Tracking of a Brownian phase with linear drift : algorithms and performances.

Pierre-Olivier Amblard[1], Jean-Marc Brossier[2], Eric Moisan[1]
[1]Laboratoire des Images et des Signaux, UMR CNRS 5083
Groupe Non Linéaire,
ENSIEG, BP 46, 38402 Saint-Martin d'Hères cedex, France
[2]ST-Microelectronics
12, rue Jules Horowitz, BP 217, 38019 Grenoble cedex
Bidou.Amblard@lis.inpg.fr, Jean-Marc.Brossier@st.com, Eric.Moisan@lis.inpg.fr

ABSTRACT

We study the problem of tracking a Brownian phase with linear drift at the output of a simple digital receiver. The classical Costas and Decision feedback loops are compared to the particle implementation of the optimal nonlinear filter. The study of the asymptotic performance of the loops allows to tune the loops in order to minimize the mean square error. We then show that the particle filter outperforms the loops in the acquisition step and in the behavior in front of cycle slips.

1 Phase tracking

We consider here the problem of phase synchronisation in digital communications [1, 8]. We study the ideal case for which the matched filtering have been performed; the symbol synchronisation is perfect and intersymbol interference is absent. The only problem to deal with is the elimination of a residual phase. The output of the digital receiver may then be written as

$$y_k = a_k e^{i\xi_k} + n_k \quad (1)$$

where sequence $\{n_k\}$ is a sequence of independent, identically distributed (i.i.d.) complex valued random variables, of power σ_n^2. This noise is furthermore assumed to be Gaussian and circular (real and imaginary parts are independent). The sequence a_k is also an i.i.d. sequence and is the sequence of symbols that code the emitted message. In this work, the digital modulation used is a PSK modulation scheme. However, results presented in the paper are restricted to the Binary PSK: each a_k takes the value ± 1 with probability 1/2.
The residual phase ξ_k is the result of a bad synchronisation between the clocks of the emitter and the receiver. We suppose that the oscillators differ from a constant shift in frequency and from a small random jitter. The phase can therefore be considered as a Brownian motion with a linear drift, or

$$\xi_k = \xi_{k-1} + \varepsilon + w_k \quad (2)$$

where w_k is a Gaussian white noise with zero mean and variance σ_w^2; the slope ε is furthermore unknown. We finally assume that sequences a_k, ξ_k and n_k are statistically independent.
The problem considered here is the estimation of the phase. Furthermore, we study the in-line case, and our problem is thus a tracking problem. In the sequel, we present three algorithms: two second order phase-locked loops (PLL) and the particle implementation of the optimal filter (particle filter). Section 3 is dedicated to a comparative study of the performance of the algorithms. Asymptotic analysis for the PLLs is performed theoretically, and this allows to put the loops in their optimal asymptotic regime (minimization of the mean square error). We then compare numerically the loops to the particle filter. In a second step, we perform numerical investigation of the transient phase of the algorithms.

2 Algorithms

Phase-locked loops The PLLs studied here are second-order adaptive algorithms or multistep algorithms [2]. The loops are made of two coupled adaptive algorithms, one delivering the estimated phase, the other the estimated slope. The generic form of these PLLs is

$$\begin{aligned} \hat{\xi}_k &= \hat{\xi}_{k-1} + \varepsilon_{k-1} + \gamma_1 \chi_k \\ \varepsilon_k &= \varepsilon_{k-1} + \gamma_2 \chi_k \end{aligned}$$

The usual structure of adaptive algorithms clearly appears: correction of the last estimation by adding a term proportional to a kind of error. In the case of BPSK ($a_k = \pm 1$), error terms write

$$\begin{aligned} \chi_k^C &= \mathrm{Im}(y_k^2 e^{-2i(\hat{\xi}_{k-1}+\varepsilon_{k-1})}) \\ \chi_k^D &= \mathrm{Im}(y_k e^{-i(\hat{\xi}_{k-1}+\varepsilon_{k-1})}) \mathrm{Sign}\left(\mathrm{Re}(y_k e^{-i(\hat{\xi}_{k-1}+\varepsilon_{k-1})})\right) \end{aligned}$$

and correspond respectively to the Costas loop and to the Decision Feedback Loop. The nonlinearity in these terms allows to remove the digital modulation; the residual phase is then compensated by the past estimate. Therefore, the χs are more or less the sine of the phase error.
Finally, γ_1 and γ_2 are the stepsizes of the algorithms and rule not only the transient behavior of the algorithms,

but also their asymptotic behavior. The choice of these parameters is thus a crucial point, and is furthermore linked to the performance of the loops.

Particle filter The problem to solve is the estimation of the phase given observations and an *a priori* on the phase. We are thus in the framework of optimal filtering of an hidden Markov model. The solution of optimal filtering in a Bayesian context has been known since the sixties [3]; it consists in a recursion equation for the *a posteriori* density $p(\xi_k|\boldsymbol{y}_{1:k})$ (where $\boldsymbol{y}_{1:k} = (y_1, \ldots, y_k)$). This recursion

$$p\left(\xi_k|\boldsymbol{y}_{1:k}\right) = \frac{p(y_k|\xi_k)\int p(\xi_k|\xi_{k-1})p\left(\xi_{k-1}|\boldsymbol{y}_{1:k-1}\right)d\xi_{k-1}}{p\left(y_k|\boldsymbol{y}_{1:k-1}\right)} \tag{3}$$

is cut into a prediction step that uses the state equation of the phase, and a prediction step that take into account the new observation through the likelihood. This fundamental result is unfortunately unusable in practice, since the integrals in (3) are impossible to evaluate, as well as estimators derived from the *a posteriori* density. Numerical technique are therefore needed if one wants to implement the optimal filter. Among numerical techniques, Monte-Carlo simulation methods have seen an increased interest in the last years. They lead to the particle implementation of the optimal filter or particle filter for short. For a precise presentation, we refer the reader to [4, 5, 6, 7].

The idea is to consider the *a posteriori* density as a distribution function of a gaz of particles. The density is then replaced by an ensemble of particles (random variables) of the gaz. The mean values of the density can then be approximated by empiric means evaluated with the particles. Of course, the dynamics of the particles is ruled out in order to respect the *a posteriori* density. In particular, their moves follow the principle of recursion (3) : predictive move using the *a priori* –state equation–, correction step using the likelihood –observation equation–. For algorithmic reasons, these steps are coded differently. The prediction step is stored in the trajectory of the particles (snapshots of the state equation); the likelihood is stored in a weight attached to the particle. Note that the weights are normalized so that their sum over the ensemble of particles is 1.

The particle follow a first-order Markovian model; hence they diffuse, and their likelihood (weight) must decreased with time. The particle filter thus degenerates with time, since all the particle (but one which is the most likely) see their weight going to zero. To overcome this degeneracy, a resampling must be done regularly, according to the weights the particules have. In this step the least likely particles are likely to die whereas the most likely particles are likely to proliferate.

The particles for the phase tracking problem have two spatial dimensions : $\boldsymbol{x}_k(i) = (x_k^1(i), x_k^2(i))$. The first dimension corresponds to the phase to estimate, and the second dimension is dedicated to the slope ε. The particle filter used here is then summed up by the following algorithm:

1. `initialization :` N `particles` $\boldsymbol{x}_0(i) \sim p(\xi_0, \varepsilon) = U(0,\pi) \times U(-\varepsilon_{\max}, +\varepsilon_{\max})$ `with uniform weights` $\tilde{w}_0(i) = \frac{1}{N}$.
2. `For` $k \geq 1$ `do` $\forall i = 1, \ldots, N$:
   (a) `prediction:` $\boldsymbol{x}_{0:k}(i) = (\boldsymbol{x}_{0:k-1}(i), \boldsymbol{x}_k(i))$ `where`
   $$\boldsymbol{x}_k(i) \sim p(\xi_k|\boldsymbol{x}_{k-1}(i))$$
   (b) `correction:`
   $$\begin{aligned} w_k(i) &= \tilde{w}_{k-1}(i)p(y_k|\boldsymbol{x}_k(i)) \\ \tilde{w}_k(i) &= \frac{w_k(i)}{\sum_{j=1}^N w_k(j)} \end{aligned}$$
   (c) `If` $-\sum_i \tilde{w}_k(i)\log_2\tilde{w}_k(i) < \eta$ `resample according to` $\tilde{w}_k(i)$`, and then` $\tilde{w}_k(i) = 1/N$
   (d) `estimation:`
   $$\hat{\xi_k} = \hat{E}\left[\xi_k|\boldsymbol{y}_{1:k}\right] = \sum_{i=1}^N \tilde{w}_k(i)x_k^1(i)$$

The particle filter presented here is one of the simplest (also known as the bootstrap filter [7]). Other more elaborated versions (*e.g.* approximation of the optimal importance function) have been tested on our problem: the ameliorations provided by these versions where not important enough compared to the great increase in complexity to justify their use.

However some tricks are used that are not described in the algorithm. Among them, the most important concerns the resampling. The discrete probability defined by the weights of the particle can be seen as an approximation of a continuous density which we have to sample from. Since the approximation is very poor, the quality of the sampling can be increased by using kernel estimates [9]. The implementation can be done by adding to the variable generated from the discrete law a small perturbation drawn from the kernel used in the kernel approximation. This is particularly important for the constant parameter ε, since the evolution of the state equation $\varepsilon_k = \varepsilon_{k-1}$ is fixed by the initial condition, and do not allow a correct exploration of the state space. A random perturbation of the trajectory allows the correct exploration. We thus add to ε a Gaussian variable of variance $1/N$ at each resampling. This is not the optimal way of doing (see [9]) but this works very efficiently. Finally, note that the choice of the threshold η is not a crucial point; if η is too large, resampling occurs

too often and results in an increase of the complexity unbalanced by the low increase in performance; if η is too small, performance are poor. We typically use $\eta = 0.5 \log_2 N$.

3 Performance

Asymptotic MSE Asymptotic performance of the loops are difficult to obtain because of the nonlinearity of the algorithms. Furthermore, they are approximately known in the case of first order loops [12], but unknown for the second-order case, to the best of our knowledge. By considering the small errors approximation, correct after convergence and under the assumption of small noises, a linearization may be performed and allows to calculate the asymptotic performance. The details of the calculations are presented in [10], and we present now the main conclusions that can be drawn.

It is easy to show that the loops are convergent in the mean, and this for a wide range of the stepsizes (γ_1, γ_2). The evaluation of the asymptotic mean square error (MSE) as a function of γ_1 and γ_2 is long and requires the use of a symbolic calculator. The first conclusion is: for a fixed γ_1, the MSE is minimal when γ_2 goes to zero. This result is logical since γ_2 is the stepsize of an algorithm that estimate a constant parameter. Secondly, for a fixed γ_2, we show that the MSE is minimized for a no zero value $\gamma_1^\star$ of the stepsize γ_1. In particular, it can be shown for $\gamma_2 \to 0$ that

$$\gamma_{1,D}^\star = \frac{-\sigma_w^2 + \sigma_w \sqrt{\sigma_w^2 (1-2\phi)^2 + 2\phi^2 \sigma_n^2}}{2\sigma_w^2(\phi - 1) + \phi \sigma_n^2}$$

$$\gamma_{1,C}^\star = \frac{-\sigma_w^2 + \sigma_w \sqrt{\sigma_w^2 + 2\sigma_n^2 + \sigma_n^4}}{2\sigma_n^2 + \sigma_n^4}$$

where $\phi = 2 \int_0^{1/\sigma_n} \exp(-u^2) du / \sqrt{\pi}$. In the sequel, the loops are used with these values of the parameters, so that the asymptotic MSE is minimal (note that γ_2 cannot be chosen equal to 0; therefore it is set to a small value, typically 10^{-3}.)

Figure 1 depicts the asymptotic minimal MSE evaluated theoretically (dashed lines), numerically (circles and squares) for the loops as a function of the observation noise power (20 snapshots of 5000 samples each are used; we measured the mean square of the sinus of the error by averaging from sample 3000 to sample 5000, and then by averaging over snapshots). We also draw the Cramér-Rao bound for the problem at hand (the calculation has been performed using the recent result of [11]). The behavior of the loops is very good since they almost reach the bound.

Furthermore, we notice that the small error approximation leads to satisfactory results for a wide range of observation noise power. This is especially true for the DFL. In the case of the Costas loop, approximations are poor as soon as $\sigma_n = 0.4$ or 0.5. Finally, note the superiority of the DFL on the Costas loop : the DFL will be the only loop considered in the sequel.

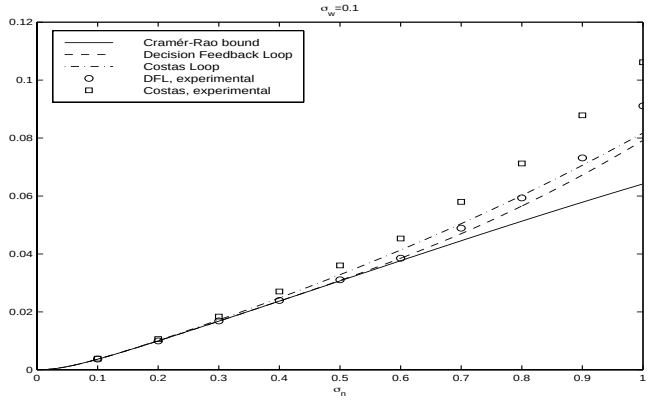

Figure 1: Asymptotic Minimum Mean Square Error and Cramér-Rao bound for the phase. Circles and squares correspond to an experimental evaluation of the performance.

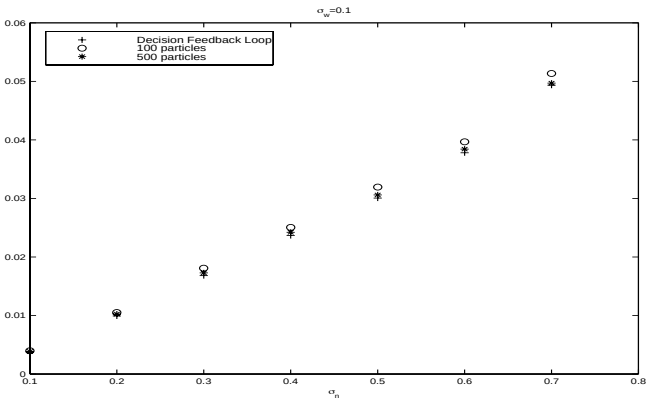

Figure 2: Asymptotic Mean Square Error for the DFL and for the particle filter.

Figure 2 shows the asymptotic MSE evaluated numerically for the DFL and for the particle filter (100 and 500 particles). This figure shows that the "optimal" configuration of the DFL is equivalent asymptotically to the optimal filter.

Acquisition time The second step in assessing performance consists in the study of the acquisition time or convergence time. The theoretical study of the convergence of the loops is difficult since the ODE (Ordinary Differential Equation) associated to it [2] is nonlinear and two-dimensional. We thus studied acquisition using Monte-Carlo simulations.

The acquisition time may be defined as the smallest time at which the algorithm enters and stays in a small interval center around the true solution. The width of this interval in our simulations has been set to 5% of the value of the solution. We then evaluate the acquisition time for 2000 snapshots and calculate an histogram. This has been performed for $\sigma_w = 0.1$, $\varepsilon = 0.5$ and two values of the observation noise power : $\sigma_n = 0.5$ and 1.

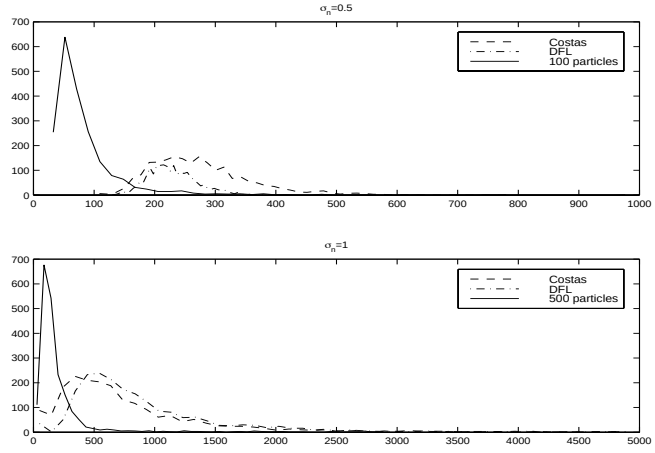

Figure 3: Histograms of acquisition time, for two values of the observation noise power.

Results are depicted in figure (3) and clearly show the superiority of the particle filter. Indeed, acquisition time is much lower for the particle filter than for the loops (note however that for the difficult case $\sigma_n = 1$, the number of particles is important). To understand the superiority of particle filtering, recall that the loops are optimized in order to minimize their asymptotic MSE. But, it is well-known that constant stepsize algorithms suffer from the "convergence speed-asymptotic MSE" compromise. However, this compromise does not exist for the optimal filter: it optimizes jointly its speed of convergence and its asymptotic MSE.

Cycle slips The final point in the performance analysis is the determination of the probability of cycle slips. Recall that we estimate a phase, a parameter that is defined *modulo* $2\pi/k$, k being the number of states of the PSK used (2 here). A cycle slip of the algorithm corresponds to a jump between two determinations of the phase. These slips lead to catastrophic consequences in the reception quality : for binary PSK, a slip of one cycle transforms a ± 1 in a ∓ 1! The numerical study of cycle slips remains to be done (for realistic conditions, the numerical evaluation of the probability of cycle slips requires incredible time of computer calculation), but experience shows that the particle filter is much more robust than the loops in front of cycle slips.

4 To conclude

To end this paper, we recall the main points of this work. The loops can be placed in their optimal asymptotic configuration, and in that case, they are asymptotically equivalent to the optimal filter implemented using particle filtering. The superiority of the particle filter is in the acquisition time and in its robustness in front of cycle slips. But, the price to get this superiority is of course the complexity of the algorithm. But we can bet that future chips will be able to support that complexity. A possibility to get simpler algorithms is to jointly use the particle filtering concept and adaptive algorithms: we have tried to run 10 DFLs in parallel, each one issuing from a different initial condition, and choosing at each time the more likely loop. The results are very efficient and almost equivalent to the particle filter.
Finally, note that we work on the introduction of abrupt changes in the model. An abrupt change can easily be handled by the particle filter by using Poisson processes for example. This is more difficult in the case of adaptive algorithms since it requires the coupling of the loops and sequential algorithms for detecting abrupt changes.

References

[1] J.M. BROSSIER, Signal et communications numériques. Egalisation et synchronisation, Hermès, 1997.

[2] A. BENVENISTE, M. METIVIER, R. PRIOURET, Adaptive algorithms and stochastic approximations, Springer Verlag, 1990.

[3] A. JAZWINSKI, Stochastic processes and filtering theory, Academic Press, 1970.

[4] N. GORDON, D. SALMOND, A. SMITH, Novel approach to nonlinear/nonGaussian Bayesian state estimation, IEE-proceedings F 140 (2) (1993) 107–113.

[5] P. D. MORAL, J. NOYER, G. RIGAL, G. SALUT, Résolution particulaire et traitement non-linéaire du signal: Applications radar/sonar, Traitement du Signal 12 (4) (1995) 287–301.

[6] A. DOUCET, S. GODSILL, C. ANDRIEU, On sequential Monte Carlo sampling methods for Bayesian filtering, Statist. Comp. (10) (2000) 197–208.

[7] A. DOUCET, N. D. FREITAS, N. GORDON, Sequential Monte-Carlo methods in practice, Springer, 2001.

[8] H. MEYR and M. MOENECLAEY and S.A. FECHTEL, Digital communications receivers : synchronisation, channel estimation and signal processing, John Wiley & Sons,1998.

[9] C. MUSSO, N. OUDJANE, F. L. GLAND, Improving regularised particle filters, in [7], Ch. 12.

[10] P. O. AMBLARD, J.M. BROSSIER, E. MOISAN, Phase tracking: what do we gain from optimality. Particle filtering *versus* phase-locked loops, submitted to Signal Processing, March 2001.

[11] P. TICHAVSKY, C. MURAVCHIK, A. NEHORAI, Posterior Cramér-Rao bounds for discrete time nonlinear filtering, IEEE Trans. on SP 46 (5) (1998) 1386–1396.

[12] A. BENVENISTE, M. JOINDOT, P. VANDAMME, Analyse théorique des boucles de phase numériques en présence de canaux dispersifs, Tech. rep., NT/MER/TSF/1, CNET Lannion (1980).