Optimal filtering in hidden and pairwise Gaussian Markov systems

I. GORYNIN, E. AZERAF, W. SABBAGH, E. MONFRINI and W. PIECZYNSKI
SAMOVAR, Telecom SudParis, CNRS
Université Paris-Saclay
9, rue Charles Fourrier, 91000 Evry
FRANCE
ivan.gorynin@telecom-sudparis.eu   http://citi.telecom-sudparis.eu/fr/

Abstract: In a hidden Markov model (HMM), the system goes through a hidden Markovian sequence of states (**X**) and produces a sequence of emissions (**Y**). We define the hidden Gaussian Markov model (HGMM) as the HMM where the hidden process is Gaussian and is affected by a normal white noise. The Kalman filter (KF) is a fast optimal statistical estimation method for the HGMMs and is very popular among the practitioners. However, the classic HGMM formulation is too restrictive. It extends to recent pairwise Gaussian Markov model (PGMM) where we assume that the pair (**X**, **Y**) is Gaussian Markovian. Moreover, there exists a KF version for the PGMM. The PGMM is more general than HGMM and in particular, the PGMM hidden process is not necessarily Markovian. The authors share their findings on about enhancing the KF when improving HGMM to PGMM. We discover singular cases where HGMM is at least ten times less accurate than the PGMM. On average, PGMM outperforms HGMM by twenty percent.

Key-Words :— Bayesian estimation, Hidden Markov models, Pairwise Markov models, Kalman filter.

1 Introduction

Let us consider two random sequences $\mathbf{X} = (X_1, ..., X_N)$, and $\mathbf{Y} = (Y_1, ..., Y_N)$, taking their values in R^m and R^q, respectively. We consider that the pair $(\mathbf{X}, \mathbf{Y})$ is a zero-mean Gaussian Markovian process. Its distribution is characterized by $\mathbf{Z}_1 = (X_1^T, Y_1^T)^T$ and the equations (for $n = 1, \ldots, N-1$):

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} A_{n+1}(1,1) & A_{n+1}(1,2) \\ A_{n+1}(2,1) & A_{n+1}(2,2) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} B_{n+1}(1,1) & B_{n+1}(1,2) \\ B_{n+1}(2,1) & B_{n+1}(2,2) \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}, \tag{1.1}$$

where $((U_2)^T, (V_2)^T)^T$, …, $((U_N)^T, (V_N)^T)^T$ are white noise normal vectors and $A_{n+1}(1,1)$, …, $B_{n+1}(2,2)$ are matrices. If $\mathbf{Z}_n = ((X_n)^T, (Y_n)^T)^T$, $\mathbf{W_{n+1}} = ((U_{n+1})^T, (V_{n+1})^T)^T$ and $\mathbf{Z} = (\mathbf{Z}_1, ..., \mathbf{Z}_N)$, we represent (1.1) by

$$\mathbf{Z_{n+1}} = \mathbf{A_{n+1}} \mathbf{Z_n} + \mathbf{B_{n+1}} \mathbf{W_{n+1}}. \tag{1.2}$$

This model is known as the *pairwise* Gaussian Markov model (PGMM). It generalizes the classic *hidden* Gaussian Markovian model (HGMM). The HGMM equations are:

$$X_{n+1} = F_{n+1} X_n + Q_{n+1} U_{n+1}; \tag{1.3}$$
$$Y_{n+1} = H_{n+1} X_{n+1} + R_{n+1} V_{n+1}, \tag{1.4}$$

where F_{n+1}, Q_{n+1}, H_{n+1} and R_{n+1} are matrices and U_1, V_2, …, U_N, V_N are white noise normal vectors [1,2]. Indeed, HGMM (1.3)-(1.4) verifies

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} F_{n+1} & 0 \\ H_{n+1} F_{n+1} & 0 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} Q_{n+1} & 0 \\ H_{n+1} Q_{n+1} & R_{n+1} \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}, \tag{1.5}$$

which shows that a HGMM is a PGMM thanks to (1.1).

Note that for each n, the PGMM transition kernel $p(x_{n+1}, y_{n+1} | x_n, y_n)$ depends on the eight matrices $A_{n+1}(1,1)$, … , $B_{n+1}(2,2)$, while HGMM depends only on four matrices F_{n+1}, Q_{n+1}, H_{n+1} and R_{n+1}.

Let us recall that there exist HGMM and PGMM versions of the Kalman filter (KF) [3,4]. In the paper, we aim to apprehend the difference between them from the perspective of simulation experiments. That is to know if the PGMM outperforms the HGMM significantly. The scope of this paper includes only the context of scalar state and observation spaces, *i.e.* where $m = q = 1$.

The paper structure is the following. The next section is devoted to KF in the PGMM framework: we describe the algorithm and we prove the involved equations. Section 3 presents the scalar stationary reversible Gaussian processes which we study through simulations. Section 4 contains experiments and last section contains concluding remarks and perspectives.

2 Kalman filter in pairwise Gaussian Markov models

Let us consider a PGMM (1.1). The PGMM-KF runs as follows.

If we set $E\left[\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}^T\right] = \begin{bmatrix} \Gamma^x & \Gamma^{xy} \\ \Gamma^{yx} & \Gamma^y \end{bmatrix}$, we have $p(x_1 | y_1) = \mathrm{N}(M_1, \Gamma_1)$ where M_1 and Γ_1 are

$$M_1 = \Gamma^{yx}[\Gamma^x]^{-1} y_1 \text{ and } \Gamma_1 = \Gamma^y - \Gamma^{yx}[\Gamma^x]^{-1}\Gamma^{xy}. \quad (2.1)$$

Formally, we can consider that for all n $p(x_n | y_1^n) = \mathrm{N}(M_n, \Gamma_n)$ which leads us to focus on how to compute M_{n+1} and Γ_{n+1} from y_{n+1}, M_n and Γ_n:

(i) compute the mean and the variance of $p(x_{n+1}, y_{n+1} | \mathbf{y}_1^n)$ by

$$\begin{bmatrix} M_{n+1}^1 \\ M_{n+1}^2 \end{bmatrix} = \mathbf{A}_{n+1}\begin{bmatrix} M_n \\ y_n \end{bmatrix},$$
$$\begin{pmatrix} a_{n+1} & \beta_{n+1} \\ \gamma_{n+1} & \delta_{n+1} \end{pmatrix} = \mathbf{B}_{n+1}[\mathbf{B}_{n+1}]^T + \mathbf{A}_{n+1}\begin{bmatrix} \Gamma_n & 0 \\ 0 & 0 \end{bmatrix}\mathbf{A}_n^T; \quad (2.2)$$

(ii) compute M_{n+1} and Γ_{n+1} by

$$M_{n+1} = M_{n+1}^1 + \beta_{n+1}\delta_{n+1}^{-1}(y_{n+1} - M_{n+1}^2), \quad (2.3)$$
$$\Gamma_{n+1} = \alpha_{n+1} - \beta_{n+1}\delta_{n+1}^{-1}\gamma_{n+1}.$$

Proof We use the law of total expectation $E[U] = E[E[U|V]]$ with $U = \begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix}$, $V = X_n$ and $E[.] = E[. | \mathbf{y}_1^n]$: $E\left[\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} \middle| \mathbf{y}_1^n\right] = E\left[\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} \middle| X_n, \mathbf{y}_1^n\right]$ $= E\left[\mathbf{A}_{n+1}\begin{bmatrix} X_n \\ y_n \end{bmatrix} \middle| \mathbf{y}_1^n\right] = \mathbf{A}_{n+1}\begin{bmatrix} M_n \\ y_n \end{bmatrix}$.

Similarly, $E[\mathbf{Z}_{n+1}\mathbf{Z}_{n+1}^T | \mathbf{y}_1^n] = E[E[\mathbf{Z}_{n+1}\mathbf{Z}_{n+1}^T | X_n, \mathbf{y}_1^n]] =$ $\mathbf{B}_{n+1}\mathbf{B}_{n+1}^T + E\left[\mathbf{A}_{n+1}\begin{bmatrix} X_n \\ y_n \end{bmatrix}\begin{bmatrix} X_n^T & y_n^T \end{bmatrix}\mathbf{A}_n^T \middle| \mathbf{y}_1^n\right] =$ $\mathbf{B}_{n+1}\mathbf{B}_{n+1}^T + A_{n+1}\begin{bmatrix} \Gamma_n + M_n M_n^T & M_n y_n^T \\ y_n M_n^T & y_n y_n^T \end{bmatrix}\mathbf{A}_n^T$,

so $\begin{bmatrix} a_{n+1} & \beta_{n+1} \\ \gamma_{n+1} & \delta_{n+1} \end{bmatrix} = E[\mathbf{Z}_{n+1}\mathbf{Z}_{n+1}^T | \mathbf{y}_1^n] - \mathbf{A}_{n+1}\begin{bmatrix} M_n M_n^T & M_n y_n^T \\ y_n M_n^T & y_n y_n^T \end{bmatrix}\mathbf{A}_n^T =$ $= \mathbf{B}_{n+1}\mathbf{B}_{n+1}^T + \mathbf{A}_{n+1}\begin{bmatrix} \Gamma_n & 0 \\ 0 & 0 \end{bmatrix}\mathbf{A}_n^T$. Then we use the formula for multivariate normal distributions to compute the conditional probability density function $p(x_{n+1} | \mathbf{y}_1^{n+1}) = p(x_{n+1} | y_{n+1}, \mathbf{y}_1^n)$.

□

3 Gaussian stationary reversible Markov models

A stationary PGMM (1.1) is such that $p(x_n, y_n, x_{n+1}, y_{n+1})$ does not depend on n: for each $n = 1, \ldots, N-1$,

$$p(x_n, y_n, x_{n+1}, y_{n+1}) = p(x_1, y_1, x_2, y_2). \quad (3.1)$$

For $\mathbf{Z}_n = ((X_n)^T, (Y_n)^T)^T$, the distribution of a stationary PGMM derives from the variance-covariance matrix

$$\begin{bmatrix} E[\mathbf{Z}_1\mathbf{Z}_1^T] & E[\mathbf{Z}_1\mathbf{Z}_2^T] \\ E[\mathbf{Z}_2\mathbf{Z}_1^T] & E[\mathbf{Z}_2\mathbf{Z}_2^T] \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma}_{Z_1} & \mathbf{\Gamma}_{Z_1Z_2} \\ \mathbf{\Gamma}_{Z_2Z_1} & \mathbf{\Gamma}_{Z_2} \end{bmatrix}, \quad (3.2)$$

with $\mathbf{\Gamma}_{Z_1} = \mathbf{\Gamma}_{Z_2}$.

We say that a stationary PGMM is reversible if $p(x_2, y_2 | x_1, y_1) = p(x_1, y_1 | x_2, y_2)$ holds.

4 Experiments

Let us now consider scalar stationary reversible PGMMs. Since we study the correlations-induced effect, we restrict our study to the case where the variances of X_1, X_2, Y_1, Y_2 are all unitary. So the covariance matrix of (X_1, X_2, Y_1, Y_2) is:

$$\begin{bmatrix} \mathbf{\Gamma}_{Z_1} & \mathbf{\Gamma}_{Z_1Z_2} \\ \mathbf{\Gamma}_{Z_2Z_1} & \mathbf{\Gamma}_{Z_2} \end{bmatrix} = \begin{bmatrix} 1 & b & a & d \\ b & 1 & d & c \\ a & d & 1 & b \\ d & c & b & 1 \end{bmatrix}. \quad (4.1)$$

Thus, we define a stationary reversible PGMM with four correlations a, b, c and d. In fact, we can establish that the HGMM derived from such a PGMM is constrained to $d = ab$ and $c = ab^2$. Thus, for given a and b, the difference between a PGMM and the corresponding HGMM increases when the difference between d and ab or c and ab^2 increases.

A PGMM is a HGMM if the conditions below hold:

(H1) X_2 and Y_1 are independent given X_1;

(H2) Y_1 and Y_2 are independent given (X_1, X_2).

In order to study the influence of each of these limiting conditions, we also consider two intermediary models: PGMM with "independent noise" (PGMM-IN), in which (H1) holds, but not (H2), and HGMM with "correlated noise" (HGMM-CN), in which (H2) holds but not (H1). In the sequel we refer the genuine PGMM as PGMM "with correlated noise" (PGMM-CN), and the classic HGMM as HGMM-IN.

We supply the graphical representations of these sub-models in Fig. 1-4. Note that if $(\mathbf{X},\mathbf{Y})$ is stationary reversible, (H1) holds if and only if $\mathbf{X}$ is Markovian [9].

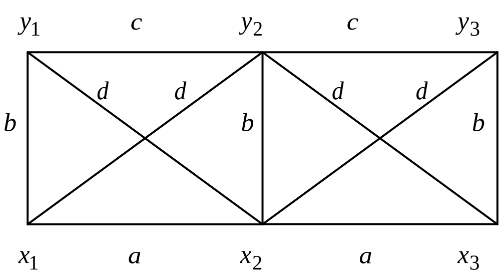

Fig. 1. The dependence graph of the PGMM-CN: free *(a,b,c,d)*.

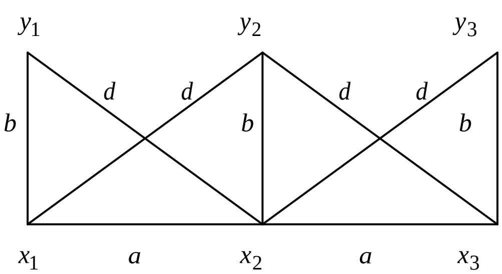

Fig. 2. The dependence graph of the PGMM-IN : free *(a,b,d)*; $c=\frac{b(d-ab)+d(b-ad)}{1-a^2}$.

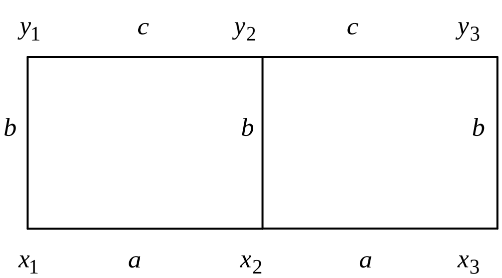

Fig. 3. The dependence graph of the HGMM-CN: free *(a,b,c)*; *d* constraint to *d=ab*.

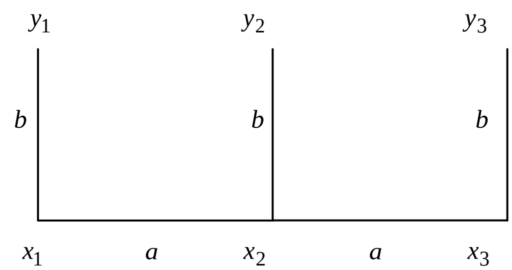

Fig. 4. The dependence graph of the HGMM-IN: free *(a,b)*; *c* and d are constraint to $c=ab^2$, $d=ab$.

In the case where PGMM is stationary reversible, $\mathbf{A_{n+1}}$ and $\mathbf{B_{n+1}}$ in (1.2) do not depend on n and verify

$$A=\begin{bmatrix} a & d \\ d & c \end{bmatrix}\begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix}^{-1} \tag{4.2}$$

and

$$BB^T=\begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix}-\begin{bmatrix} a & d \\ d & c \end{bmatrix}\begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix}^{-1}\begin{bmatrix} a & d \\ d & c \end{bmatrix}. \tag{4.3}$$

We study how the quality of the estimate of the state vector worsens due to the miss-modeling. The equations to transform PGMM into PGMM-IN, HGMM-CN and HGMM-IN (see Remark 1 for $r_2(\theta)$) are:

$$r_2(\theta)=(a,b,\frac{b(d-ab)+d(b-ad)}{1-a^2},d); \tag{4.4}$$

$$r_3(\theta)=(a,b,c,ad); \tag{4.5}$$

$$r_4(\theta)=(a,b,ab^2,ad), \tag{4.6}$$

where $\theta=(a,b,c,d)$. To cast a PGMM-CN in PGMM-IN, we have to modify the value of c to come up with the expected conditional independence of Y_n and Y_{n+1} with regard to (X_n,X_{n+1}). The overall vector is multivariate normal, therefore the covariance matrix of distribution of (Y_n,Y_{n+1}) conditional on (X_n,X_{n+1}) is *cf.* (4.1):

$$\begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}-\frac{1}{1-a^2}\begin{bmatrix} b & d \\ d & b \end{bmatrix}\begin{bmatrix} 1 & -a \\ -a & 1 \end{bmatrix}\begin{bmatrix} b & d \\ d & b \end{bmatrix}.$$

Therefore, the conditional covariance of Y_n and Y_{n+1} is zero if and only if $c=\frac{b(d-ab)+d(b-ad)}{1-a^2}$.

We also have:

$$r_1(\theta) = \theta \tag{4.7}$$

because the PGMM-CN is the most inclusive.

Let us now denote by $\varepsilon_k(\theta)$ the asymptotic error rate of the Kalman filter which uses the proposal model $r_k(\theta)$ to estimate the state vector which arises from θ. In other words, we define

$$\varepsilon_k(\theta) = \lim_n E[(\mathbf{L}_n(r_k(\theta))\mathbf{Y}_1^n - X_n)^2 | \theta], \tag{4.8}$$

where $\mathbf{L}_n(r_k(\theta))$ stands for the KF response at the step n with the proposal $r_k(\theta)$. We involve the conditional expectation in a sense that the distribution of $(X_n, \mathbf{Y}_1^n)$ depends on θ (the base model parameter). Let us observe that for each θ, we have $r_1(\theta) = \theta$ therefore, for each k, $\varepsilon_k(\theta) \geq \varepsilon_1(\theta)$. We can expand and simplify the expression of $\varepsilon_k(\theta)$:

$$\varepsilon_k(\theta) =$$
$$\lim_n [\mathbf{L}_n(r_k(\theta))\mathbf{\Gamma}_{Y_1^n}(\theta)\mathbf{L}_n^T(r_k(\theta)) - 2\mathbf{L}_n(r_k(\theta))\mathbf{\Sigma}_{Y_1^n, X_n}(\theta) + 1],$$

where $\mathbf{\Gamma}_{Y_1^n}(\theta) = E[\mathbf{Y}_1^n(\mathbf{Y}_1^n)^T]$ is the variance matrix of the vector $\mathbf{Y}_1^n$ and $\mathbf{\Sigma}_{Y_1^n, X_n}(\theta) = E[\mathbf{Y}_1^n X_n]$ is the covariance between $\mathbf{Y}_1^n$ and X_n. Both are available analytically. For our computations, we suppose that we reach the limit at $n = 100$.

We go further and we define for each *k, k'* the cross-model error rate:

$$\xi_{k,k'}(\theta) = \varepsilon_{k'}(r_k(\theta)). \tag{4.9}$$

We present in Table 1 the cross-model error rates for $\theta = (a,b,c,d) = (0.325, 0.250, 0.925, 0.025)$. This parameter, found empirically by looping over all possible parameters, gives a maximal error gap $\varepsilon_4(\theta) - \varepsilon_1(\theta)$. The related parameters $r_2(\theta)$, $r_3(\theta)$ and $r_4(\theta)$ are given in Table 2.

We conclude from Table 1 that there exist situations in which using the classic HGMM-IN instead of the PGMM-CN (with the same a and b in both models) significantly worsens the KF efficiency. Besides, we observe that HGMM-CN was not as accurate as PGMM-CN. Therefore, we conclude generally that the effectiveness of the PGMM-CN is not only due to relaxing condition (H2) but to relaxing both (H1) and (H2). Indeed, we note that the KF is particularly sensitive to the value of d in this case study.

	Base model parameters $r_k(\theta)$			
Proposals	HGMM -IN	HGMM -CN	PGMM -IN	PGMM -CN
HGMM-IN	0,931	0,984	0,939	0,998
HGMM-CN	0,931	0,721	0,940	0,489
PGMM-IN	0,931	0,984	0,937	0,958
PGMM-CN	0,931	0,721	0,937	0,020

Table 1. The cross-model error rates $\xi_{k,k'}(\theta)$ for θ which maximizes the error gap $\varepsilon_4(\theta) - \varepsilon_1(\theta)$. Each model name relates to a projection indexed by *k* or *k'*.

	Parameters			
Projections	a	b	c	d
HGMM-IN	0,325	0,250	0,020	0,081
HGMM-CN	0,325	0,250	0,925	0,081
PGMM-IN	0,325	0,250	-0,009	0,025
PGMM-CN	0,325	0,250	0,925	0,025

Table 2. The parameter θ which maximizes the error gap $\varepsilon_4(\theta) - \varepsilon_1(\theta)$, and its projections to the PGMM–IN, HGMM–CN and HGMM–IN parameter spaces. Each model name relates to a projection.

To make the text easier to read, we drop the suffixes of PGMM-CN and HGMM-IN for the remainder of the paper. That is, for the sake of consistency, PGMM stands for PGMM-CN and HGMM stands for HGMM-IN.

We now evaluate the average gain in performance resulting from using PGMM instead of HGMM *on average*. Our methodology is to sample random matrices (4.1) from the related uniform distribution [5], and we note the log-gap $\frac{\varepsilon_k(\theta)}{\varepsilon_1(\theta)}$ for each of them. In order to imitate the real-world data, we discard any θ for which $a < 0.5$, $b < d$ or $b < c$. We report in Fig. 5 the empirical cumulative density function (cdf) that we obtain in our experiment. We also report in Table 3 the average cross-model error rates.

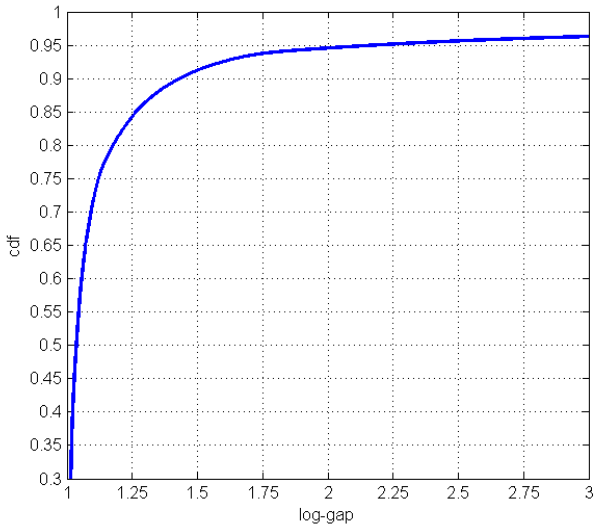

Fig. 5. The cdf of the log-gaps between PGMM and HGMM.

	Base model parameters $r_k(\theta)$			
Proposals	HGMM -IN	HGMM -CN	PGMM -IN	PGMM -CN
HGMM-IN	0,489	0,477	0,486	0,475
HGMM-CN	0,489	0,466	0,491	0,458
PGMM-IN	0,489	0,477	0,482	0,468
PGMM-CN	0,489	0,466	0,482	0,444

Table 3. Average cross-model error rates $E[\xi_{k,k'}(\theta)]$. Each model name relates to a projection indexed by *k* or *k'*.

Here are the conclusions that we have done:

(i) the average log-gap is 1.216, what enables us to state that the PGMM should improve the HGMM by 22% on average;
(ii) according to Fig. 5, the PGMM has a 15% chance to outperform HGMM by 25%, 10% chance to outperform by 50% and 5% to outperform by 100%;
(iii) in our simulations, the maximum values of c and d are inferior to a. That allows obtaining relevant parameters for describing the real-world systems. Without these limits, PGMM outperforms HGMM by 39% on average;
(iv) the study-case from Table 1 is the most disadvantageous for HGMM; even if parameters may not refer to real word, it shows that the effectiveness of the PGMM over HGMM may have no limit.

5 Conclusion

We show from the perspective of simulation experiments that the PGMM should improve the HGMM by 20% on average. Our future works will concern with the triplet Gaussian Markov models [6] and PGMM/TGMM parameter learning [7-9].

References:

[1] O. Cappé, E. Moulines, and T. Ryden, *Inference in hidden Markov models*, Springer, 2005.

[2] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, Wiley, 2008.

[3] V. Nemesin and S. Derrode, Robust blind pairwise Kalman algorithms using QR decompositions, *IEEE Trans. on Signal Processing*, Vol. 61, No. 1, 2013, pp. 5-9.

[4] V. Nemesin and S. Derrode, Robust partial learning in linear Gaussian systems, *IEEE Trans. on Automatic Control*, Vol. 60, No. 9, 2015, pp. 2518-2523.

[5] A. Genz, Methods for generating random orthogonal matrices. *Monte Carlo and Quasi-Monte Carlo Methods.* 1998, pp. 199—213.

[6] W. Pieczynski and F. Desbouvries, On triplet Markov chains, *International Symposium on Applied Stochastic Models and Data Analysis*, Brest, France, May 2005.

[7] M. V. Kulikova, Gradient-based Parameter Estimation in Pairwise Linear Gaussian System, IEEE Transactions on Automatic Control, accepted, 2016.

[8] R. S. Liptser and A. N. Shiryaev, Statistics of Random Processes: I. General theory, Springer, 2001.

[9] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski, Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise, Signal Processing, Vol. 91, No. 2, 2011, pp. 163-175.