer to (i) is affirmative, the next question is whether the same thing would hold for the first approximation (when expanding x_t^c) filter:

$$dx_t^L = \frac{\sigma(t)}{\epsilon}[\, dy_t - c(t)\, x_t^L dt \,]$$

It turns out that both filters are asymptotically optimal as is shown next.

Proof of (F$_2$):

An upper bound on the MS-error corresponding to filters such as (F_2) can be obtained by following the first steps in the proof of proposition 3-2 (also section 2-2 in [7]). In this case

$$E\,(\, x_t - x_t^c\,)^2 \le u^k(t) \tag{42}$$

where $(k(t) = \frac{\sigma(t)}{\epsilon})$:

$$\dot{u}^k = 2\sigma^2(t) + 2\left[\,\overline{a}(t) - \frac{\sigma(t)c(t)}{\epsilon}\,\right] u^k$$
$$u^k(0) = \sigma_0^2 \tag{43}$$

By setting $u^k(t) \sim \sum u_i^k(t)\epsilon^i$ in (43), one easily obtains

$$u_0^k(t) = 0 \qquad ; \qquad u_1^k(t) = \frac{\sigma(t)}{c(t)}$$

Hence:

$$p(t) = E\,(x_t - x_t^c)^2 = \frac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon^2)\ , \qquad 0 \le t \le T$$

(Recall that: $\quad p(t) \ge l(t) = \dfrac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon^2)$).

Proof of $(\mathbf{F}_3)$:

Similarly, it is readily obtained that $p^L(t) := E\,[\,x_t - x_t^L\,]^2$ satifies

$$\dot{p}^L = 2\sigma^2(t) + 2\,E\,(\,x_t - x_t^L\,)g(t, x_t) - 2\frac{c(t)\sigma(t)}{\epsilon}\,p^L$$

Using the Schwartz inequality:

$$E\ ab \le E^{\frac{1}{2}}\,a^2\ .\ E^{\frac{1}{2}}\,b^2$$

and the comparison theorem (see appendix) we get $\quad p^L(t) \le u^L(t) \quad$ where

$$\dot{u}^L = 2\sigma^2(t) + 2\theta(t)\,(u^L)^{\frac{1}{2}} - 2\frac{c(t)\sigma(t)}{\epsilon}\,u^L \tag{44}$$

with $\theta(t) = E^{\frac{1}{2}} g^2(t, x_t)$. Expanding $u^L \sim \sum_0^\infty u_i^L\,\epsilon^{\frac{i}{2}}$ in (44) and equating powers of ϵ

gives $u_0^L = u_1^L = 0$ and $u_2^L = \dfrac{\sigma(t)}{c(t)}$, hence

$$u^L(t) = \frac{\sigma(t)}{c(t)}\,\epsilon + O(\,\epsilon^{\frac{3}{2}}\,) \quad , \qquad 0 \le t \le T \tag{45}$$

Therefore

$$p(t) = p^L(t) = \frac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon) \qquad 0 \le t \le T$$

The proof is complete.

Remark (1) :

(i) If $\sigma(t) = \sigma$ and $c(t) = c$ then $\dot{u}_1(t) = \dot{l}_1(t) = 0$ and the next terms in the expansion of $u(t)$ and $l(t)$ are :

$$u_2(t) = \frac{1}{c^2}\,\overline{a}(t)$$
$$l_2(t) = \frac{1}{c^2}\,\underline{a}(t)$$

so that $u(t) = l(t) + O(\epsilon^3)$ if and only if $\Delta a = 0$, i.e., both g and h are linear.

(ii) In [19], it was shown that for incrementally conic nonlinearities we have the following lower bound $L(t)$:

$$p(t) \geq L(t) = (1 - s(t))\,r(t) \tag{46}$$

where $s(t)$ is the unique nonnegative root of

$$(1 - s(t))\,e^{s(t)} = e^{-d(t)} \tag{47}$$

$$d(t) = \int_0^t \left[\frac{\Delta a(s)}{\sigma^2(s)} + \frac{\Delta c^2(s)}{\epsilon^2} \right] q(s)\,ds \tag{48}$$

$$\begin{aligned} \dot{q} &= \sigma^2(t) + \frac{c^2(t)}{\epsilon^2}\,r^2(t) + 2\left[\,\overline{a}(t) - \frac{c^2(t)}{\epsilon^2}\,r\,\right] q \\ q(0) &= \sigma_0^2 \end{aligned} \tag{49}$$

$$\begin{aligned} \dot{r} &= \sigma^2(t) + 2a(t)r - \frac{c^2(t)}{\epsilon^2}\,r^2 \\ r(0) &= \sigma_0^2 \end{aligned} \tag{50}$$

From (34) and (39) we readily get that $r(t) = \frac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon^2)$. It is therefore clear from (46) that if $s(t) = O(\epsilon)$, then $L(t) = \frac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon^2)$, as we have used here. This is indeed the case: (49) implies $q(t) = O(\epsilon)$ and (48) that $d(t) = O(\epsilon)$ $(\Delta c = 0)$. Assuming $s(t) \sim \sum_0^\infty s_n\,\epsilon^n$ and letting ϵ go to zero in (47) gives that $1 - s_0 = e^{-s_0}$ necessarily. This has the unique solution $s_0 = 0$, hence $s(t) = O(\epsilon)$.

Remark (2) : Almost linear observations.

The same results in the previous proposition can be extended to the particular class of non-linearities $h \in < [c \ , \Delta c\,]$ where Δc is also a small parameter. Indeed, the upper and lower bounds u and l on $p(t)$ and $p^*(t) := E(x_t - x_t^*)^2$ where x_t^* is the BOF in (F_1) (with cx_t^* and c replaced by $h(x_t^*)$ and $\overline{c}$) are given by (39) and (40):

$$
\begin{aligned}
u(t) &= \frac{\sigma(t)}{\underline{c}(t)}\,\epsilon + O(\epsilon^2) \\
&= \frac{\sigma(t)}{c(t)}\,\epsilon\,(\,1 + \frac{\Delta c}{c} + O((\Delta c\,)^2)\ ) + O(\epsilon^2) \\
&= \frac{\sigma(t)}{c(t)}\,\epsilon + \frac{1}{c}\,\epsilon\,\Delta c\ + O(\epsilon^2) + \epsilon O((\Delta c\,)^2)
\end{aligned}
$$

Thus for small $\Delta\ c$

$$
u(t) = \frac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon\ ,\ \Delta c\,)
$$

Similarly

$$
\begin{aligned}
l(t) &= \frac{\sigma(t)}{\overline{c}(t)}\,\epsilon + O(\epsilon^2) \\
&= \frac{\sigma(t)}{c(t)}\,\epsilon\,(\,1 - \frac{\Delta c}{c} + O((\Delta c\,)^2)\ ) + O(\epsilon^2) \\
&= \frac{\sigma(t)}{c(t)}\,\epsilon + O(\epsilon\ ,\ \Delta c\,))
\end{aligned}
$$

It is not hard either to establish that for the analogous of the filters (F_2) and (F_3) (as in (31) and (32), but with cx replaced by $h(x)$) the upper bounds are

$$
u^k(t) = \frac{\sigma(t)}{\underline{c}(t)}\,\epsilon + O(\epsilon^2)
$$

and

$$
u^L(t) = \frac{r(t)}{\underline{c}(t)}\,\epsilon + O(\,\epsilon^{\frac{3}{2}}\,)
$$

which makes these filters asymptotically optimal too as Δc and ϵ become smaller with

$$
p(t) = \frac{\sigma(t)}{c(t)}\,\epsilon + o(\epsilon).
$$

Application to the Beneš filter ([4]):

Let

$$dx_t = f(x_t)\, dt + dw_t \tag{51}$$

$$dy_t = x_t\, dt + dv_t \tag{52}$$

where the drift f satisfies

$$f_x(x) + f^2(x) = a\, x^2 + b\, x + c \tag{53}$$

with $a \geq 0$ to prevent finite time escape situations.

As mentioned earlier, this is one of the few nonlinear filtering problems which was shown to admit a finite number of sufficient statistics.

We are interested here in investigating this type of filtering problems when the diffusion process $\{x_t\}$ is measured in a low noise channel. In particular, we would like to know what type of implementation simplifications will result from this additional assumption. Accordingly, let $\{x_t\}$ be as in (51) and:

$$dy_t = x_t\, dt + \epsilon\, dv_t \tag{54}$$

In order to know how ϵ enters Beneš' original formulas, let us resolve the DMZ equation in Fisk-Stratonovich form by following the steps outlined below. The unnormalized pdf $u(t,x)$ satisfies the following stochastic PDE:

$$\begin{aligned} du &= \left(L^*(u) - \frac{1}{2}\frac{x^2}{\epsilon^2}\, u \right) dt + \frac{x}{\epsilon^2}\, u\, dy \\ L^*(u) &= \frac{1}{2} u_{xx} - (f\, u)_x \end{aligned} \tag{55}$$

which in our case is

$$du = \left[\frac{1}{2}\, u_{xx} - f\, u_x - \left(f_x + \frac{1}{2}\, \frac{x^2}{\epsilon^2}\right) u \right] dt + \frac{x^2}{\epsilon^2}\, u\, dy \tag{56}$$

By letting $V(t,x) = e^{-\frac{1}{\epsilon^2} x\, y_t}\, u(t,x)$, the stochastic differentials in (65) are eliminated. We obtain the following classical PDE (robust DMZ):

$$V_t = \frac{1}{2}\, V_{xx} + \left(\frac{y_t}{\epsilon^2} - f\right) V_x - \left(\frac{y_t}{\epsilon^2} f + f_x + \frac{1}{2}\, \frac{x^2}{\epsilon^2} - \frac{1}{2}\frac{y_t^2}{\epsilon^4}\right) V \tag{57}$$

Using $V(t,x) = e^{\int_0^x f(\sigma)d\sigma} \rho(t,x)$ and (53) in (57), we get after some computations that:

$$\rho_t = \frac{1}{2}\rho_{xx} + \frac{1}{2}y_t \ \rho_x + \left[\frac{1}{2}\frac{{y_t}^2}{\epsilon^4} - \frac{1}{2}\frac{1}{\epsilon^2}(1+\epsilon^2 a)x^2 - \frac{1}{2}bx - \frac{1}{2}c \ \right] \rho$$

It can be easily verified that ρ is given by

$$\rho(t,x) = \exp\left\{ -\frac{(x-\mu_t)^2}{2\theta(t)} \right\}$$

where

$$\begin{aligned} \dot{\theta}(t) &= 1 - \frac{1}{\epsilon^2}(1+\epsilon^2 a)\,\theta^2(t) \quad , \quad \theta(0) = 0 \\ d\,\mu_t &= -\frac{1}{\epsilon^2}(1+\epsilon^2 a)\,\theta(t)\mu_t \ dt - \frac{1}{2}\theta(t)\ b \ dt + \frac{1}{\epsilon^2}\theta(t)dy_t \end{aligned} \tag{58}$$

$$u(t,x) = e^{\frac{1}{\epsilon^2}xy_t} \exp\left\{ \int_0^x f(\sigma)d\sigma - \frac{1}{2}\frac{(x-\mu_t)^2}{\sigma(t)} \right\} \tag{59}$$

Our goal is to see under what circumstances can μ_t be a good approximation for the conditional mean $E(x_t \mid \mathcal{Y}_0^t)$ given by:

$$E\left[x_t \mid \mathcal{Y}_0^t \right] = \int x \ \frac{u(t,x)}{\int u(t,x)} \ dx$$

It turns out that for cone bounded drifts in (53) (e.g. $f(x) = th(x)$ or linear), the following holds.

Claim :

$\{\mu_t\}$ is asymptotically optimal as $\epsilon \to 0$.

To see this, rewrite (58) in the more suggestive form:

$$d\,\mu_t = \frac{\theta(t)}{\epsilon^2}\left[dy_t - (1+\epsilon^2 a)\,\mu_t \ dt \ \right] - \frac{1}{2}\theta(t)\ b \ dt$$

and notice that $\theta(t) = \frac{\epsilon}{(1+\epsilon^2 a)^{\frac{1}{2}}} th\left((1+\epsilon^2 a)\,\frac{t}{\epsilon} \right) \sim \epsilon + O(\epsilon^3)$. It is not hard then to show that $\mu_t = \mu_t^L + O(\epsilon)$ where

$$d\,\mu_t^L := \frac{1}{\epsilon}\left[dy_t - \mu_t^L \ dt \ \right]$$

is precisely the linear BOF obtained in last proposition which was shown to be asymptotically optimal as ϵ becomes smaller.

Notice that for the particular case $f(x) = th(x)$, $a = b = 0$ and $c = 1$ and hence

$$d\mu_t = \frac{th(\frac{t}{\epsilon})}{\epsilon} [dy_t - \mu_t \, dt]$$

5 EXAMPLES AND SIMULATION RESULTS :

Example 1 : In this example, the asymptotic optimality of " KF" for WNL systems (section 3-2) is illustrated. We consider:

$$\begin{aligned} dx_t &= ax_t \, dt + \epsilon \, th(x_t) dt + \sigma dw_t \\ dy_t &= cx_t \, dt + \rho \, dv_t \\ x_0 &\sim N (m_0 , \sigma_0^2) \end{aligned}$$

where $f(.) = th(.) \in < [\frac{1}{2} , \frac{1}{2}]$, i.e., $\underline{\mu} = 0$, $\bar{\mu} = 1$, $\mu = \Delta\mu = \frac{1}{2}$.

Simulation resuls were done using Monte Carlo technique and the following numerical data:

$$\begin{aligned} a &= -1 \; , \; \sigma = \rho = 0.3 \\ c &= 1 \; , \; m_0 = 0 \; , \; \sigma_0 = 0.1 \end{aligned}$$

The results are summarized in the plots of figures 1(a),1(b) and 1(c), which correspond to different values of ϵ ($\epsilon = 0.2$, 0.1 and 0.05 respectively). In each figure, we have plotted $p^k(t) := E\,(x_t - x_t^k)^2$, $r(t)$ and $l(t)$; the latter being the lower bound on the optimal MS-error $p(t)$ which therefore lies between $l(t)$ and $p^k(t)$.

The plots corroborate the results of Proposition 3-2 in which it is stated that the " KF" is asymptotically optimal as ϵ becomes smaller and that $r(t)$ is a good approximation for the (unknown) optimal MS-error $p(t)$ in the sense that $p^k(t) \sim p(t) = r(t) + O(\epsilon)$.

Example 2 :

This second example deals with the asymptotic optimality of the BOF and CGBOF in the case of low measurements noise level filtering problems. The following model is considered:

$$\begin{aligned} dx_t &= arctg\,(x_t)dt + \sigma dw_t \\ dy_t &= cx_t\, dt + \epsilon\, dv_t \\ x_0 &\sim \mathcal{N}\,(m_0, \sigma_0^2) \end{aligned}$$

where $g(.) = arctg\,(.) \in\ < [\frac{1}{2}, \frac{1}{2}]$, i.e., $a = \Delta a = \frac{1}{2}$ and

$$a = -1 \ , \ \sigma = c = 1 \ , \ m_0 = 0 \ , \ \sigma_0^2 = 0.5$$

The simulations are summarized in figures 2(a) and 2(b) which correspond to the performance of the BOF and CGBOF respectively. Each figure contains 3 sets of plots corresponding to $\epsilon = 0.3$, 0.1 and 0.05 from top to bottom. Each set of 3 curves consist of the upper bound $u(t)$ on the BOF, the MS-error $p^F(t) = E\,(x_t - x_t^F)^2$ and the lower bound $l(t)$ on the optimal MS-error $p\,(t)$.

Again, these plots agree with the results of Proposition 4-1 in which it is stated that the BOF and CGBOF are both asymptotically optimal as ϵ becomes smaller and that $\frac{\sigma(t)}{c\,(t)}\,\epsilon$ (equal to ϵ here) is a good approximation for the (unknown) optimal MS-error $p\,(t)$.

Remark : It can be seen in figure 2(b) that the MS-error $p^c(t)$ exceeds the (BOF) upper bound $u\,(t)$ in all three cases as might be expected. To see why this is so, it suffices to recall that the CGBOF was obtained by approximating the BOF gain $k^*(t) := \frac{c\,(t)}{\epsilon^2}\, u\,(t)$ by $\frac{\sigma(t)}{\epsilon}$ since $u\,(t) \sim \frac{\sigma(t)}{c\,(t)}\,\epsilon$. However, it was remarked earlier that this last approximation does not hold in the immediate vicinity of $t = 0$ (boundary layer problem). Outside this region (which shrinks to zero as $\epsilon \rightarrow 0$), the CGBOF performes in a comparable fashion than the BOF with the speed advantage.

6 CONCLUSION

We investigated the asymptotic behavior question of one dimensional nonlinear filtering problems involving drifts with bounded derivatives using an upper and lower bound approach to show that the a priori mean square error associated with some suboptimal filters approaches the optimal one asymptotically.

This approach proved that significant information relevant to this type of problems can be infered from the knowledge of the derivative bounds (i.e., of the cone in which the nonlinearities reside).

In particular, it is shown that in the case of weakly nonlinear systems, that the " KF " (designed for the underlying linear system) is asymptotically optimal as $\epsilon \to 0$. In other words the nonlinearity can be ignored as long as the asymptotic behavior is concerned.

In the case of diffusions measured in a low noise channel, three asymptotically optimal filters were obtained, one of which is linear. Furthermore, asymptotic values for the unknown optimal MS-error were obtained in both cases.

The main point is that upper and lower bounds on the optimal MS-error, when available, may be used (in addition to performance testing of suboptimal designs) as a relatively simple tool to study certain nonlinear filtering problems.

APPENDIX

Comparison theorem [15] :

Let $F(x,y)$ and $G(x,y)$ be continuous in the rectangle

$$D: \quad |x - x_0| < a \quad , \quad |y - y_0| < b$$

and suppose that $F(x,y) < G(x,y)$ everywhere in D. Let $y(x)$ and $z(x)$ be the solutions of

$$\begin{aligned} \dot{y} &= F(x,y) \quad , \quad y(x_0) = \alpha \\ \dot{z} &= G(x,y) \quad , \quad z(x_0) = \alpha \end{aligned}$$

Let I be the largest subinterval of $(x_0 - a \,,\, x_0 + a)$ where both $y(x)$ and $z(x)$ are defined and continuous ; then for $x \in I$

$$\begin{aligned} z(x) &< y(x) \quad , \quad x < x_0 \\ z(x) &> y(x) \quad , \quad x > x_0 \end{aligned}$$

REFERENCES

[1] Jazwinski, A.H. : " Stochastic Processes and Filtering Theory " , Academic Press, New York , 1970

[2] Kalman, R.E. : " A New Approach to Linear Filtering and Prediction Problems " J.Basic Eng., Trans. ASME, Series D, Vol. 82, #1, March 1960, pp34-35

[3] Kalman, R.E. ; Bucy, R.S. : " New Results in Filtering and Prediction theory ", 1961, J. Basic Eng, Trans AMSE, Series D, Vol 83, pp 95-107

[4] Beneš , V.E. : " Exact Finite Dimensional Filters for Certain Diffusions with Non-Linear Drifts " , Stochastics, 1981

[5] Wong, W.S. : " New Classes of Finite Dimensional Nonlinear Filters " , Systems and Control Letters, 3, 1983, 155-164

[6] Baras, J. ; Blankenship, G.L. ; Mitter, S.K : " Nonlinear Filtering of Diffusion Processes " , IFAC, Triennial World Congress, Kyoto, Japan, 1981

[7] Saydy, L. ; Blankenship, G.L. : " Optimal Stationary Behavior In Some Nonlinear Stochastic Filtering Problems - A Bound Approach " , Submitted.

[8] Brockett, R.W. : " Asymptotically Optimal Estimation " , IEEE, Decision and Control Conference, 1981

[9] Katzur, R. ; Bobrovsky, B.Z ; Schuss, Z. : " Asymptotic Analysis of the Optimal Filtering Problem for One Dimensional Diffusions Measured in a Low Noise Channel, Part 1 " , to appear in SIAM J of Appl Math

[10] Bobrovsky, B.Z. ; Zakai, M. : " Asymptotic Bounds on the Minimal Error of Non Linear Filtering " , Hazewinkel,M. and Willems,J.C. (eds), Stochastic Systems : the Mathematics of Filtering and Identification and Applications, 573-581

[11] Bobrovsky, B.Z. ; Zakai, M. : " Asymptotic A Priori Estimates for the Error in the Nonlinear Filtering Problem " , IEEE, Trans. on Inf. Theory, Vol 28, March 1982

[12] Bobrovsky, B.Z. ; Geering, H.P. : " Simple and Efficient Linear and Nonlinear Filters by Regular Perturbation Methods " , Balakrishnan,A.V. and Thoma,M. (eds), Stochastic Differential Systems, Springer Verlag, 1982

[13] Saydy, L. : " Asymptotic Behavior Of Nonlinear Filters " M.S. Thesis, University Of MD At College Park, Aug. 1985.

[14] Gilman, A.S. ; Rhodes, I.B. : " Cone Bounded Nonlinearities and Mean Square Bounds - Estimation Upper Bounds " , IEEE A.C., Vol 18, #6, June 1973, pp 260-265

[15] Hille, E. : " Lecture notes on ODE's " , Addison Wesley, 1965.

[16] Blankenship, G.L. : " Some Approximation Methods in Nonlinear Filtering " , IEEE Control and Decision Proc., 1980, 51-56

[17] Blankenship, G.L. : " Asymptotic Analysis in Mathematical Physics and Control Theory : Some Problems with Common Features " , Ricerche di Automatica, Vol. 10, #2, Dec. 1979

[18] Arnold, L. : " Stochastic Differential Equations " , John Wiley & Sons, New York, 1974

[19] Gilman, A.S. ; Rhodes, I.B. : " Cone Bounded Nonlinearities and Mean Square Bounds - Estimation Lower Bounds " , IEEE A.C., 1976

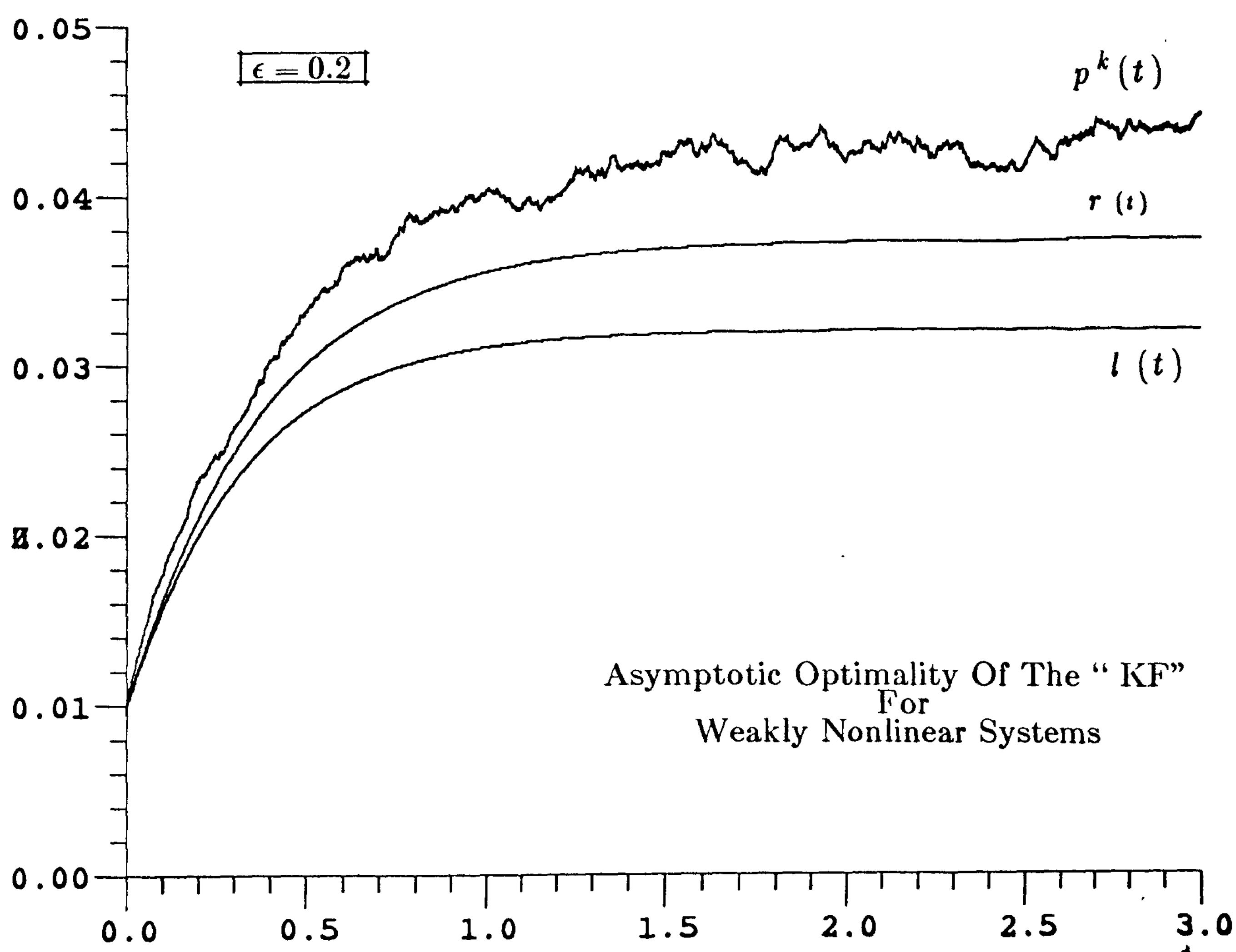

figure 1 (a) : "KF" performance

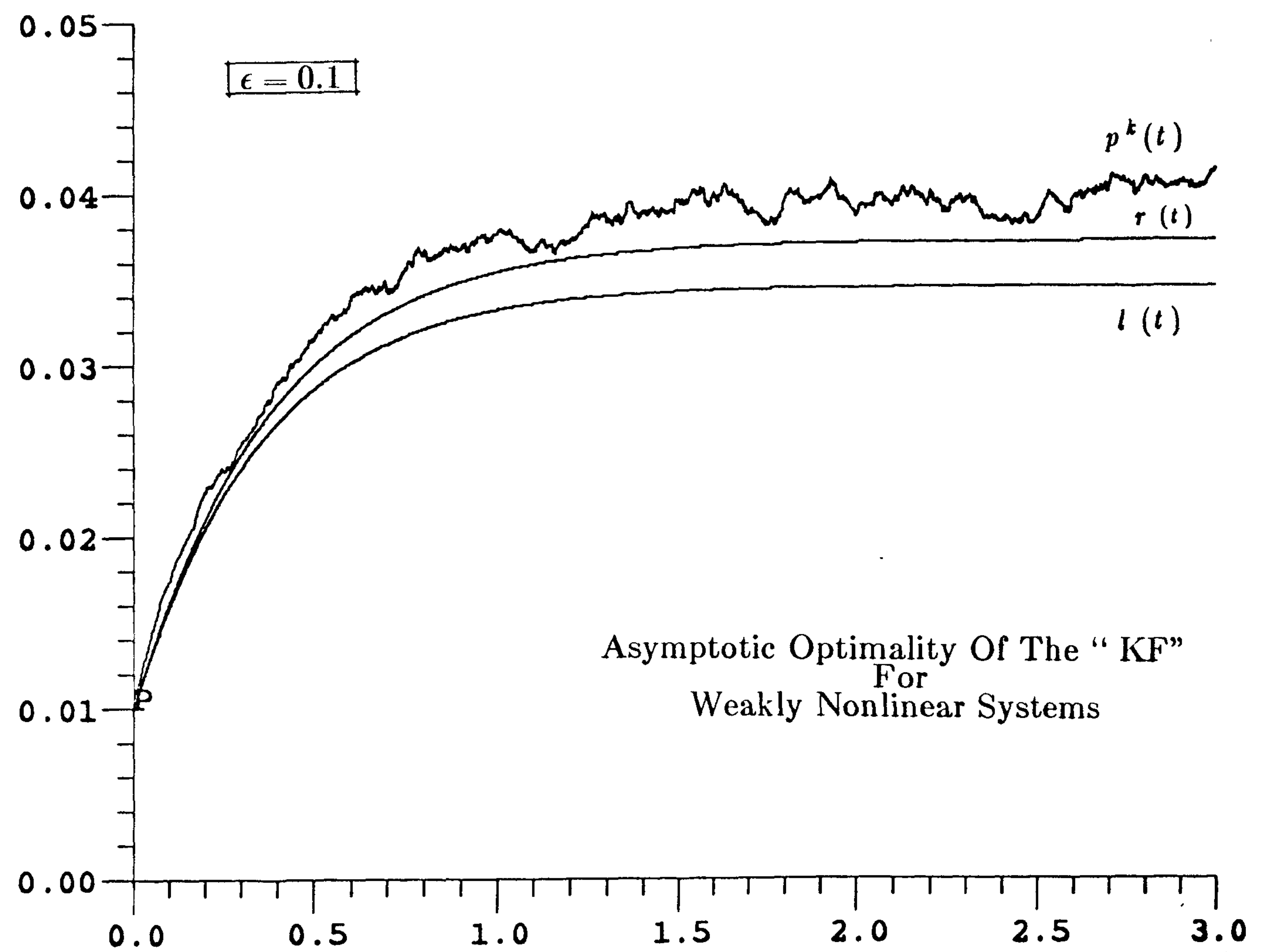

figure 1 (b) : " KF" performance

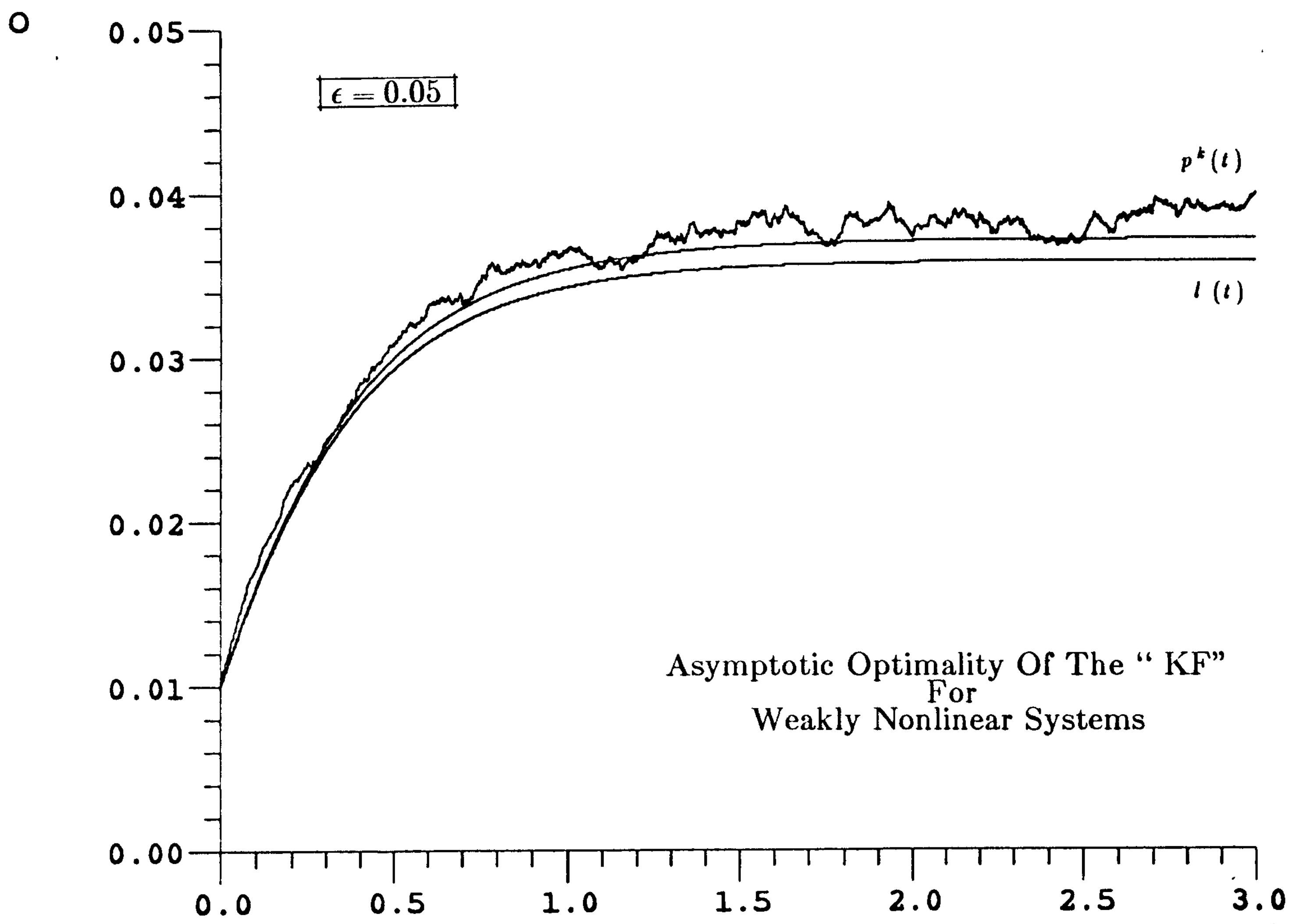

figure 1 (c) : " KF" performance

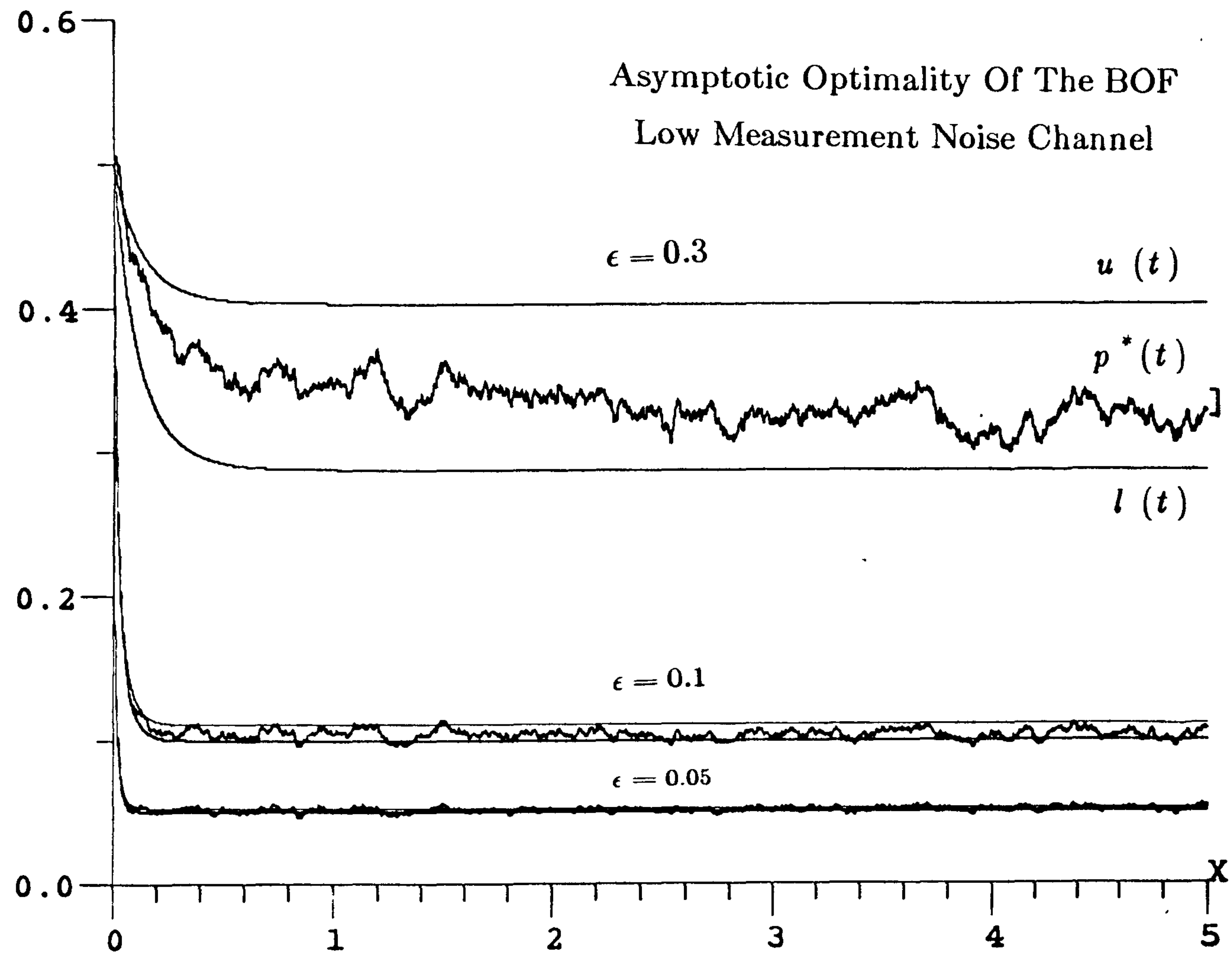

figure 2 (a) : BOF performance

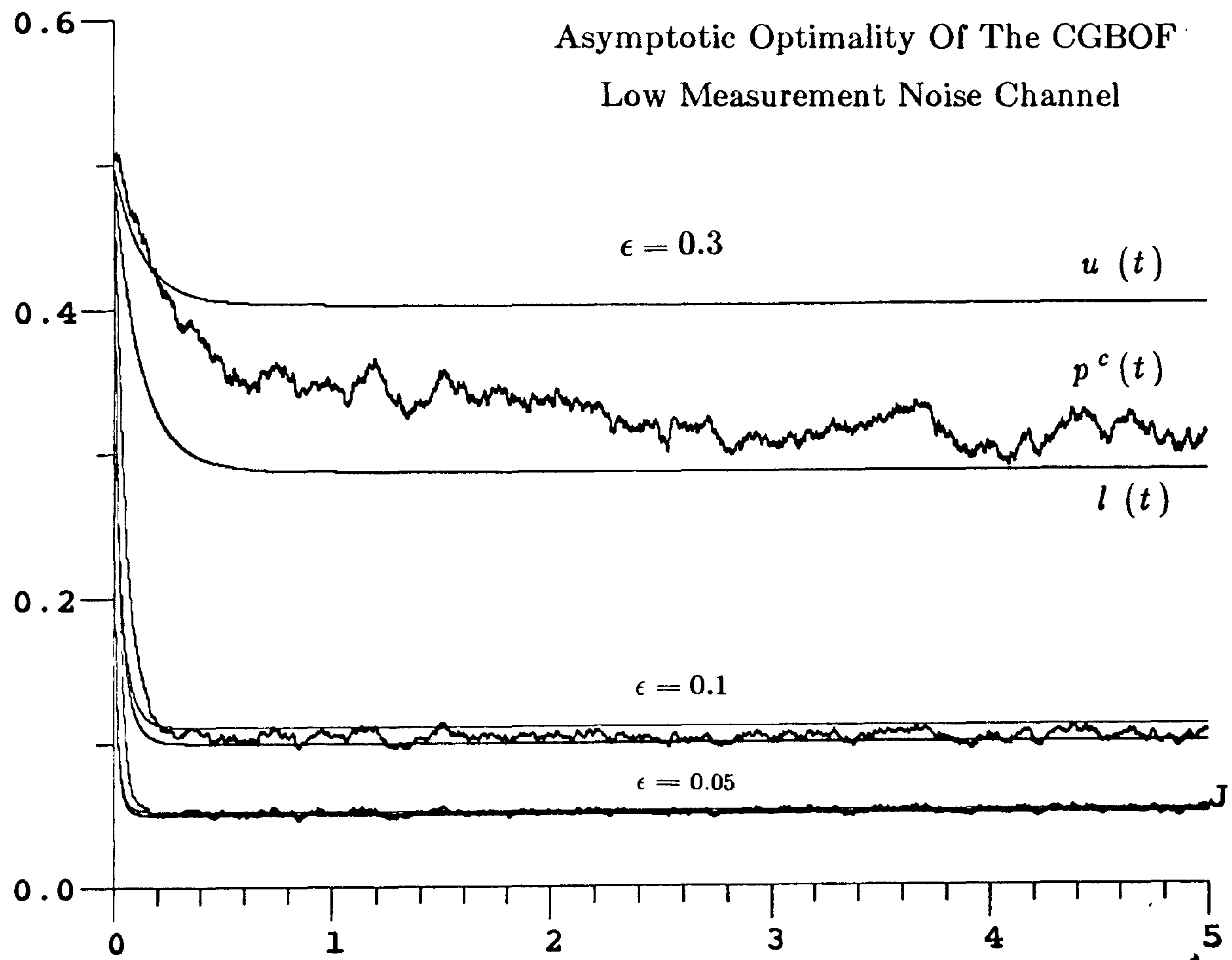

figure 2 (b) : CGBOF performance

```
      program  weak
c
c
c     USAGE :
c     =======
c
c     Program "weak" carries out Monte Carlo simulations
c     for one dimensional filtering problems of the form:
c
c     dxt = a xt dt + eps * f(xt) dt + sig*dwt , 0<=t<=ziT
c     dyt = c xt dt + rho*dvt
c     x0 ~ N ( xm0 , sig0 ** 2 )
c
c     where the nonlinearity has a bounded derivative :
c
c     xmuu <= fx(x) <= xmub
c
c     Program weak generates E [ ( xt - zt )**2 ] together with
c     the upper and lower bounds u(t) and l(t) on the optimal
c     MS error.
c     If ioption=1 then zt is the "KF" filtered estimate.
c     If ioption=2 then zt is the BOF filtered estimate.
c     When ioption=0, two sample paths for xt and zt are generated;
c     where zt is either from "KF" (iflag=1) or BOF (iflag=2).
c
c     INPUT DATA:
c     ===========
c
c     (i) iT, N:
c         ------
c         iT = time horizon
c         N = number of subdivisions in the time interval [ 0, ziT ]
c             (should be large enough in order for the discretized
c             stochastic differential to yield a good approximation).
c             N<=5000, unless the array dimensions are changed.
c     (ii) xm0,sig0,sig,a,c,rho,dseed0,dseed1,dseed2:
c          -------------------------------------------
c          xm0,sig0,sig,a,c,rho: parameters of the model
c          dseed0,dseed1,dseed2: initializations for the random number
c          generator. These could be any (distinct) numbers between 0
c          and 1.0e20, preferably as large as possible.
c     (iii) M, NS :
c           -------
c           M = number of values to be printed out.
c           NS = number of sample paths used to compute expectations.
c     (iv) ioption, iflag :  already described.
c          ---------------
c     (v) eps,xmuu,xmub : small parameter and derivative bounds.
c         ---------------
c
c     OUTPUT DATA:
c     ============
c
c     ioption = 0 :
c     -------------
c     An array of 2N values is generated. The first set of N numbers
c     corresponds to the (simulated) true state; i.e.:
c     x(i*dT), i=0,1,...,N-1, where dT = ziT/N.
c     The other N values are those of the filtered estimate zt (either
c     "KF" or BOF, depending on iflag).
```

```
c     ioption > 0 :
c     -------------
c     An array of 3N numbers is obtained the first N values of which
c     are those of p(t) = E [(xt - zt)**2] (zt being either "KF" or
c     BOF, depending on ioption), namely:
c     p(i*dT), i=0,1,....,N-1, where dT=ziT/N.
c     Similarily, the second and third set of values are those of u(t)
c     and l(t) respectively.
c     remark: in the case of the "KF", no computable upper bound exists.
c             Instead of u(t), the solution of the riccati equation
c             associated with the limiting linear system r(t) is printed
c
c     TIPS :
c     ======
c
c     (a) Progam weak uses the IMSLS library for random number
c         generation. E.g. low could be run as follows:
c         % f77 -o runlow low.f -limsls
c         % runlow <inputfile >outputfile
c         where inputfile is a file in which the data is prealably
c         stored.
c     (b) The nonlinearity f ( currently equal to tanh(x)) may be
c         changed by modifying fk accordingly in the subroutines
c         observy, kalfilt and bofilt.
c     (c) The quality of the simulation results depends strongly on
c         how large N and NS are. Typically, N=1000 and NS>=500.
c
c
      dimension er(5000),xx(5000),xxf(5000)
      dimension u(5000),x(4),dx(4)
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
c
      read *,iT,N
      read *,xm0,sig0,sig,a,c,rho,dseed0,dseed1,dseed2
      read *,M,NS
      read *,ioption,iflag
      read *,eps,xmuu,xmub
c
      deltat=1.0*iT/N
      sqd=sqrt(deltat)
c
      if (ioption.eq.0) go to 63
c
      do 50 i=1,N
      er(i)=0.
50    continue
c
      if (ioption.eq.2) go to 29
c
c     ioption=1 ---> N values of u(t) (=r(t) here) are computed
c     and used to compute the mmse for the "kf" applied to
c     the w.n.l filtering pb.
c
      kr=0
      kswitch=0
      call ric(kr,kswitch,u)
c
      do 60 j=1,NS
```

```
      call kfsub(dseed0,dseed1,dseed2,xx,xxf,u)
c
      do 70 k=1,N
      er(k)=er(k)+(xx(k)-xxf(k))**2
70    continue
c
60    continue
      go to 22
c
c     ioption=2 :
c     upper bound (N values:u(0)...u(iT)) are computed
c     and used to compute the BOF mmse error next
c
  29  kr=1
      kswitch=0
      call ric(kr,kswitch,u)
      do 37 j=1,NS
      call bofsub(dseed0,dseed1,dseed2,xx,xxf,u)
      do 38 k=1,N
      er(k)=er(k)+(xx(k)-xxf(k))**2
38    continue
37    continue
c
c     M (<=N) values of the mmse error are printed next
c
22    er(1)=sig0**2
      print *,er(1)
      do 80 k=2,M
      er(k)=er(k)/NS
      print *,er(k)
80    continue
      go to 67
c
c     ioption=0 :
c     two sample paths of the true and (*-) filtered state are
c     computed.
c     iflag=1 ----> kf-filtered  ;  iflag=2 ----> bof-filtered
c
63    if (iflag.eq.2) go to 119
      kr=0
      kswitch=0
      call ric(kr,kswitch,u)
      call kfsub(dseed0,dseed1,dseed2,xx,xxf,u)
      go to 121
c
119   kr=1
      kswitch=0
      call ric(kr,kswitch,u)
      call bofsub(dseed0,dseed1,dseed2,xx,xxf,u)
c
121   do 65 k=1,M
      print *,xx(k)
65    continue
c
      do 66 k=1,M
      print *,xxf(k)
66    continue
c
c
c     upper bound u(t) is printed next.
```

```
c     In the case of the "KF", i.e. ioption=1 this is
c     just r(t) which is neither an upper bound for
c     pk(t) nor for p(t).
c
67    do 135 i=1,N
 135  print *,u(i)
c
c     lower bound l(t) is first computed then printed
c
      kr=1
      kswitch=1
      call ric(kr,kswitch,u)
      do 136 i=1,N
  136 print *,u(i)
c
  137 stop
      end
c
c
c     *****************
c     SOUBROUTINE KFSUB
c     *****************
c
      subroutine kfsub(dseed0,dseed1,dseed2,xx,xxf,u)
c
      real xk,xfk,yk,yyk
      double precision dseed0,dseed1,dseed2
      dimension xx(5000),xxf(5000)
      dimension u(5000)
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
c
      do 10 k=1,N
      km1=k-1
      call kalfilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk,u)
      xx(k)=xk
      xxf(k)=xfk
10    continue
c
      return
      end

c     *******************
c     SUBROUTINE BOFSUB :
c     *******************
c
      subroutine bofsub(dseed0,dseed1,dseed2,xx,xxf,u)
c
      real xk,xfk,yk,yyk
      double precision dseed0,dseed1,dseed2
      dimension xx(5000),xxf(5000)
      dimension u(5000)
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
c
      do 10 k=1,N
      km1=k-1
      call bofilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk,u)
      xx(k)=xk
      xxf(k)=xfk
```

```
10    continue
c
      return
      end

c     ********************
c     SOUBROUTINE OBSERVY :
c     ********************


      subroutine observy(km1,dseed0,dseed1,dseed2,xk,yk)
c
c     ****************************************************
c     observy generates the observation yk=y(k*deltat)
c     and xk=x(k*deltat) from the model :
c     dx(t)=f(x(t)).dt + sig.dw(t) , x(0)=x0 N(m0,sig0^2)
c     dy(k)=g(x(t)).dt + rho.dv(t) , y(0)=0
c     w(t),v(t) standard N(0,t) , deltat=iT/N , sqd its sqrt
c     ggnqf(dseed) generates a N(0,1)-variate Zk(dseed) .
c     the value of dseed is internally changed by ggnqf for
c     a future call.
c     ****************************************************
c
      real xk,xfk,yk,yyk
      real ggnqf,Zk,Qk
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
      if(km1.gt.0) go to 98
      xk=sig0*ggnqf(dseed0)+xm0
      yk=0.
      go to 99
  98  Zk=ggnqf(dseed1)
      Qk=ggnqf(dseed2)
      fk=a*xk+eps*tanh(xk)
      gk=c*xk
      xkp1=xk+fk*deltat+sig*sqd*Zk
      ykp1=yk+gk*deltat+rho*sqd*Qk
      xk=xkp1
      yk=ykp1
  99  gfg=0.
      return
      end

c     ********************
c     SOUBROUTINE KALFILT :
c     ********************

      subroutine kalfilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk,u)
c
c     ****************************************************
c     Using observations from from the model in subroutine
c     observy this subroutine generates xfk=xf(k*deltat)
c     where xf(t) is the kalfilt (constant gain filter) :
c     dxf(t)=f(xf(t)).dt + sig/rho[dy(t) - c.xf(t).dt]
c     xf(0)=E(x0)=m0
c     kalfilt is asymptotically optimal as rho--->0, f cone
c     bounded and observations linear.
c      (kalfilt also returns the true state xk)
c     ****************************************************
```

```
c
      real xk,xfk,yk,yyk
      real ggnqf,Zk,Qk
      dimension u(5000)
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
      if(km1.gt.0) go to 78
      xfk=xm0
      yyk=0.
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      go to 79
   78 fk=a*xfk
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      yykp1=yk
      dyyk=yykp1-yyk
      gain=c*u(km1)/(rho**2)
      xfkp1=xfk+fk*deltat+gain*(dyyk-c*xfk*deltat)
      xfk=xfkp1
      yyk=yykp1
   79 return
      end
c
c
c     ******************
c     SUBROUTINE BOFILT:
c     ******************
c
      subroutine bofilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk,u)
c
c     ****************************************************
c     Using observations from from the model in subroutine
c     observy this subroutine generates xfk=xf(k*deltat)
c     where xf(t) is the kalfilt (constant gain filter) :
c     dxf(t)=f(xf(t)).dt + sig/rho[dy(t) - c.xf(t).dt]
c     xf(0)=E(x0)=m0
c     kalfilt is asymptotically optimal as rho--->0, f cone
c     bounded and observations linear.
c      (kalfilt also returns the true state xk)
c     *****************************************************
c
      dimension u(5000)
      real xk,xfk,yk,yyk
      real ggnqf,Zk,Qk
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
      if(km1.gt.0) go to 78
      xfk=xm0
      yyk=0.
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      go to 79
   78 fk=a*xfk+eps*tanh(xfk)
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      yykp1=yk
      dyyk=yykp1-yyk
c
      bofgain=c*u(km1)/(rho**2)
      xfkp1=xfk+fk*deltat+bofgain*(dyyk-c*xfk*deltat)
c
```

```
      xfk=xfkp1
      yyk=yykp1
   79 return
      end
c
c     ***************
c     SUBROUTINE RIC:
c     ***************
c
c     kr=kswitch=0 ---> r(t) is computed
c     kr=1 ; kswitch=0 -----> u(t) is computed
c     kr=1 ; kswitch=1 -----> l(t) is computed
c
      subroutine ric(kr,kswitch,u)
      dimension u(5000),x(4),dx(4)
      common /const/deltat,sqd,xm0,sig0,sig,c,rho,N,iT
      common /param/a,eps,xmuu,xmub
      h=deltat
      deltmu=(xmub-xmuu)/2
      if(kswitch.eq.1) goto 17
      p1=xmub
      go to 18
   17 p1=xmuu
   18 s2=sig**2
      c2=c**2
      r2=rho**2
      quant=4*r2*((eps*deltmu)**2)/s2
      wc2=c2+kswitch*quant
      nn=1
      x(1)=sig0**2
      u(1)=x(1)
      t=0.0
      k=0
      m=0
c     write the ode
    1 dx(1)=s2+2.0*(a+kr*eps*p1)*x(1)-wc2*(x(1)**2)/r2
      call runta(nn,k,ii,x,dx,t,h)
      go to (1,2),ii
    2 m=m+1
      u(m+1)=x(1)
      if (t.le.iT) go to 1
      return
      end

       subroutine runta(nn,k,ii,x,dx,t,h)
       dimension y(4),z(4),x(4),dx(4)
       k=k+1
       go to (1,2,3,4,5),k
     2 do 10 j=1,nn
       z(j)=dx(j)
       y(j)=x(j)
    10 x(j)=y(j)+0.5*h*dx(j)
    25 t=t+0.5*h
     1 ii=1
       return
     3 do 15 j=1,nn
       z(j)=z(j)+2.0*dx(j)
    15 x(j)=y(j)+0.5*h*dx(j)
       ii=1
       return
```

```
    4 do 20 j=1,nn
      z(j)=z(j)+2.0*dx(j)
   20 x(j)=y(j)+h*dx(j)
      go to 25
    5 do 30 j=1,nn
   30 x(j)=y(j)+(z(j)+dx(j))*h/6.0
      ii=2
      k=0
      return
      end
```

```
      program low
c
c
c     USAGE :
c     =======
c
c     Program "low" carries out Monte Carlo simulations
c     for one dimensional filtering problems of the form:
c
c     dxt = f(xt) dt + sig*dwt    ,    0<=t<=ziT
c     dyt = c xt dt + eps*dvt
c     x0 ~ N ( xm0 , sig0 ** 2 )
c
c     where f has bounded derivatives : alphau<=fx(x)<=alphab.
c     Program low generates E [ ( xt - zt )**2 ] together with
c     the upper and lower bounds u(t) and l(t) on the optimal
c     MS error.
c     If ioption=1 then zt is the CGBOF filtered estimate.
c     If ioption=2 then zt is the BOF filtered estimate.
c     When ioption=0, two sample paths for xt and zt are generated;
c     where zt is either from CGBOF (iflag=1) or BOF (iflag=2).
c
c     INPUT DATA:
c     ===========
c
c     (i) ziT, N:
c         -------
c         ziT = time horizon
c         N = number of subdivisions in the time interval [ 0, ziT ]
c             (should be large enough in order for the discretized
c             stochastic differential to yield a good approximation).
c             N<=5000, unless the array dimensions are changed.
c     (ii) alphab, alphau: upper and lower bounds on the derivative.
c          --------------
c     (iii) xm0,sig0,sig,c,eps,dseed0,dseed1,dseed2:
c           ----------------------------------------
c           xm0,sig0,sig,c,eps: parameters of the model
c           dseed0,dseed1,dseed2: initializations for the random number
c           generator. These could be any (distinct) numbers between 0
c           and 1.0e20, preferably as large as possible.
c     (iv) M, NS :
c          -------
c          M = number of values to be printed out.
c          NS = number of sample paths used to compute expectations.
c     (v) ioption, iflag :  already described.
c         ----------------
c
c     OUTPUT DATA:
c     ============
c
c     ioption = 0 :
c     -------------
c     An array of 2N values is generated. The first set of N numbers
c     corresponds to the (simulated) true state; i.e.:
c     x(i*dT), i=0,1,...,N-1, where dT = ziT/N.
c     The other N values are those of the filtered estimate zt (either
c     CGBOF or BOF, depending on iflag).
c     ioption > 0 :
c     -------------
c     An array of 3N numbers is obtained the first N values of which
```

```
c       are those of p(t) = E [(xt - zt)**2] (zt being either CGBOF or
c       BOF, depending on ioption), namely:
c       p(i*dT), i=0,1,....,N-1, where dT=ziT/N.
c       Similarily, the second and third set of values are those of u(t)
c       and l(t) respectively.
c
c       TIPS :
c       ======
c
c        (a) Progam low uses the imsls library for random number
c            generation. E.g. low could be run as follows:
c            % f77 -o runlow low.f -limsls
c            % runlow <inputfile >outputfile
c            where inputfile is a file in which the data is prealably
c            stored.
c        (b) The nonlinearity f (by default equal to atan(x)) may be
c            changed by modifying fk in the subroutines observy, cgfilt
c            and bofilt.
c        (c) The quality of the simulation results depends strongly on
c            how large N and NS are. Typically, N=1000 and NS>=500.
c
c
        dimension er(5000),xx(5000),xxf(5000)
        dimension u(5000),x(4),dx(4)
        double precision dseed0,dseed1,dseed2
        common /const/deltat,sqd,xm0,sig0,sig,c,eps,N ,ziT,alphab
c
        read *,ziT,N
        read *,alphab,alphau
        read *,xm0,sig0,sig,c,eps,dseed0,dseed1,dseed2
        read *,M,NS
        read *,ioption,iflag
c       ioption: =1 ---> cgbof    =2 ----> bof (ms-errors)
c       iflag:  =1 ----> cgbof    =2 ----> bot (sample paths)
c
        deltat=1.0*ziT/N
        sqd=sqrt(deltat)
c
c
        if (ioption.eq.0) go to 63
c
        do 50 i=1,N
        er(i)=0.
50      continue
c
        call ric(u)
        if (ioption.eq.2) go to 29
c
        do 60 j=1,NS
        call cgsub(dseed0,dseed1,dseed2,xx,xxf)
c
        do 70 k=1,N
        er(k)=er(k)+(xx(k)-xxf(k))**2
70      continue
c
60      continue
        go to 22
c
c       ioption=2 :
c       upper bound (N values:u(0)...u(ziT)) are computed
```

```
c     and used to compute the BOF mmse error next
c
29    do 37 j=1,NS
      call bofsub(dseed0,dseed1,dseed2,xx,xxf,u)
      do 38 k=1,N
      er(k)=er(k)+(xx(k)-xxf(k))**2
38    continue
37    continue
c
c     M (<=N) values of the mmse error are printed next
c
22    er(1)=sig0**2
      print *,er(1)
      do 80 k=2,M
      er(k)=er(k)/NS
      print *,er(k)
80    continue
c
c     the upper bound values u(k)'s are printed.
c     remember it known that this upper bounds bof
c     but not necessarily cgbof.
c
      do 81 k=1,M
81    print *,u(k)
c
c     the lower bound is printed next by giving new
c     values to ric-subroutine.
c
      dalpha=(alphab-alphau)/2
      alphab=alphau
      tempo=c**2+(2*eps*dalpha/sig)**2
      c=sqrt(tempo)
c     next u is really l
      call ric(u)
      do 82 k=1,M
82    print *,u(k)
      go to 67
c
c     ioption=0 :
c     two sample paths of the true and (cg-) filtered state are
c     computed
c
63    if (iflag.eq.2) go to 181
      call cgsub(dseed0,dseed1,dseed2,xx,xxf)
      go to 182
181   call ric(u)
      call bofsub(dseed0,dseed1,dseed2,xx,xxf,u)
c
182   do 65 k=1,M
      print *,xx(k)
65    continue
c
      do 66 k=1,M
      print *,xxf(k)
66    continue
c
67    stop
      end
c
c
```

```
c     SOUBROUTINE CGSUB
c
      subroutine cgsub(dseed0,dseed1,dseed2,xx,xxf)
c
      real xk,xfk,yk,yyk
      double precision dseed0,dseed1,dseed2
      dimension xx(5000),xxf(5000)
      common /const/deltat,sqd,xm0,sig0,sig,c,eps,N,ziT,alphab
c
      do 10 k=1,N
      km1=k-1
      call cgfilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk)
      xx(k)=xk
      xxf(k)=xfk
10    continue
c
      return
      end

c     ******************
c     SUBROUTINE BOFSUB :
c     ******************
c
      subroutine bofsub(dseed0,dseed1,dseed2,xx,xxf,u)
c
      real xk,xfk,yk,yyk
      double precision dseed0,dseed1,dseed2
      dimension xx(5000),xxf(5000)
      dimension u(5000)
      common /const/deltat,sqd,xm0,sig0,sig,c,eps,N,ziT,alphab
c
      do 10 k=1,N
      km1=k-1
      call bofilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk,u)
      xx(k)=xk
      xxf(k)=xfk
10    continue
c
      return
      end

c     SOUBROUTINE OBSERVY :


      subroutine observy(km1,dseed0,dseed1,dseed2,xk,yk)
c
c     *****************************************************
c     observy generates the observation yk=y(k*deltat)
c     and xk=x(k*deltat) from the model :
c     dx(t)=f(x(t)).dt + sig.dw(t) , x(0)=x0 N(m0,sig0^2)
c     dy(k)=g(x(t)).dt + eps.dv(t) , y(0)=0
c     w(t),v(t) standard N(0,t) , deltat=ziT/N , sqd its sqrt
c     ggnqf(dseed) generates a N(0,1)-variate Zk(dseed) .
c     the value of dseed is internally changed by ggnqf for
c     a future call.
c     *****************************************************
c
      real xk,xfk,yk,yyk
      real ggnqf,Zk,Qk
```

```
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,eps,N,ziT,alphab
      if(km1.gt.0) go to 98
      xk=sig0*ggnqf(dseed0)+xm0
      yk=0.
      go to 99
   98 Zk=ggnqf(dseed1)
      Qk=ggnqf(dseed2)
      fk=atan(xk)
      gk=c*xk
      xkp1=xk+fk*deltat+sig*sqd*Zk
      ykp1=yk+gk*deltat+eps*sqd*Qk
      xk=xkp1
      yk=ykp1
   99 gfg=0.
      return
      end


c     SOUBROUTINE CGFILT :


      subroutine cgfilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk)
c
c     ***************************************************
c     Using observations from from the model in subroutine
c     observy this subroutine generates xfk=xf(k*deltat)
c     where xf(t) is the cgfilt (constant gain filter) :
c     dxf(t)=f(xf(t)).dt + sig/eps[dy(t) - c.xf(t).dt]
c     xf(0)=E(x0)=m0
c     cgfilt is asymptotically optimal as eps--->0, f cone
c     bounded and observations linear.
c     (cgfilt also returns the true state xk)
c     ****************************************************
c
      real xk,xfk,yk,yyk
      real ggnqf,Zk,Qk
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,eps,N,ziT,alphab
      if(km1.gt.0) go to 78
      xfk=xm0
      yyk=0.
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      go to 79
   78 fk=atan(xfk)
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      yykp1=yk
      dyyk=yykp1-yyk
      xfkp1=xfk+fk*deltat+(sig/eps)*(dyyk-c*xfk*deltat)
      xfk=xfkp1
      yyk=yykp1
   79 return
      end
c
c
c     ******************
c     SUBROUTINE BOFILT:
c     ******************
c
      subroutine bofilt(km1,dseed0,dseed1,dseed2,xk,xfk,yyk,u)
```

```
c
c     **************************************************
c     Using observations from from the model in subroutine
c     observy this subroutine generates xfk=xf(k*deltat)
c     where xf(t) is the cgfilt (constant gain filter) :
c     dxf(t)=f(xf(t)).dt + sig/eps[dy(t) - c.xf(t).dt]
c     xf(0)=E(x0)=m0
c     cgfilt is asymptotically optimal as eps--->0, f cone
c     bounded and observations linear.
c     (cgfilt also returns the true state xk)
c     ***************************************************
c
      dimension u(5000)
      real xk,xfk,yk,yyk
      real ggnqf,Zk,Qk
      double precision dseed0,dseed1,dseed2
      common /const/deltat,sqd,xm0,sig0,sig,c,eps,N,ziT,alphab
      if(km1.gt.0) go to 78
      xfk=xm0
      yyk=0.
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      go to 79
   78 fk=atan(xfk)
      call observy(km1,dseed0,dseed1,dseed2,xk,yk)
      yykp1=yk
      dyyk=yykp1-yyk
c
      bofgain=c*u(km1)/(eps**2)
      xfkp1=xfk+fk*deltat+bofgain*(dyyk-c*xfk*deltat)
c
      xfk=xfkp1
      yyk=yykp1
   79 return
      end
c
c     ***************
c     SUBROUTINE RIC:
c     ***************
c
      subroutine ric(u)
      dimension u(5000),x(4),dx(4)
      common /const/deltat,sqd,xm0,sig0,sig,c,eps,N,ziT,alphab
      h=deltat
      nn=1
      x(1)=sig0**2
      u(1)=x(1)
      t=0.0
      k=0
      m=0
c     write the ode
    1 dx(1) = sig**2 + 2.0*alphab*x(1) - (c**2)*(x(1)**2)/(eps**2)
      call runta(nn,k,ii,x,dx,t,h)
      go to (1,2),ii
    2 m=m+1
      u(m+1)=x(1)
      if (t.le.ziT) go to 1
      return
      end

       subroutine runta(nn,k,ii,x,dx,t,h)
```

```
      dimension y(4),z(4),x(4),dx(4)
      k=k+1
      go to (1,2,3,4,5),k
 2    do 10 j=1,nn
      z(j)=dx(j)
      y(j)=x(j)
10    x(j)=y(j)+0.5*h*dx(j)
25    t=t+0.5*h
 1    ii=1
      return
 3    do 15 j=1,nn
      z(j)=z(j)+2.0*dx(j)
15    x(j)=y(j)+0.5*h*dx(j)
      ii=1
      return
 4    do 20 j=1,nn
      z(j)=z(j)+2.0*dx(j)
20    x(j)=y(j)+h*dx(j)
      go to 25
 5    do 30 j=1,nn
30    x(j)=y(j)+(z(j)+dx(j))*h/6.0
      ii=2
      k=0
      return
      end
```