CODE-AIDED TIMING SYNCHRONIZATION FOR MULTI-*H* CPM AT LOW SIGNAL-TO-NOISE RATIO

Xie Shunqin, Zhou Ke, Chen Dahai, Li Xianglu
Institute of Electronic Engineering, China Academy of Engineering Physics
Mianyang, Sichuan, China 621999
suntrain08@gmail.com

ABSTRACT

In order to solve the problem of timing synchronization at low signal-to-noise ratio(SNR) for Multi-*h* CPM, a code-aided early-late loop(ELL) algorithm is proposed. The algorithm is based on the iterative detection of serially concatenated Multi-*h* CPM with convolutional codes. The ELL timing estimator based on sequence detection is extended to the maximum-logarithmic maximum a posteriori (max-log MAP) detection. By using the information updated by iterative detection, the timing accuracy of multi-*h* CPM can be improved at low SNR. The simulation results show that, even when the bit signal-to-noise ratio (Eb/No) is as low as 3dB~5dB, the estimating variance of the proposed synchronization can be close to the Cramer Rao bound(MCRB) of ARTM CPM. After this timing synchronizing, the detection performance of the 10th iteration is only 0.03dB loss compared with the performance with ideal synchronization.
Keywords: Multi-*h* CPM; Timing; Synchronization; Code-aided; Early-late loop

INTRODUCTION

As one of waveforms in the IRIG106 standard[1], Multi-*h* CPM signals have a constant envelope and take into account both the spectrum efficiency and power efficiency. However, the problem of their timing synchronization is especially serious as a quaternary, partial response CPM signals[2]. Even at the region of medium and high SNR where the uncoded system is working, the traditional non-data-aided(NDA)[3] timing or decision-directly(DD)[4] timing technologies are difficult to work. Although the NDA timing method based on the average filter has a wide acquisition range of timing error, its accuracy is poor. On the other hand, the DD timing method has good performance, but it has false-lock problem. Although some solutions for this issue were proposed in some literature[5], it could only restart capture when the false-lock occurred, which would delay the capture time.

We adopted an timing scheme based on early-late loop (ELL) and sequence detection for the uncoded Multi-*h* CPM system in[6]. The timing synchronization algorithm could effectively solve the false-lock problem of timing synchronization for multi-*h* CPM signal. At the same time, it had a high timing synchronization accuracy. After simplifying with the PAM decomposition or other efficient low-complexity techniques[7], the practicability of the ELL timing was improved.

Due to the characteristics of convolutional code, multi-*h* CPM can be used as an inner code combined with a convolutional code in the serially concatenated coded-modulation system with interleaving and iterative decoding[8]. However, in the SNR region of the coded system operating, the timing synchronization problem is further highlighted. Even the ELL timing based on sequence detection can not achieve the perfect timing synchronization when the Eb/No at the region of 3dB-5dB.

In this paper, based on the iterative detection of serially concatenated Multi-*h* CPM, the ELL timing estimation in sequence detection is extended to the Max-log-MAP algorithm, which forms the code-aided ELL(CA-ELL) timing synchronization. Simulation results will be compared with another code-aided timing synchronization based on expectation maximum(EM) algorithm[9].

ITERATIVE DETECTION OF SERIALLY CONCATENATED MULTI-*H* CPM

The transmission and iterative reception of serially concatenated Multi-*h* CPM with a convolutional code is shown in Figure 1.

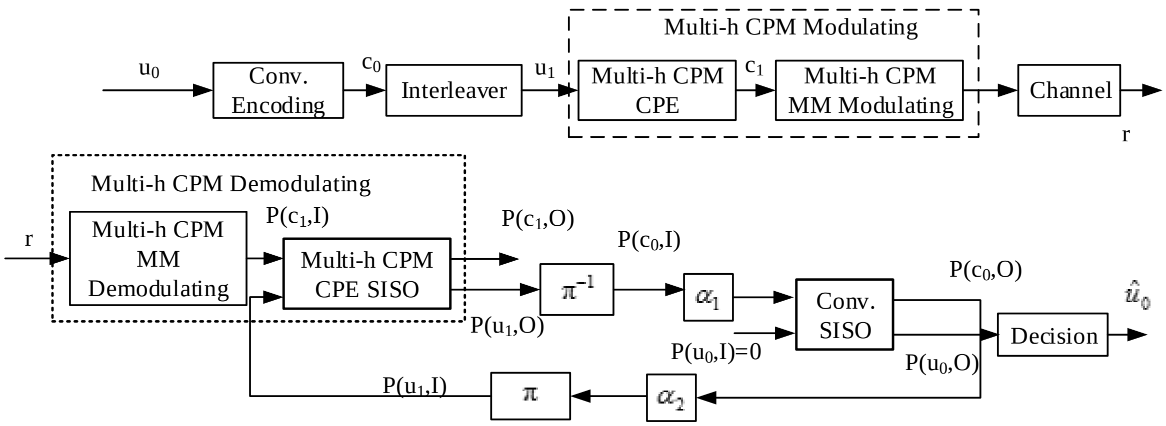

Figure 1The transmission and iterative reception of serially concatenated Multi-***h* CPM with a convolutional code.**

In the transmitter, the random interleaver is placed after convolutional encoding and symbol mapping, and interleaved symbols are modulated as Multi-*h* CPM signal and transmitted to the channel. Additive Gauss white noise channel is adopted in this paper. In the receiver, the soft-input and soft-output (SISO) module of multi-*h* CPM uses the max-log MAP algorithm. The extrinsic information from output of the SISO module is de-interleaved (π^{-1}) and weighted by a coefficient, then sent to the SISO of convolutional code as its prior information. The output from the SISO of convolutional code decoding is returned to the SISO's input of Multi-*h* CPM after weighting and interleaving (π), which is used as priori information of multi-*h* CPM to complete the iterative detection.

The max-log MAP algorithm of Multi-*h* CPM is based on the BCJR algorithm, which mainly includes three parts: forward recursion, backward recursion and posteriori probability calculation. The forward recursive ($\alpha_k(\sigma)$) and backward recursive ($\beta_{k-1}(\sigma)$) of state σ at the *k*th symbol interval are:

$$\alpha_k(\sigma) = \max_{e:\sigma_k^E(e)=\sigma}\left(\alpha_{k-1}\left(\sigma_k^S(e)\right) + \lambda_k\left(u(e);I\right) + \lambda_k\left(c(e);I\right)\right)$$
$$\beta_{k-1}(\sigma) = \max_{e:\sigma_k^S(e)=s}\left(\beta_k\left(\sigma_k^E(e)\right) + \lambda_k\left(u(e);I\right) + \lambda_k\left(c(e);I\right)\right) \quad (1)$$

Where $\sigma_k^S(e)$ and $\sigma_k^E(e)$ indicate the start state and termination state respectively of the path '*e*' at the *k*th symbol interval. $\lambda_k\left(u(e);I\right)$ represents a priori information input of the information symbol *u*(*e*) corresponding to the path '*e*' at the *k*th symol interval, which is from the convolutional decoder; $\lambda_k\left(c(e);I\right)$ is the input of a priori information of the codeword *c* (*e*) corresponding to the path '*e*', which is from the output of matched filters.

The max-log MAP algorithm can be realized as fixed-window or sliding-window. In order to adapt to the timing synchronizing process, the sliding-window method is used in this paper.

TIMING ERROR DETECTION USED ELL BASED ON SEQUENCE DETECTION FOR MULTI-*H* CPM

In the maximum likelihood sequence detection(MLSD)of Multi-*h* CPM, the maximum accelerated metric of all surviving paths on every state σ is assumed as λ_{surv}^{ML}, which can be expressed as:

$$\lambda_{surv}^{ML} = \max_{\sigma}\left\{\lambda_\sigma\right\} \quad (2)$$

The relationship between λ_{surv}^{ML} and normalized timing offset (τ) is shown in Figure 2, and λ_{surv}^{ML} is shortened to λ_{ML}. It can be seen that $\lambda_{ML}(\tau)$ has the largest value at $\tau = 0$ and is symmetrical about $\tau = 0$. While the $\lambda_{ML}(\tau)$ is shifted to the left and right sides with 0.5δT (where the *T* is the symbol interval), the maximum likelihood (ML) metrics of the early branch and late branch can be obtained respectively:

$$\lambda_{lt}(\tau) = \lambda_{ML}(\tau - 0.5\delta T)$$
$$\lambda_{el}(\tau) = \lambda_{ML}(\tau + 0.5\delta T) \quad (3)$$

$\lambda_{ML}(\tau)$, $\lambda_{lt}(\tau)$ and $\lambda_{el}(\tau)$ are shown in figure 2. From the Figure 2, $\lambda_{\mathrm{el}}(\tau)$ and $\lambda_{\mathrm{lt}}(\tau)$ have the following relationship:

$$\begin{cases} \lambda_{\mathrm{el}}(\tau) > \lambda_{\mathrm{lt}}(\tau), \tau < 0 \\ \lambda_{\mathrm{el}}(\tau) = \lambda_{\mathrm{lt}}(\tau), \tau = 0 \\ \lambda_{\mathrm{el}}(\tau) < \lambda_{\mathrm{lt}}(\tau), \tau > 0 \end{cases} \quad (4)$$

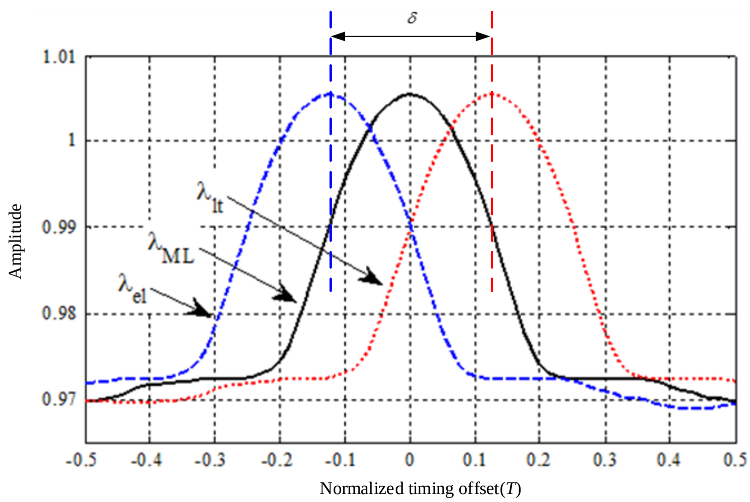

Figure 2 Metrics vs normalized timing offset

Therefore, the *i*th estimation of timing error can be constructed as the difference between the early branch and late branch metrics:

$$e(i) = \lambda_{lt}^{i} - \lambda_{el}^{i} \tag{5}$$

Where $e(i)$ is the timing error estimation of the received signal in the interval of $(iD+1)\ T\sim(iD+D)\ T$ (D is block length of the accumulated metrics), and can be expressed as:

$$\begin{aligned} \lambda_{el}^{i} &= \max_{\sigma}\left\{\left(\lambda_{\sigma}^{el}(n)\Big|_{n=iD+1}^{n=iD+D}\right)\Big/D\right\} \\ \lambda_{lt}^{i} &= \max_{\sigma}\left\{\left(\lambda_{\sigma}^{lt}(n)\Big|_{n=iD+1}^{n=iD+D}\right)\Big/D\right\} \end{aligned} \tag{6}$$

Where, $\lambda_{\sigma}(n)\Big|_{n=iD+1}^{n=iD+D}$ represents the accumulated metric corresponding to the surviving path of state σ in time interval of $\left[(iD+1)T,(iD+D)T\right]$.The sampling offset δ of the early and late branch will determine the capture range and accuracy of the timing error estimation. The S-curves of the timing error estimator with different δ are shown in Figure 3.

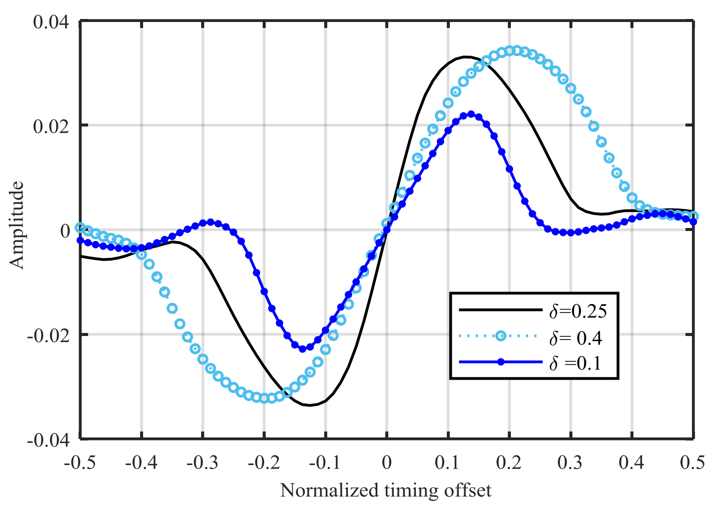

Figure 3 S-Curves of timing error estimator based on early-late loop

The timing error detector(TED) according to this principle is shown in Figure 4. The sampling offsets of the 'late' branch and 'early' branch are $0.5\delta T$ and $-0.5\delta T$ respectively compared with the demodulation branch whose sampling offset is 0. The accumulated metrics are obtained by 'adding-comparing-selecting'(ACS) of Viterbi algorithm. The metrics of the whole states outputs every D symbol, and then the cumulative values of all metrics are initialized. Then the maximum metric of the early branch λ_{el}^{i} and the maximum metric of late branch λ_{lt}^{i} are selected respectively. After subtraction of these two metrics, the timing error estimate value $e\,(i)$ is obtained.

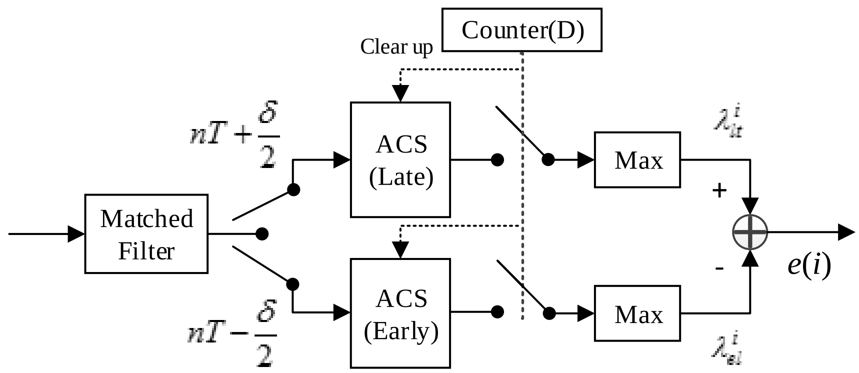

Figure 4 Timing error detector(TED) with early-late loop based on sequence detection.

CODE-AIDED TIMING SYNCHRONIZOR BASED ON EARLY-LATE LOOP

The max-log MAP has the 'ACS' process like the sequence detection, therefore the timing synchronization based on ELL can be applied to the Max-log MAP. Since the timing error detector in max-log MAP can be updated by the information from the decoding module of channel code, this timing scheme is called code-aided timing synchronization. The iterative receiver with code-aided ELL(CA-ELL) timing synchronization is shown in Figure 5.

The accumulated metrics of early branch and late branch in CA-ELL are:

$$\begin{aligned}\lambda_{\mathrm{el}}(i)&=\max_{\sigma}\left\{\left(\alpha_k^{\mathrm{el}}(\sigma)\Big|_{k=iD+1}^{k=iD+D}\right)\Big/D\right\}\\ \lambda_{\mathrm{lt}}(i)&=\max_{\sigma}\left\{\left(\alpha_k^{\mathrm{lt}}(\sigma)\Big|_{k=iD+1}^{k=iD+D}\right)\Big/D\right\}\end{aligned}\tag{7}$$

Where $\alpha_k^{\mathrm{el}}(\sigma)$ and $\alpha_k^{\mathrm{lt}}(\sigma)$ are forward recursions for early branch and late branch respectively as equation(1). Compared with the metric accumulating in sequence detection, the CA-ELL timing uses the prior information that can be updated iteratively. With the increase of the number of iterations, the reliability of the α values and the accumulated metrics will be improved gradually.

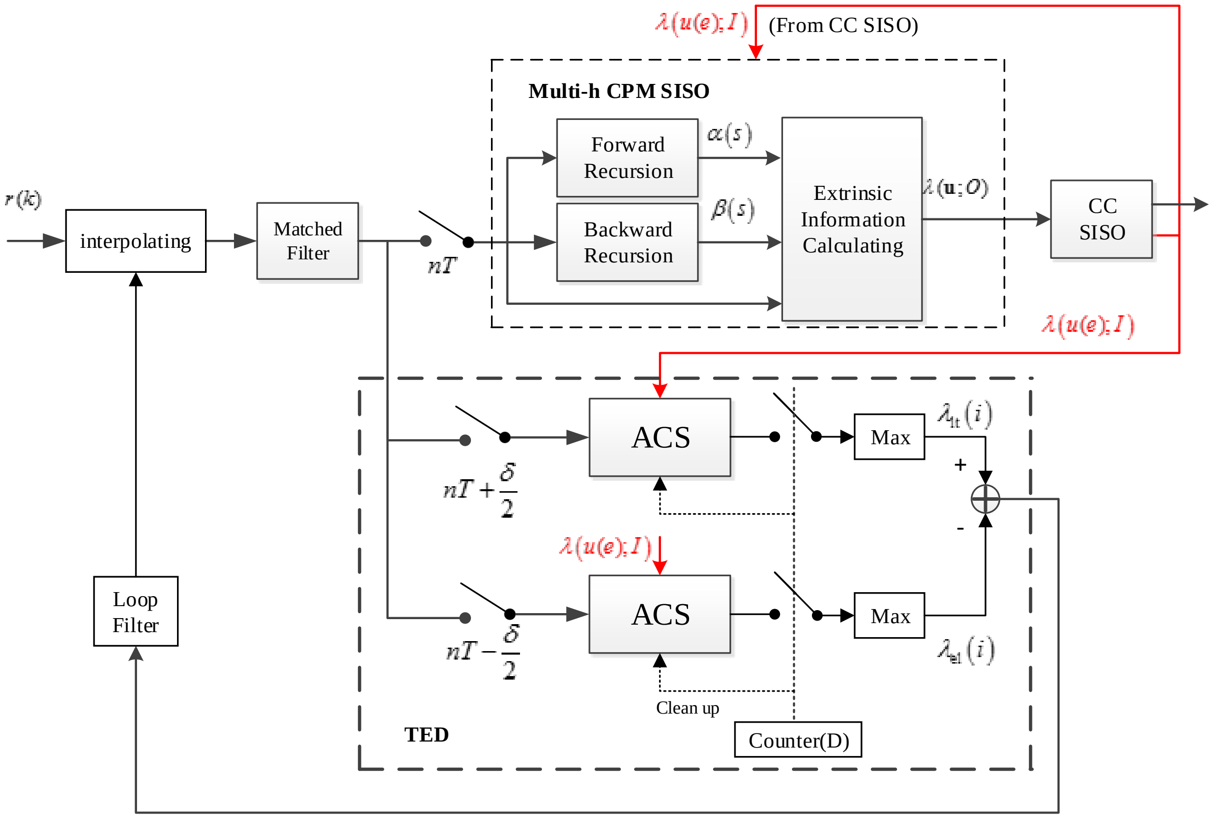

Figure 5 Timing loop with code-aided ELL for serially concatenated Multi-*h* CPM with a convolutional code

Figure 6 gave the iterative S-curves of proposed timing error estimator, which were obtained by 4 iterations with ARTM CPM signal in the case of Eb/No=3.5dB.

It can be seen from the iterative S-curves that, firstly, when the number of iterations increases, the slope of the S-curves at t=0 increases gradually, which means that the accuracy of timing estimation increases with the number of iterations; secondly, the estimating range of timing error remains in [-0.5T, 0.5T]; thirdly, the S-curve has only one zero-crossing point at t=0, so there is no false-lock point for the timing loop with the CA-ELL TED.

In order to reduce the complexity of the algorithm, a simplified scheme of frequency pulse truncation was adopted in this paper. Under the simplification of this algorithm, the state number of ARTM CPM could be reduced from 256 to 64, and the detection performance had almost no loss.

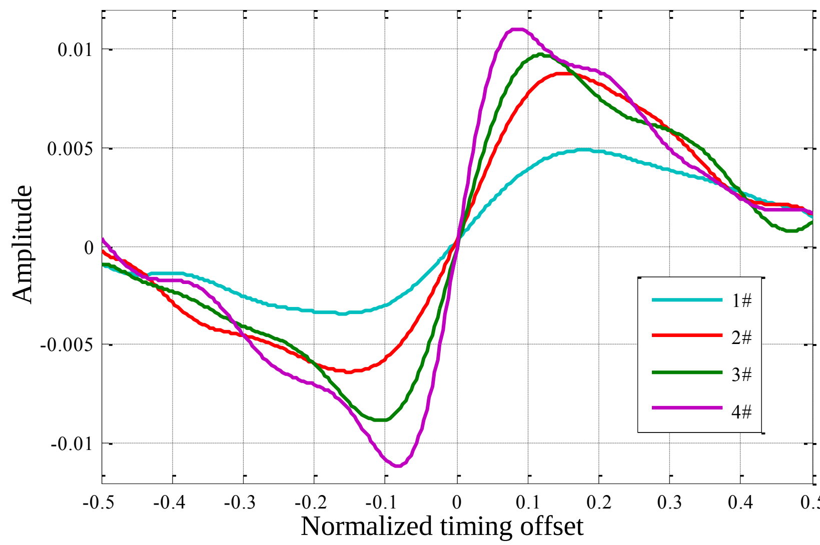

Figure 6 Iterative S-Curves with code-aided timing synchronization for ARTM CPM

SIMULATION RESULTS AND ANALYSIS

When Eb/No was 3dB to 5dB, the normalized variance of the timing estimation for ARTM CPM signals was shown in Figure 7. Among them, the code-aided timing synchronization algorithm based on EM algorithm and the lower bound of timing estimation (MCRB) were used as reference. The simulation parameters and conditions were set as follows: The Max-Log MAP algorithm used the way of sliding windows of which the window length was 64, the sampling offset δ is 0.25, the initialized timing error was set to 0.3*T*, the first order timing loop was adopted and whose normalized loop bandwidth was set to $B_L T$=0.001. After 10 iterations, the variance and BER of the last iteration were statistically counted and analyzed.

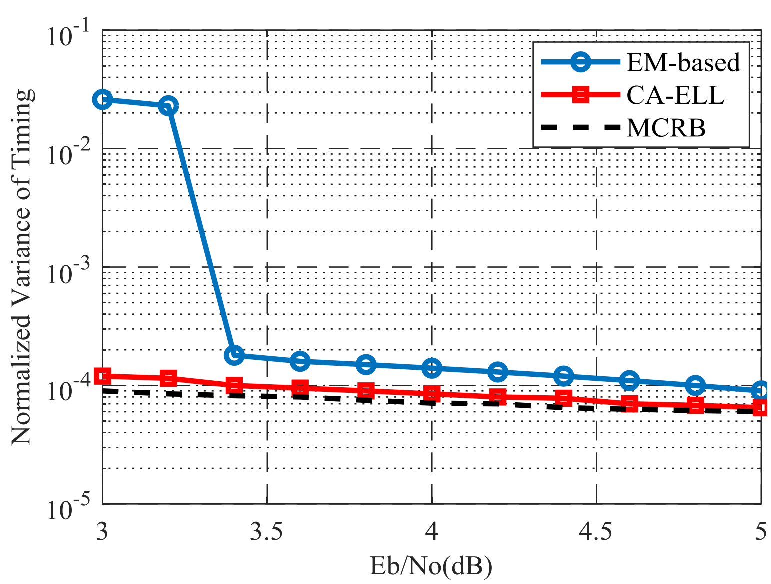

Figure 7 Normalized variance of proposed timing estimation for ARTM CPM1.

From Figure 7, we could see that these two code-aided timing algorithms could get closer to the MCRB in the region of Eb/No>3.4dB. However, when Eb/No was less than 3.4dB, the CA-ELL timing could always get close to the MCRB, while the EM-based algorithm got worse seriously. This was because the EM-based timing was based on the soft information from symbol-by-symbol decision (posterior probability information), but when Eb/No<3.4dB, the performance of this algorithm was deteriorated due to the poor reliability of the decision. On the other hand， the CA-ELL timing was based on the optimal sequence estimation instead of the symbol-by-symbol decision. Therefore, the reliability of proposed TED was higher in low SNR and CA-ELL timing performed better than the EM-based algorithm.

After the convergence of the timing loop, the corresponding BER curves were shown in figure 8, in which the BER performance with ideal synchronization was used as reference.

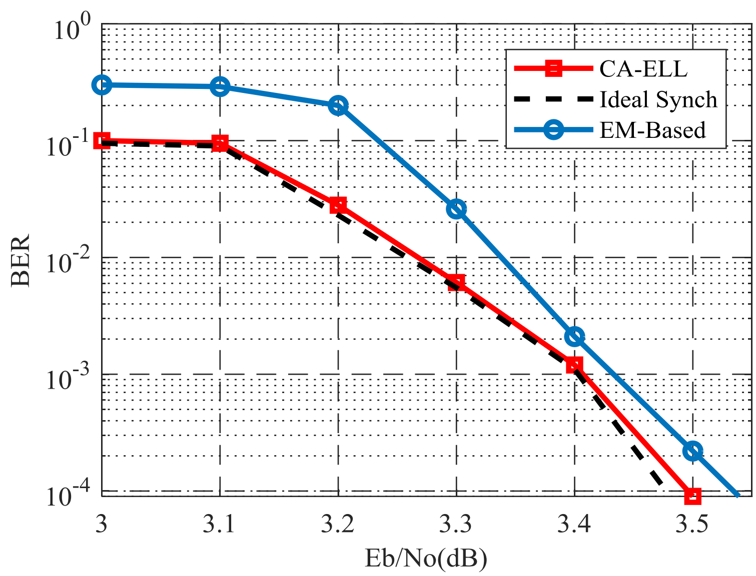

Figure 8 BER performance with timing synchronizations

As can be seen from figure 8, after the code-aided timing synchronization based on CA-ELL, the BER performance of the 10th iteration for ARTM CPM signal had only 0.03dB loss compared with the ideal synchronization(BER=10^{-4}). The performance loss of the BER based on the EM algorithm is about 0.08dB (BER=10^{-4}). But when Eb/No is less than 3.4dB, the deterioration of BER performance with EM-based timing is more serious than the CA-ELL-based timing, which is consistent with the results in Figure 7.

CONCLUSIONS

Based on the iterative detection of serially concatenated Multi-*h* CPM, a code-aided timing synchronization with early-late loop and max-log MAP detection is proposed in this paper. This timing method solves the synchronization problem of Multi-*h* CPM signals at low SNR. The simulation results of ARTM CPM show that the proposed code-aided timing synchronization can still be close to the MCRB of timing when Eb/No is low to 3dB, and the BER performance is only about 0.03dB loss compared to the ideal synchronization.

REFERENCES

[1] Range Commanders Council Telemetry Group, Range Commanders Council, White Sands Missile Range, New Mexico, *IRIG Standard 106-04: Telemetry Standards*, 2004.

[2] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*. New York & London: Plenum Press, 1997.

[3] A. N. D'Andrea, U. Mengali, and M. Morelli, "Symbol timing estimation with CPM modulation," *Communications, IEEE Transactions on*, vol. 44, no. 10, pp. 1362-1372, 1996.

[4] P. Chandran and E. Perrins, "Decision-Directed Symbol Timing Recovery for SOQPSK," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 2, pp. 781-789, 2009.

[5] E. Perrins, "A Timing False Lock Detector for M-ary Partial-Response CPM," *IEEE Wireless Communications Letters,* vol. 2, no. 6, pp. 671-674, 2013.

[6] X. Shunqin, X. Tao, Z. Sheng, Y. Chun, W. Shilian, and Z. Jian, "Timing synchronization based on early/late loop with multi-*h* continuous phase modulation," *Journal of National University of Defense Technology,* vol. 38, no. 4, pp. 113-118, 2016. (in Chinese)

[7] E. Perrins and M. Rice, "Reduced-complexity detectors for Multi-*h* CPM in aeronautical telemetry," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 43, no. 1, pp. 286-300, 2007.

[8] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis,design, and iterative decoding," *Information Theory, IEEE Transactions on*, vol. 44, no. 5, pp. 909-926, 1998.

[9] Z. Xiangchao, X. Rui, Z. Danfeng, and F. Fu, "Soft timing synchronization algorithm for CPM signals," presented at the 2013 IEEE International Conference on Green Computing and Communications (GreenCom) and IEEE Internet of Things (iThings) and IEEE Cyber, Physical and Social Computing (CPSCom), Beijing, China, 20-23 Aug., 2013.