Particle filter applied to noisy synchronization in polynomial chaotic maps

Moussa Yahia, Pascal Acco and Malek Benslama

Abstract—Polynomial maps offer analytical properties used to obtain better performances in the scope of chaos synchronization under noisy channels. This paper presents a new method to simplify equations of the Exact Polynomial Kalman Filter (ExPKF) given in [1]. This faster algorithm is compared to other estimators showing that performances of all considered observers vanish rapidly with the channel noise making application of chaos synchronization intractable. Simulation of ExPKF shows that saturation drawn on the emitter to keep it stable impacts badly performances for low channel noise. Then we propose a particle filter that outperforms all other Kalman structured observers in the case of noisy channels.

Keywords—Chaos synchronization, Saturation, Fast ExPKF, Particle filter, Polynomial maps.

I. INTRODUCTION

CHAO TIC synchronization under noisy channel played a key role during last decade in chaotic telecommunication systems. First theoretical works on chaos synchronization neglected noise considerations [2], [3]. Then, the idea to use coupled chaotic oscillators in telecommunication is introduced [4], [5]. Additive noise in the channel destroys synchronization properties and rises the problem of noise cleaning. As performances quickly decay in presence of noise other communication schemes, as non-coherent and impulse synchronization, where considered to avoid synchronization. Kolumbán et. al give a review of communication schemes and performance limits in three papers [6], [7], [8].

The problem of synchronization takes roots in control system theory and can be seen as the state estimation of a stochastic non-linear system.

A. Kalman structured observers

Kalman filtering can be applied to synchronize systems. In the discrete time and linear case, the emitter state $x_k$ is modeled by a linear dynamical function $f(x)$, with additive dynamic noise $\eta_k \sim \mathcal{G}(0, Q)$ \(^1\) and measurement noise $v_k \sim \mathcal{G}(0, R)$:

$$
\begin{align*}
   x_{k+1} &= f(x_k) + \eta_k \\
   y_k &= h(x_k) + v_k
\end{align*}
$$

(1)

The linear measurement function $h(x)$ and measurement noise $v_k$ represent the channel model and channel noise respectively. As the dynamic noise $\eta_k$ represents real noise in the emitter, or approximate numerical noise encountered in the simulation or calculator, the initial state $x_0$ is a random variable with pdf\(^2\)

$$
p(x_0) = \mathcal{G}(x_0, \sigma_x^2)\).

The Kalman filter constructs an estimated state $\hat{x}_k$ knowing the measurements $y_k$ at the receiver. This receiver is optimal in regard of the mean square error criteria (MSE)

$$
MSE = \frac{1}{N} \sum_{k=0}^{N} (x_k - \hat{x}_k)^2
$$

(2)

The structure of the Kalman observer needs to estimate mean, variance and covariance of stochastic variables:

$$
\begin{align*}
\hat{x}_{k+1} &= \hat{x}_{k+1/k} + K_{k+1} (y_{k+1} - \hat{y}_{k+1/k}) \\
P_{k+1} &= P_{k+1/k} - K_{k+1} P_{k+1/k} y_{k+1/k} \\
K_{k+1} &= \frac{P_{k+1/k} y_{k+1/k}}{y_{k+1/k} P_{k+1/k} y_{k+1/k}}
\end{align*}
$$

(3)

For linear functions $f$ and $h$, expressions 3 are analytical and easy to compute, whereas many techniques where developed to deal with non-linear functions $f$ and $h$.

The Extended Kalman filter (EKF) uses successive linear approximations of the mean and variance statistics to build its state estimates [9][10]. Strongly non-linear systems, like chaotic dynamics, approximations generate large estimation errors which limit the use of EKF to weakly nonlinear systems.

The Unscented Kalman Filter (UKF) was introduced [11], [12] to solve this problem. The Gaussian random state is represented using a minimal number of samples, called sigma points. At each iteration, the UKF propagate the sigma points through the true nonlinear function and computes the posterior mean and covariance approximation. The approximation of higher order moments is enabled by changing the sigma points weights. This solution is advantageous to cope with strongly nonlinear system and to avoid computation of function derivative as for the EKF.

Norgaard et al. [13] use polynomial interpolation of the dynamical function $f$ and exploit Stirling’s formula to obtain the mean and covariance of the state distribution. This nonlinear transformation is then exploited in a recursive manner owing to a Kalman Filter structure.

More recently, Luca et al [1] continue this work and propose a closed-form state estimator named Exact Polynomial Kalman Filter(ExPKF) for polynomial nonlinear system with $f$ and $h$ polynomial of respective order $N$ and $M$.

ExPKF can then synchronize to a chaotic polynomial map with optimal performances. Still is the problem of emitter stability when noise is added to the chaotic recurrence.

\(^1\)probability density function

\(^2\)polynomial chaotic maps
B. Emitter stability

For secure and/or wide-band communication purpose, the emitter iterative map \( T: x_{k+1} = f(x_k) \) is placed in an attractive basin of a chaotic strange attractor. We will focus on chaotic system whose attractive basin is a simple connex set \( U = [a, b] \). Then chaotic properties guarantee that the map is onto, i.e. \( f(U) \subset U \), and there is no divergent trajectories.

Stability problems arise when the dynamic noise \( \eta_k \) is added to \( f(x_k) \). The noise can then push the stable state \( x_k \) out the attractive basin and start to follow a divergent trajectory.

An easy solution is to saturate to the set \( U \) each iteration of the map. In this case the dynamic noise \( \eta_k \) in 1 is replaced with:

\[
\eta_k^* = \begin{cases} 
\eta_k & \text{when } \eta_k \in [a - f(x_k), b - f(x_k)] \\
0 & \text{when } \eta_k < a - f(x_k) \\
b & \text{when } \eta_k > b - f(x_k)
\end{cases}
\] (4)

Whatever the noise \( \eta_k \) is, the system still remains in the attractive basin \( U \). A counterpart of this solution is that the dynamic noise \( \eta_k^* \) pdf is no more Gaussian and no more uncorrelated to the state \( x_k \).

C. Contents

In section II we give a simpler expression of mean, variance and covariance of the ExPKF that leads to a faster algorithm than [1]. Section III compares synchronization performances and shows the impact of saturation effects. To take into account emitter saturations a particle filter is introduced in the last section.

II. FASTER EXPKF ALGORITHM

In the paper [1], the authors use full Taylor series expansion to obtain exact expression of mean, variance and covariance of any random variable distribution. In this section we will compute directly the necessary statistics values of the transformed random variable through a polynomial function. Using our simpler expressions we obtain an ExPKF faster to compute.

A. Statistics through polynomial transformation

Consider a single dimensional polynomial form:

\[ y = g(x) = \sum_{n=0}^{N} a_n x^n \] (5)

In order to find the statistical values of a transformed random variable \( y \) we first focus on terms \( z = x^n \), the initial independent distribution \( x \) can written by the following form

\[ x = \bar{x} + \delta x \] (6)

where \( \delta x \) is the zero mean random variable extracted from the initial random variable \( x \) of mean \( \bar{x} \). By mean of Pascal’s triangle the terms \( x^n \) can then be expanded to:

\[ x^n = \sum_{i=0}^{n} C_n^i \bar{x}^{n-i} \delta x^i \] (7)

where \( C_n^i = \frac{n!}{i!(n-i)!} \), is the combinatorial function, consequently the expression 5 becomes:

\[ y = \sum_{n=0}^{N} a_n \sum_{i=0}^{n} C_n^i \bar{x}^{n-i} \delta x^i \] (8)

The mean \( \bar{y} = E[y] = E[g(x)] \) can be expressed as

\[ \bar{y} = \sum_{n=0}^{N} a_n \sum_{i=0}^{n} C_n^i \bar{x}^{n-i} m_i \] (9)

where \( m_i \) denote the ith-order moment of the random variable \( \delta x \). In order to facilitate the computation this last Equation (9) is written in a more compact form as

\[ \bar{y} = t^T a_{0:N} C_{0:N}^T m_{0:N} \] (10)

where \( a_{0:N} \) is the polynomial coefficient vector \( [a_0, ..., a_N]^T \), \( m_{x:j} = [m_1, ..., m_j]^T \) and \( C_{i:j}^x \) denoting a lower triangular matrix computed such as

\[ C_{i:j}^x = \begin{bmatrix}
C_0^0 & 0 & 0 & \cdots & 0 \\
C_1^0 & C_1^1 & 0 & \cdots & 0 \\
C_2^0 & C_2^1 & C_2^2 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_j^0 & C_j^1 & C_j^2 & \cdots & C_j^j
\end{bmatrix} \] (11)

Variance \( \sigma_y^2 = E[y^2] - \bar{y}^2 \) is expressed from expectation of polynomial \( y^2 \) which coefficient vector noted \( e_{0:2N} \) is obtained by the convolution product of the vector \( \bar{a}_{0:N} \) with itself:

\[ e_{0:2N} = \bar{a}_{0:N} * \bar{a}_{0:N} \] (12)

then the term \( E[y^2] \) is the same calculus than (9) applied to the polynomial vector (12)

\[ E[y^2] = \sum_{n=0}^{2N} e_n \sum_{i=0}^{n} C_n^i \bar{x}^{n-i} m_i \] (13)

finally we get the variance \( \sigma_y^2 \) in the compact matrix form

\[ \sigma_y^2 = t^T 0_{0:2N} C_{0:2N}^x m_{0:2N} - \bar{y}^2 \] (14)

Covariance \( P_{xy} \) between the two variables \( x \) and \( y \) is obtained from the polynomial \( xy \) whose coefficient are \( d_{0:N+1}^T \):

\[ P_{xy} = E[(x - \bar{x})(y - \bar{y})]E[xy] - \bar{x}\bar{y} = d_{0:N+1}^T C_{0:N+1}^x m_{0:N+1} - \bar{x}\bar{y} \] (15)

In the case where the estimates are supposed Gaussian; \( m_0 = 1, m_1 = 0 \) and for any \( k > 1 \),

\[ m_k = \begin{cases} 
(k - 1)\sigma_x^2 m_{k-2} & \text{if } k \text{ is even} \\
0 & \text{otherwise}
\end{cases} \] (16)

where \( \sigma_x^2 \) is the variance of the priori stat \( x \), then the moments vector is given by:

\[ m_{0:N}^y = [m_0, m_1, m_2, m_3, m_4, ...] = [1, 0, \sigma_x^2, 0, 3\sigma_x^4, ...]^T \]
B. Algorithm complexity

It is necessary to know in general, how much computation time is involved in implementing the algorithm, in other words to make a complexity calculation before passing to the execution on the CPU. The purpose of our redraw of the expressions is to minimize the number of operations to generate the required statistical values of the transformed random variable via the polynomial function. In the latter work (see [1]) and for an N order polynomial map, the matrix multiplication evidently requires N² multiplications, plus a smaller number of operations to compute the variance $\sigma^2$. So this ExPKF Algorithm appears to be an $O(N^3)$ process. In our development it is easy to observe through the expressions (10), (14) and (15) that the ExPKF can, in fact, be computed in $O(N^2)$ operation with an algorithm called the ExPKF Fast Algorithm which will be mentioned below.

C. ExPKF fast algorithm

Those previous simple expressions developed in (II-A) are then used to complete the prediction step of the ExPKF. This step computes the predicted statistics used to complete the update step (3).

The mean and the covariance of the predicted state are obtained with (10) and (14):

$$\tilde{x}_{k+1/k} = E[f(x_k)] + \eta_k = \alpha_{0:N}^T C_{0:N}^{x/k} x_k + \eta_k$$

$$P_{k+1/k} = E[(x_{k+1/k} - \tilde{x}_{k+1/k})^2] = \alpha_{0:2N}^T C_{0:2N}^{x,x} m_{0:2N} + \eta_k$$

The predicted observation, the transition/innovation covariances and the variance are obtained with (10), (15) and (14):

$$\tilde{y}_{k+1/k} = E[h(x_{k+1/k})] = b_{0:M}^T C_{0:M}^{x/k} x_{k+1/k}$$

$$P_{\tilde{y}_{k+1/k}} = E[(\tilde{y}_{k+1/k} - \tilde{y}_{k+1/k})^2] = \alpha_{0:2M}^T C_{0:2M}^{x,x} m_{0:2M} + \eta_k$$

$$P_{\tilde{y}_{k+1/k}} = \tau_{0:2M}^T C_{0:2M}^{x,y} m_{0:2M}$$

III. SATURATION IMPACT ON SYNCHRONIZATION PERFORMANCE

Luca et al. compare synchronization performances of the EKF, UKF and the Scaled Uncented Kalman Filter (ScUKF) to the ExPKF. Figure 7 in [1] gives the normalized mean square error $MSE/R$ obtained with those filters for a fourth order Chebyshev polynomial $f(x) = T_4(x) = 8x^4 - 8x^2 + 1$. We note that for noise variance $R > 10^{-2}$ all filters performances tend to a limit $MSE/R \approx 0.94$. That means we can expect synchronization only for very clean channels. Luca’s simulations are done with a dynamic noise $Q = \frac{a}{R^2}$ which do not help to distinguish whether the dynamic noise or measurement noise limits the $MSE/R$ performances to 0.94.

Moreover, Section I-B stands that dynamic noise $\eta_k$ generates saturation at the emitter. To seek the real impact of saturations, Fig. 2 shows performances with $Q$ and $R$ varying independently using our fast ExPKF algorithm.

Saturations occur more and more frequently as noise variance $Q$ increase. The hypothesis of Gaussian dynamic noise $\eta_k$ is not met and turn this noise into a non Gaussian correlated noise $\eta_n$ of equation 4. Then the ExPKF becomes under optimal. The Exact Polynomial Kalman Filter Initialized ExPKFI is then introduced as an optimal filter which can handle saturations. This filter uses an ideal second channel to transmit the information s(k) whether or not a saturation at step time k exists. This filter is given as a theoretical boundary to measure performance losses due to saturation.

The ExPKFI operates like the ExPKF when there is no saturation $s(k) = 0$. When an upper/lower saturation happens $s(k) = a$ or $b$, then the filter is re-initialized to $\tilde{x}_k = s(k)$ with estimated covariance $P_{\tilde{x}_k/y_k} = 0$ as the state is perfectly known at each saturation.

Fig. 2 leads to the following remarks:
IV. PARTICLE FILTER

Particle Filter (PF) was introduced by Gordon et al. in [14]. The same model 1 is extended to any nonlinear function \( h \) and the weights \( w \) are updated using Bayesian rules with the posterior pdf.

The main problem of this algorithm is that the weight \( \delta \) decreases and that the filter is likely to degenerate. However an indicator for the degree of degeneracy:

\[
N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}(w_i^\text{f})^2}
\]

which is the effective sample size presented in [15]. When \( N_{\text{eff}} < 0.75 \) it is necessary to resample the particle as proposed in the generic particle filter GPF (Algorithm 3 of [15]).

B. GPF applied to polynomial map

To compare with results obtained with Kalman filter, we apply the GPF to the same polynomial map with Gaussian

\[
\text{iid} \sim \text{N}(0, \sigma^2)
\]
noises $\eta_k$ and $\nu_k$. To handle saturations phenomena, the noise $\eta_k$ is replaced in the GPF filter by its Non-Gaussian uncorrelated counterpart $\eta_k^\star$. In fact this is simply done in the same way as for the emitter by computing Gaussian noise $\eta_k$ and truncate the value of $f(x_k) + \eta_k$ into the set $U$.

It appears clearly in Fig 3 that the GPF works very efficiently for high channel noise and tends to the performance of the ExPKFI.

Small measurement noise lower the value of $P(y_k/x_k = \zeta_k)$ and then decrease rapidly the particle weights. Then several resampling steps are operated which explain the unexpected bad performances of the filter for low noise channel.

V. C ONCLUSION

This paper offers two main results. First we redraw equations of the Exact Polynomial Kalman Filter. We minimize the Algorithm complexity to $O(N^2)$, however, we obtained a faster ExPKF algorithm which confirms the results of Luca et al. Then we apply particle filtering to polynomial chaotic maps to handle properly saturations that are drawn in the emitter to keep it stability. Performance in terms of mean square synchronization error are plotted and compared to previous results. The particle filter outperform all results and offer possibility to use synchronization under noisy channels for communication applications.

REFERENCES