Asynchronous Physical-layer Network Coding: Symbol Misalignment Estimation and Its Effect on Decoding

Yulin Shao, Soung Chang Liew, *Fellow, IEEE,* and Lu Lu, *Member, IEEE*

Abstract

In practical asynchronous physical-layer network coding (APNC) systems, the symbols from multiple transmitters to a common receiver may be misaligned. The knowledge of the amount of symbol misalignment, hence its estimation, is important to PNC decoding. This paper addresses the problem of symbol-misalignment estimation and the problem of optimal PNC decoding given the misalignment estimate, assuming the APNC system uses the root-raised-cosine pulse to carry signals (RRC-APNC). For symbol-misalignment estimation, we put forth an optimal estimator that is substantially more accurate than prior schemes. Our estimator makes use of double baud-rate samples, and hence is information-lossless since the double baud-rate samples capture all the information embedded in the continuous-time signal shaped by the assumed RRC pulse. Extensive simulations show that the mean-square-error (MSE) gains of our double baud-rate estimator over the baud-rate estimator are up to 8 dB. For PNC decoding, we devise optimal APNC decoders under inaccurate symbol-misalignment estimates. In particular, we prove and present a new whitening transformation to whiten the noise of the double baud-rate samples. Given the whitened samples, we construct a factor graph for a sum-product algorithm (SPA) to optimally decode the network-coded message. We further investigate the decoding performance of various estimation-and-decoding solutions for RRC-APNC. Simulation results indicate that: (i) A large roll-off factor of the RRC pulse is preferable in order to reduce decoder complexity while maintaining good performance. (ii) With a large roll-off factor, double baud-rate decoding is indispensable. Compared with a baud-rate sampling system (both estimation and decoding are based on baud-rate samples), a double baud-rate sampling system (both estimation and decoding are based on double baud-rate samples) yields 4.5 dB gains on symbol error rate (SER) performance in an AWGN channel, and 2 dB gains on packet error rate (PER) performance in a flat Rayleigh fading channel. (iii) Under a large roll-off factor, our double baud-rate estimator yields considerable improvements on

Y. Shao and S. C. Liew are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. Email: {sy016, soung}@ie.cuhk.edu.hk. L. Lu is with the Institute of Network Coding, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. Email: lulu@ie.cuhk.edu.hk.

decoding performance over the baud-rate estimator assuming the double baud-rate decoder is used in both cases. In an AWGN channel, the SER performance gains are up to 2 dB. In a flat Rayleigh fading channel, the PER performance gains are up to 1 dB.

Index Terms

Asynchronous PNC, pulse shaping, symbol-misalignment estimation, SPA, noise whitening.

I. INTRODUCTION

Physical-layer Network Coding (PNC) improves throughput in wireless networks by turning interference into useful network-coded messages [1], [2]. When operated with PNC, the information exchange of a two-way relay network (TWRN) is divided into two phases: the uplink phase and the downlink phase. In the uplink phase, two end nodes transmit their messages to a relay simultaneously, and the relay tries to decode a network-coded message based on the overlapped signals received from the two end nodes. In the downlink phase, the relay broadcasts the network-coded message to both end nodes, and each end node extracts the other node's message using its pre-stored self-message and the received network-coded message [3].

PNC with asynchrony is termed Asynchronous PNC (APNC) [4]. APNC does away with tight synchronization of the transmission times of the two end nodes. As a result, the transmitted signals from the two end nodes may arrive at the relay with a relative time offset (i.e., symbol misalignment). Recently, APNC that uses the root-raised-cosine (RRC) pulse has attracted much attention. Compared with APNC using the rectangular pulse [3]–[5], APNC with the RRC pulse (RRC-APNC) is more spectrally efficient.

Research efforts on RRC-APNC have focused on two issues critical to its inner workings, namely, accurate symbol-misalignment estimation [6]–[8] and optimal PNC decoding under Inter-Symbol Interference (ISI) [9], [10]. This paper makes advances on the solutions to these two problems, as elaborated below.

Symbol-misalignment estimation – Estimating the symbol misalignment is crucial in APNC because accurate estimate is required for good PNC decoding performance. Symbol misalignment estimators for RRC-APNC based on oversampling were proposed in [6] and [7]. These estimators are highly complex owing to the oversampling. In [7], simulations showed that as much as 8-times oversampling is needed for accurate estimation. Our earlier work [8] presented a maximum likelihood (ML) estimator based on baud-rate sampling (i.e., no oversampling). This baud-rate

estimator was shown to be able to achieve estimation accuracy comparable to that of the 8-times oversampling estimators in [6] and [7].

An issue left open by [8] is that baud-rate sampling may not capture all the symbol misalignment information contained in the received signal when the RRC pulse is used. According to the Nyquist-Shannon sampling theorem, the Nyquist rate is the lower bound on the sampling rate that can guarantee no information loss when converting the continuous-time signal into the discrete-time signal (i.e., mathematically, the discrete-time signal can be converted back to the continuous-time signal exactly) [11]. For the signals shaped by the raised-cosine (RC) pulse or the RRC pulse, the Nyquist rate is between the baud-rate and the double baud-rate (i.e., it is defined by a roll-off factor β within the range of $0 \leq \beta \leq 1$). This observation suggests that (i) the baud-rate estimator in [8] may still be suboptimal and further improvement is possible with double baud-rate sampling for RRC-APNC; (ii) oversampling beyond the double-baud rate is not necessary because all signal information has already been captured with double baud-rate sampling.

With the above motivations, this paper puts forth an ML-optimal estimator that estimates symbol misalignment based on double baud-rate samples of the preamble. Simulation results show that this double baud-rate estimator substantially improves the symbol-misalignment estimation accuracy over the prior baud-rate estimator [8]. The mean-square-error (MSE) performance gains of the double baud-rate estimator over the baud-rate estimator are up to 8 dB under both AWGN and Rayleigh fading channels.

Optimal decoders under ISI – In APNC, following the symbol-misalignment estimator, a PNC decoder will then make use of the estimated symbol misalignment to perform PNC decoding. For a conventional point-to-point system using the RRC pulse, the optimal decoder is a symbol-by-symbol decoder since we could always sample at specific positions to eliminate the ISI [11]. However, for RRC-APNC, when the symbol boundaries of the two end nodes are not aligned, ISI is inevitable because there are no ISI-free sampling positions for both signals. Decoders designed for the ISI channel in RRC-APNC can be applied to harness useful information contained in the ISI. Two examples are the BCJR decoder in [9] and the MLSE decoder in [10]. In these papers, perfect knowledge of the symbol misalignment is assumed as prior information to the receiver, and hence the decoders are designed under an ideal setup, i.e., all the samples are collected at the exact symbol boundaries of one of the two end nodes. However, in practice, the symbol misalignment needs to be estimated and the exact symbol

boundaries may not be known perfectly.

To fill this gap, our paper here investigates the design of optimal RRC-APNC decoders in the presence of inaccurate symbol misalignment estimates. We construct a factor graph with a tree structure, from which a sum-product algorithm (SPA) can be devised to optimally decode the network-coded messages. In particular, for the APNC decoder using double baud-rate samples, we prove and present a new whitening transformation to whiten the colored noise. This whitening transformation involves only matrix decomposition and multiplication, hence is linear and computationally efficient.

Integration of symbol-misalignment estimation and PNC decoding – Prior work on APNC either focused on the issue of symbol-misalignment estimation or the issue of PNC decoding. The overall APNC system that takes into account the effect of symbol-misalignment estimation on PNC decoding was not studied. To fill this gap, this paper studies various integrated estimation-and-decoding solutions for RRC-APNC. Specifically, there are four possible solutions at the receiver: baud-rate estimation and baud-rate decoding (solution I); double baud-rate estimation and baud-rate decoding (solution II); baud-rate estimation and double baud-rate decoding (solution III); double baud-rate estimation and double baud-rate decoding (solution IV). Our comprehensive simulations on these estimation-and-decoding solutions indicate that: (i) Practical RRC-APNC demands a large roll-off factor to enable feasible decoding when complexity is taken into account. (ii) Under a large roll-off factor, double baud-rate decoding is indispensable for RRC-APNC. When $\beta = 1$, solution IV yields 4.5 dB symbol error rate (SER) gains over solution I in an AWGN channel, and 2.5 dB packet error rate (PER) gains over solution I in a flat Rayleigh fading channel. (iii) Under a large roll-off factor, solution IV yields considerable improvements on decoding performance over solution III. When $\beta = 1$, the SER performance gains are up to 2 dB in an AWGN channel, the PER performance gains are up to 1 dB in a flat Rayleigh fading channel.

The remainder of this paper is organized in the following manner. Section II presents the integrated estimation-decoding framework for RRC-APNC. Section III details the baud-rate and double baud-rate symbol-misalignment estimators. Section IV devises the optimal RRC-APNC decoders under both baud-rate and double baud-rate sampling. Numerical results are also presented in Section III and IV. Section V concludes this paper. Throughout the paper, lowercase bold letters represent vectors while uppercase bold letters represent matrices.

II. SYSTEM MODEL

We study a TWRN operated with PNC, where two end nodes A and B exchange information with the help of a relay R. Time domain RRC pulses are adopted by both nodes A and B. The system model is shown in Fig.1. In the PNC uplink, nodes A and B transmit signals to the relay R simultaneously, and there is a symbol misalignment between the signals of nodes A and B at relay R due to the imperfect synchronization of the arrival times of their symbols (i.e., the boundaries of the symbols from A and B are not aligned) [4].
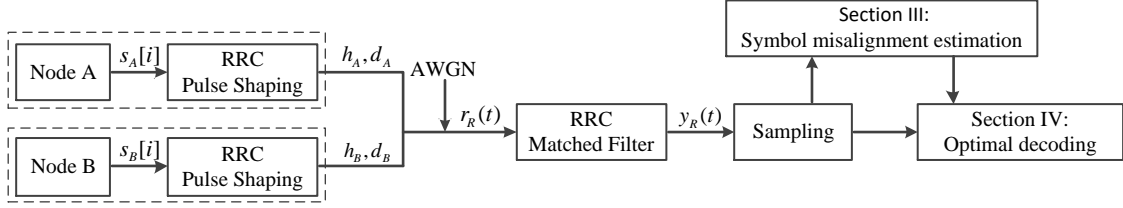

Fig. 1. System model for the uplink of a TWRN operated with PNC where the symbols of nodes A and B arrive at the relay R with a relative time offset (i.e., symbol misalignment). We assume nodes A and B use practical RRC pulse shapes.

We denote by s_A and s_B the symbols to be transmitted at nodes A and B. After going through the RRC pulse-shaping filter, the continuous-time baseband signal at each node can be expressed as

$$x_A(t) = \sum_i s_A[i]p'(t - iT), \tag{1}$$

$$x_B(t) = \sum_i s_B[i]p'(t - iT), \tag{2}$$

where T is the symbol duration, and $p'(t)$ is an RRC pulse with a roll-off factor $\beta : 0 \leq \beta \leq 1$.

At relay R, the received overlapping baseband continuous-time signal is given by

$$r_R(t) = h_A x_A(t - d_A) + h_B x_B(t - d_B) + n_R(t)$$
$$= h_A \sum_i s_A[i]p'(t - iT - d_A) + h_B \sum_i s_B[i]p'(t - iT - d_B) + n_R(t), \tag{3}$$

where h_A and h_B are the channel gains (in this paper, we assume slow-fading channel, and hence h_A and h_B are constant values within one packet); $n_R(t)$ is the zero-mean complex additive white Gaussian noise (AWGN) with double sided power spectral density $N_0/2$; d_A and d_B are the delays of signals $x_A(t)$ and $x_B(t)$ from nodes A and B to the relay R, and the symbol misalignment is given by $\tau = d_A - d_B$.

The received signal $r_R(t)$ is first passed through an RRC matched filter to get rid of the out-of-band noise, giving

$$
\begin{aligned}
y_R(t) &= r_R(t) * p'(t) \\
&= h_A \sum_i s_A[i]p(t - iT - d_A) + h_B \sum_i s_B[i]p(t - iT - d_B) + w_R(t),
\end{aligned}
\tag{4}
$$

where $*$ denotes the continuous-time convolution operation, $p(t)$ is the raised-cosine (RC) pulse (which is the effective pulse shape when one convolves two RRC pulses), $w_R(t)$ is the low-pass-filtered Gaussian noise term.

Given the continuous-time outputs of the RRC matched filter, we investigate two sampling schemes in this paper to obtain the discrete-time signal model.

(i) Baud-rate sampling. We sample $y_R(t)$ at baud-rate on equal-space positions $t = nT$, $n = 1, 2, ...$ The samples are given by

$$
\begin{aligned}
y^b[n] &= y_R(t = nT) \\
&= h_A \sum_i s_A[i]p(nT - iT - t_A) + h_B \sum_i s_B[i]p(nT - iT - t_B) + w^b[n] \\
&\stackrel{(a)}{=} h_A \sum_l s_A[n-l]p(lT - t_A) + h_B \sum_l s_B[n-l]p(lT - t_B) + w^b[n],
\end{aligned}
\tag{5}
$$

where t_A and t_B are the time offsets with respect to the sampling positions[1] and the relative time offset $\tau = d_A - d_B = t_A - t_B$; step (a) follows by defining $l = n - i$; $\boldsymbol{w^b}$ is independent identically distributed (i.i.d.) Gaussian noise, $w^b[n] \sim CN(0, \sigma^2)$ and $\sigma^2 = N_0/2T$.

(ii) Double baud-rate sampling. We sample $y_R(t)$ at double baud-rate on equal-space positions $t = kT/2, k = 1, 2, ...$ The samples are given by

$$
\begin{aligned}
y^d[k] &= y_R(t = \frac{k}{2}T) \\
&= h_A \sum_i s_A[i]p(\frac{k}{2}T - iT - t_A) + h_B \sum_i s_B[i]p(\frac{k}{2}T - iT - t_B) + w^d[k],
\end{aligned}
\tag{6}
$$

where t_A and t_B are as defined in the baud-rate sampling case; $\boldsymbol{w^d}$ is colored noise, and each term $w^d[k] \sim CN(0, \sigma^2)$.

---

[1]In (5), we realign the time so that the sampling positions are with respect to the time kept by the receiver clock. Specifically, we sample $y_R(t)$ at $t = nT, n = 1, 2, ...$, and the boundaries of symbols $s_A[n]$ and $s_B[n]$ with respect to the n-th sampling positions are now t_A and t_B, respectively.

We emphasize that the sampling process is a critical step in the RRC-APNC system, since it determines the amount of information we could obtain from the original continuous-time signal. When sampling the signals shaped by RC or RRC pulse, e.g., the RC pulse-shaped signal $y_R(t)$, baud-rate sampling is of low complexity but loses information when $\beta \neq 0$; while double baud-rate sampling guarantees no information-loss for all β within the range $0 \leq \beta \leq 1$.

The objective of relay R is to optimally decode the network-coded (XOR) information. To this end, two issues need to be addressed: (i) estimate the symbol misalignment accurately; (ii) devise an optimal decoder to find the most likely $s_{A \oplus B}[n]$ given by

$$s_{A \oplus B}[n] = \arg \max_{s_{A \oplus B}[n]} \sum_{\substack{s_A[n], s_B[n]: \\ s_A[n] \oplus s_B[n] = s_{A \oplus B}[n]}} \Pr\left\{s_A[n], s_B[n] \mid \boldsymbol{y}^b \text{ or } \boldsymbol{y}^d\right\}. \tag{7}$$

Estimating symbol misalignment is essentially estimate the unknown time offsets t_A and t_B in $\boldsymbol{y}^b$ and $\boldsymbol{y}^d$. This problem can be addressed by the application of two carefully designed preambles. These two preambles are assigned to nodes A and B, respectively, and are transmitted along with the data in each packet. At the receiver, we sample the overlapped preambles embedded in $y_R(t)$ and make use of the samples to estimate t_A and t_B. The optimal estimators based on the two sampling schemes are detailed in Section III.

Given the estimated symbol misalignment and the samples corresponding to the data part, we investigate the design of optimal RRC-APNC decoders in the presence of inaccurate symbol misalignment estimates in Section IV. Specifically, we construct a factor graph with a tree structure, from which a sum-product algorithm (SPA) can be devised to optimally decode the network-coded messages following the metric in (7).

III. OPTIMAL SYMBOL-MISALIGNMENT ESTIMATORS

Estimating symbol misalignment is essentially a problem of estimating unknown parameters in the presence of random noise. An optimal estimator comes is the ML estimator [11]. This section presents carefully designed preambles for symbol-misalignment estimation and derives the ML-optimal estimators under baud-rate sampling and double baud-rate sampling of the preambles.

A. Preamble structure

The preambles assigned to nodes A and B have the same structure as shown in Fig.2. The preamble consists of a Q-points Zadoff-Chu (ZC) sequence along with a G-points cyclic prefix

and a G-points cyclic suffix. ZC sequences are constant amplitude zero auto-correlation (CAZAC) sequences. They have a favorable auto-correlation property, that is, the auto-correlation of a ZC sequence with a cyclic-shifted version of itself is zero (i.e., any non-zero integer cyclic shift of the ZC sequence is orthogonal to the original ZC sequence). We assign cyclic-shifted ZC sequences to nodes A and B. Specifically, the ZC sequence assigned to node A (denote by z_A) is a $\lfloor Q/2 \rfloor$-points cyclic shifted version of that assigned to node B (denoted by z_B). The cyclic prefix and cyclic suffix are to enable the computation of the cyclic cross-correlation in the presence of symbol misalignment.

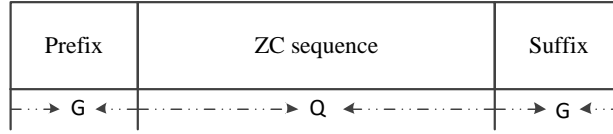| Prefix	ZC sequence	Suffix
G	Q	G

Fig. 2. The structure of the preambles assigned to nodes A and B. It consists of a ZC sequence, a cyclic prefix and a cyclic suffix.

This paper only considers fractional symbol misalignment, namely, $\tau \in [-T/2, T/2)$. As indicated in [8], the integral symbol misalignment can be estimated separately and it affects the performance of an ML estimator only when $E_b/N_0 < 0$ dB (i.e., there is essentially no error in the integral symbol misalignment estimate when $E_b/N_0 \geq 0$ dB). At the relay, the overlapping preambles are embedded in $y_R(t)$. Thus, we sample $y_R(t)$ at baud-rate or double baud-rate and utilize the samples corresponding to the preamble part to estimate the symbol misalignment. In the following, we first review the baud-rate estimator [8], to get a preliminary understanding towards the ML estimation. Then, we present our double baud-rate estimator.

B. Baud-rate estimator

The signal processing flow of the baud-rate estimator is shown in Fig.3. As can be seen, given the baud-rate samples of $y_R(t)$ in (5), we cross-correlate y^b with the conjugate of the local Q-points ZC sequences, i.e. z_A^* and z_B^*, respectively. Then, based on the cross-correlation outputs, the ML estimator can be designed.

In the following, we focus on the cross-correlation between y^b and z_A^*, and the cross-correlation between y^b and z_B^* can be derived in the same way. The cross-correlation outputs
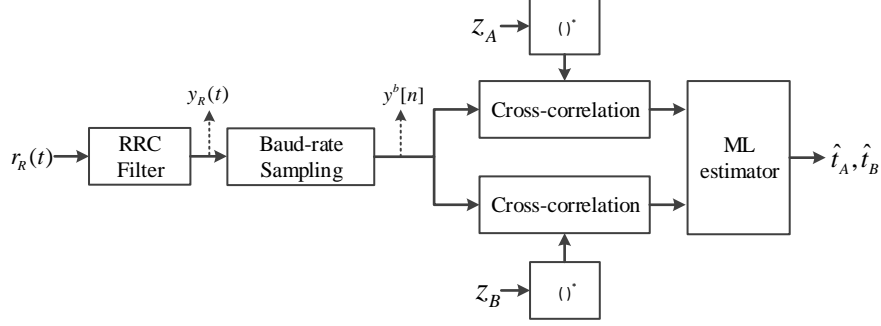
Fig. 3. Signal processing flow for the symbol-misalignment estimation under baud-rate sampling.

are given by

$$c_A^b[m] = \sum_{n=0}^{Q-1} z_A^*[n]y^b[n+m] \tag{8}$$

$$= h_A \sum_{l} p(lT - t_A) \sum_{n=0}^{Q-1} z_A^*[n]s_A[n+m-l] + h_B \sum_{l} p(lT - t_B) \sum_{n=0}^{Q-1} z_A^*[n]s_B[n+m-l] + \widetilde{w}_A^b[m],$$

where $\widetilde{w}_A^b[m] = \sum_{n=0}^{Q-1} z_A^*[n]w^b[n+m]$; the transmitted preambles are contained in s_A and s_B, respectively.

Thanks to the auto-correlation property of the ZC sequence, when the local z_A^* is aligned to the transmitted z_A embedded in s_A, a peak emerges. Thus, we could detect the peak and locate the position of z_A in s_A. In particular, the peak index is given by $I_A = \arg\max_m c_A^b[m]$. To ease exposition, we redefine the peak index as 0 by offsetting $-I_A$ points if necessary, namely, we assume the peak emerges on $c_A^b[0]$.

In this context, if we only consider a small range $-d < m \le d$ in c_A^b, where $Q \gg d$, the second term on the R.H.S. of (8) can be approximated as 0. The reasons are as follows: when $m - l \ne \lfloor Q/2 \rfloor$, the orthogonality of z_A and z_B makes it 0; when $m - l = \lfloor Q/2 \rfloor$ (namely, $z_A[n] = z_B[n+m-l]$), l must be large since $-d < m \le d$, $Q \gg d$, and hence $p(lT - t_B)$ would be small so that this term can still be ignored. Thus, (8) can be simplified as

$$c_A^b[m] \approx h_A \sum_{l} p(lT - t_A) \sum_{n=0}^{Q-1} z_A^*[n]s_A[n+m-l] + \widetilde{w}_A^b[m]$$

$$\overset{(a)}{=} h_A Q p(mT - t_A) + \widetilde{w}_A^b[m], \tag{9}$$

where $-d < m \le d$; step (a) follows by retaining the term where $l = m$ only thanks to the orthogonality of z_A and its cyclic shift. Moreover, we specify that the auto-covariance function

of the noise sequence $\widetilde{w}_A^b$ is given by

$$
\begin{aligned}
\mathbb{E}\left[\widetilde{w}_A^b[m]^* \widetilde{w}_A^b[m+j]\right] &= \mathbb{E}\left[\sum_{n=0}^{Q-1} z_A[n] w^b[n+m]^* \sum_{n'=0}^{Q-1} z_A^*[n'] w^b[n'+m+j]\right] \\
&= \sigma^2 \sum_{n=0}^{Q-1} \sum_{n'=0}^{Q-1} z_A[n] z_A^*[n'] \delta(n'+j-n) \\
&\stackrel{(a)}{=} \sigma^2 \sum_{n=\max\{0,j\}}^{\min\{Q-1,Q+j-1\}} z_A[n] z_A^*[n-j] \\
&\stackrel{(b)}{\approx} Q\sigma^2 \delta(j),
\end{aligned}
\tag{10}
$$

where step (a) follows by setting $n' = n - j$, step (b) follows since $Q \gg j$.

The key implication from Eq.(10) is that $\widetilde{w}_A^b$ can be approximated as i.i.d. Gaussian noise, and $\widetilde{w}_A^b[m] \sim CN(0, Q\sigma^2)$. Thus, we could directly perform ML estimation to estimate t_A based on (9), because the ML metric will reduce to the minimum Euclidean distance metric [12]. The estimated t_A is given by

$$
\widehat{t}_A = \arg\min_{t_A} \sum_{m=1-d}^{d} \left| c_A^b[m] - h_A Q p(mT - t_A) \right|^2.
\tag{11}
$$

Similarly, we cross-correlate $\boldsymbol{y}^b$ and $\boldsymbol{z}_B^*$. Under the same assumption that $-d < m \leq d$, we have

$$
c_B^b[m] \approx h_B Q p(mT - t_B) + \widetilde{w}_B^b[m],
\tag{12}
$$

where $\widetilde{w}_B^b$ can also be approximated as i.i.d. Gaussian noise and $\widetilde{w}_B^b[m] \sim CN(0, Q\sigma^2)$. Thus, we could perform ML estimation to estimate t_B based on (12), giving

$$
\widehat{t}_B = \arg\min_{t_B} \sum_{m=1-d}^{d} \left| c_B^b[m] - h_B Q p(mT - t_B) \right|^2.
\tag{13}
$$

Finally, the estimated symbol misalignment is given by $\widehat{\tau} = \widehat{t}_A - \widehat{t}_B$.

Remark. The advantage of this baud-rate estimator is the low complexity compared with the oversampling scheme proposed in [6] and [7]. However, for the signal shaped by the RC or RRC pulse, baud-rate is an undersampling rate when $\beta > 0$. In other words, baud-rate is not sufficient to capture all the information contained in the original continuous-time signal relevant to estimating t_A and t_B. A natural question is then whether the estimation can be further improved with a higher sampling rate? This motivates the design of our double baud-rate estimator in Subsection C.

C. Double baud-rate estimator

According to the Nyquist-Shannon sampling theorem, the Nyquist rate is the lower bound on the sampling rate that can guarantee no information loss when converting the continuous-time signal into the discrete-time signal (and vice versa) [11]. For the signals shaped by the RC or RRC pulse, the Nyquist rate is between baud-rate and double baud-rate (i.e., it is defined by a roll-off factor β within the range of $0 \leq \beta \leq 1$). This implies that double baud-rate sampling can capture all signal information embedded in $y_R(t)$ while averting unnecessary redundancy introduced by excess oversampling beyond double baud-rate sampling (e.g., 8-times sampling in [7]). Thus, in this subsection, we take the double baud-rate samples in (6) as the inputs and devise an ML-optimal estimator.

The signal processing flow of our double baud-rate estimator is shown in Fig.4(a), where the double baud-rate samples will first be cross-correlated with the local "interpolated double baud-rate ZC sequences". A continuous-time ZC sequence is the result of modulating an original discrete-time ZC sequence using a certain continuous-time pulse, e.g., the $Sinc(t)$ pulse in Fig.4(a). The "$2\times$ interpolation" means the resulting discrete-time ZC sequence after sampling the continuous-time ZC sequence at double baud-rate. Thus, the "interpolated double baud-rate ZC sequence" has twice the samples of the original ZC sequence (basically, it is the original sequence with a zero inserted between two ZC samples given than $Sinc(t)$ is assumed).

A new difficulty in the analysis of Fig.4(a) is that, the noise term in the double baud-rate samples, i.e., $\boldsymbol{w}^d$ in (6), is no longer white. To facilitate analysis, we put forth an equivalent signal processing scheme as shown in Fig.4(b). Two distinctions between the flow diagrams in Fig.4(a) and Fig.4(b) are as follows:

(i) The RRC matched filter on the front-end of Fig.4(a) is replaced with a $Sinc(2t)$ filter in Fig.4(b). In this case, the noise terms in the samples would be white, and the analyses afterwards could be simplified.

(ii) In Fig.4(b), given the $Sinc(2t)$ filter on the front-end, the noise power in each sample is doubled (i.e., $2\sigma^2$). Thus, a digital RRC filter is needed to filter the out-of-band noise. In Fig.4(b), the ZC sequences $\boldsymbol{z_A}$ and $\boldsymbol{z_B}$ are twice interpolated using the RRC filter rather than the $Sinc(t)$ filter in Fig.4(a).[2] This implies that the cross-correlation operation in Fig.4(b)

---

[2]Accordingly, an interpolated sample between two original ZC samples are nonzero in Fig.4(b) due to the use of RRC interpolation rather than the SINC interpolation in Fig.4(a).

(a) Original processing flow (analog matched filtering)

(b) Equivalent processing flow (digital matched filtering)
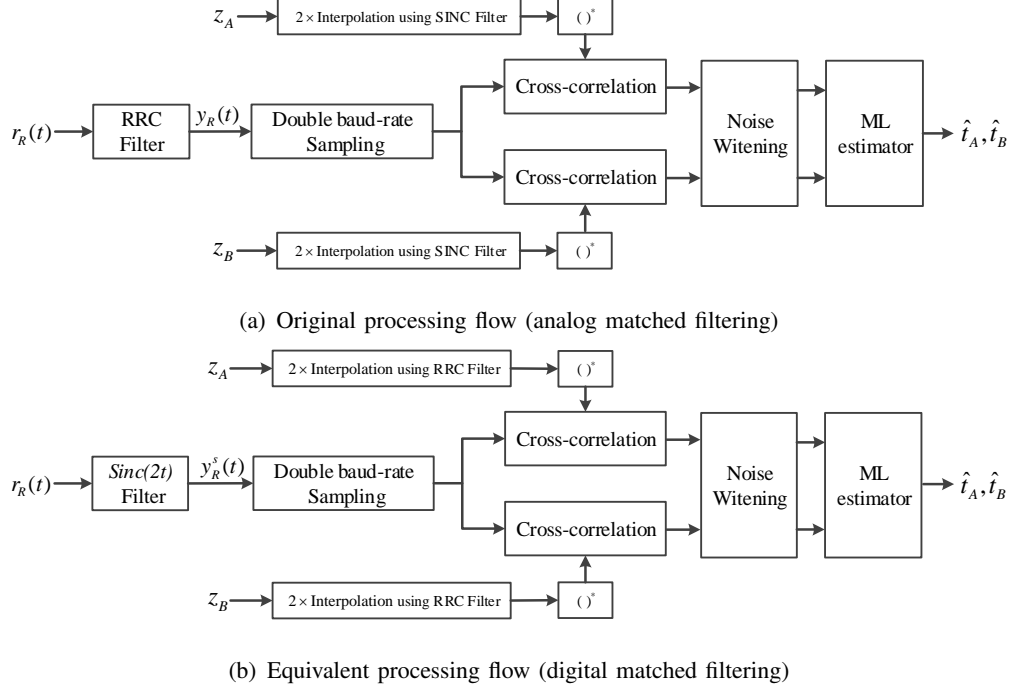
Fig. 4. Signal processing flows for the symbol-misalignment estimation under double-baud-rate sampling.

also embeds a process of digital matched filtering (i.e., the analog matched filtering in Fig.4(a) is moved to the digital domain in Fig.4(b)).

Essentially, the transformation from Fig.4(a) to Fig.4(b) is a conversion from analog matched filtering to digital matched filtering. Thus, these two flows are equivalent. In particular, after cross-correlation, the signals from the two flows are identical.

In the following, we detail the signal processing flow in Fig.4(b) step by step. Specifically, we explain how to utilize the double baud-rate samples to obtain more accurate estimation of τ.

1) Sampling: The received signal $r_R(t)$ is first fed to a $Sinc(2t)$ filter, giving

$$
\begin{aligned}
y_R^s(t) &= r_R(t) * Sinc(2t) \\
&= h_A \sum_i s_A[i] p'(t - iT - d_A) + h_B \sum_i s_B[i] p'(t - iT - d_B) + w_R^s(t),
\end{aligned} \tag{14}
$$

where $w_R^s(t)$ is the low-pass-filtered Gaussian noise term and the power of $w_R^s(t)$ is doubled compared with that of $w_R(t)$ in (4).

Next, we sample (14) at double baud-rate, and the samples are given by

$$
\bar{y}^d[k] = y_R^s(t = \frac{k}{2}T)
$$

$$= h_A \sum_i s_A[i]p'(\frac{k}{2}T - iT - t_A) + h_B \sum_i s_B[i]p'(\frac{k}{2}T - iT - t_B) + \bar{w}^d[k], \quad (15)$$

where $\bar{w}^d$ is now i.i.d. Gaussian noise, $\bar{w}^d[k] \sim CN(0, 2\sigma^2)$.

Meanwhile, we "double-interpolate" the local ZC sequences z_A, z_B using the RRC filter. The interpolated ZC sequences, denoted by z_A^d, z_B^d, are given by

$$z_A^d[k] = \sum_{i=0}^{Q-1} z_A[i]p'(\frac{k}{2}T - iT), \quad (16)$$

$$z_B^d[k] = \sum_{i=0}^{Q-1} z_B[i]p'(\frac{k}{2}T - iT), \quad (17)$$

2) *Cross-correlation:* Next, we cross-correlate $z_A^{d\,*}, z_B^{d\,*}$ with $\bar{y}^d$, respectively. Focusing on the the cross-correlation between $z_A^{d\,*}$ and $\bar{y}^d$, we have

$$c_A^d[m] = \sum_{k=0}^{2Q-1} z_A^{d\,*}[k]\bar{y}^d[k+m]$$

$$\overset{(a)}{=} Qh_A \sum_l \left[p'(lT)p'(\frac{m}{2}T + lT - t_A) + p'(lT + \frac{T}{2})p'(\frac{m+1}{2}T + lT - t_A) \right] + \widetilde{w}_A^d[m]$$

$$\overset{\text{def}}{=} v_A^d[m] + \widetilde{w}_A^d[m], \quad (18)$$

where step (a) involves a number of steps, the details of which are given in Appendix A. We further prove in Appendix A that, the noise term $\widetilde{w}_A^d$ is colored overall, but its odd and even entries are white among themselves. As stated above, the outcomes of cross-correlation in Fig.4(a) and Fig.4(b) are identical. Thus, (18) is also the cross-correlation outputs in Fig.4(a).

If we directly perform ML estimation on (18), the ML metric would be quiet complicated and there may be no analytical solution owing to the colored noise. Thus, we will first whiten the noise, and then construct the ML metric based on the whitened sequence.

3) *Noise whitening:* To whiten the noise term $\widetilde{w}_A^d$, it is important to study its covariance matrix $\Sigma(\widetilde{w}_A^d)$, since noise whitening is a process that transform $\Sigma(\widetilde{w}_A^d)$ to a diagonal matrix. We show in Appendix A that, if we only consider a small range $-d < m \leq d$ ($Q \gg d$), the auto-covariance function of $\widetilde{w}_A^d$ can be expressed as

$$\mathbb{E}\left[\widetilde{w}_A^d[m]^* \widetilde{w}_A^d[m+j]\right] \approx 2Q\sigma^2 p(\frac{j}{2}T), \quad (19)$$

where the approximation comes from the infinite time continuation of the RRC pulse and the limited prefix/suffix length.

Given (19), the covariance matrix of $\widetilde{w}_A^d$ can be written as

$$\boldsymbol{\Sigma}(\widetilde{w}_A^d) = 2Q\sigma^2\boldsymbol{\Sigma_0} = 2Q\sigma^2 \begin{bmatrix} 1 & p(\frac{1}{2}T) & 0 & p(\frac{3}{2}T) & \cdots & p(\frac{2d-1}{2}T) \\ p(-\frac{1}{2}T) & 1 & p(\frac{1}{2}T) & 0 & \cdots & 0 \\ 0 & p(-\frac{1}{2}T) & 1 & p(\frac{1}{2}T) & \cdots & p(\frac{2d-3}{2}T) \\ p(-\frac{3}{2}T) & 0 & p(-\frac{1}{2}T) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p(\frac{1-2d}{2}T) & 0 & p(\frac{3-2d}{2}T) & 0 & \cdots & 1 \end{bmatrix}. \tag{20}$$

Thus, if we could transform the matrix $\boldsymbol{\Sigma_0}$ to a diagonal matrix, the noise is whitened naturally.

A possible transformation is to use Cholesky decomposition. Cholesky decomposition makes use of the fact that any real-valued symmetric positive-definite matrix can be factorized into the product of a lower triangular matrix and its transpose [13]. In the following, we show first why $\boldsymbol{\Sigma_0}$ can be Cholesky decomposed, then how the Cholesky decomposition can help whiten the noise.

First, it is obvious that $\boldsymbol{\Sigma_0}$ is real and symmetric since $p(t)$ is an even function. Meanwhile, we prove in Appendix B that $\boldsymbol{\Sigma_0}$ is positive definite. Thus, $\boldsymbol{\Sigma_0}$ can be Cholesky decomposed and we have $\boldsymbol{\Sigma_0} = \boldsymbol{U_0^T U_0}$, where $\boldsymbol{U_0}$ is an invertible upper triangular matrix (its diagonal entries are nonzero).

Then, considering the small range $-d < m \leq d$ $(Q \gg d)$, we could rewrite (18) in vector form as

$$c_A^d = v_A^d + \widetilde{w}_A^d \tag{21}$$

where $c_A^d = \left[c_A^d[1-d], ..., c_A^d[d]\right]^T$, $v_A^d = \left[v_A^d[1-d], ..., v_A^d[d]\right]^T$ and $\widetilde{w}_A^d = \left[\widetilde{w}_A^d[1-d], ..., \widetilde{w}_A^d[d]\right]^T$. Next, we left multiply (21) by $\boldsymbol{U_0}^{-T}$, giving

$$\boldsymbol{U_0}^{-T}c_A^d = \boldsymbol{U_0}^{-T}v_A^d + \boldsymbol{U_0}^{-T}\widetilde{w}_A^d \qquad \overset{\text{def}}{=} \qquad \ddot{c}_A = \ddot{v}_A + \ddot{w}_A. \tag{22}$$

The transformed noise $\ddot{w}_A$ in (22) is now i.i.d. Gaussian noise since $\mathbb{E}\left[\ddot{w}_A\ddot{w}_A^H\right] = 2Q\sigma^2 \boldsymbol{U_0}^{-T}\boldsymbol{\Sigma_0}\boldsymbol{U_0}^{-1} = 2Q\sigma^2\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. Thus, (22) is exactly the whitened discrete-time model corresponding to (21).

Similarly, we cross-correlate $z_B^{d\,*}$ with $\bar{y}^d$ and whiten the cross-correlation results. The corresponding whitened model can be written as

$$\ddot{c}_B = \ddot{v}_B + \ddot{w}_B, \tag{23}$$

where $\ddot{c}_B$, $\ddot{v}_B$ and $\ddot{w}_B$ are the counterparts of $\ddot{c}_A$, $\ddot{v}_A$ and $\ddot{w}_A$ in (22), respectively.

4) ML estimation: Finally, based on the whitened model in (22) and (23), we could directly construct the metrics for ML estimation as follows:

$$\widehat{t}_A = \arg\min_{t_A} \mid \ddot{\boldsymbol{c}}_{\boldsymbol{A}} - \ddot{\boldsymbol{v}}_{\boldsymbol{A}} \mid^2, \tag{24}$$

$$\widehat{t}_B = \arg\min_{t_B} \mid \ddot{\boldsymbol{c}}_{\boldsymbol{B}} - \ddot{\boldsymbol{v}}_{\boldsymbol{B}} \mid^2. \tag{25}$$

The estimated symbol misalignment is given by $\widehat{\tau} = \widehat{t}_A - \widehat{t}_B$.

D. Numerical results

This subsection evaluates the performance of our double baud-rate estimator benchmarked against a baud-rate estimator. Let $\widehat{\tau}^b$ and $\widehat{\tau}^d$ be the estimated symbol misalignments using baud-rate and double baud-rate estimators. Their respective square errors are

$$\varepsilon^b = (\widehat{\tau}^b - \tau)^2, \tag{26}$$

$$\varepsilon^d = (\widehat{\tau}^d - \tau)^2, \tag{27}$$

where τ is the real symbol misalignment. In particular, we are interested in the the mean-square-error (MSE) and the probability density function (PDF) of the square error.
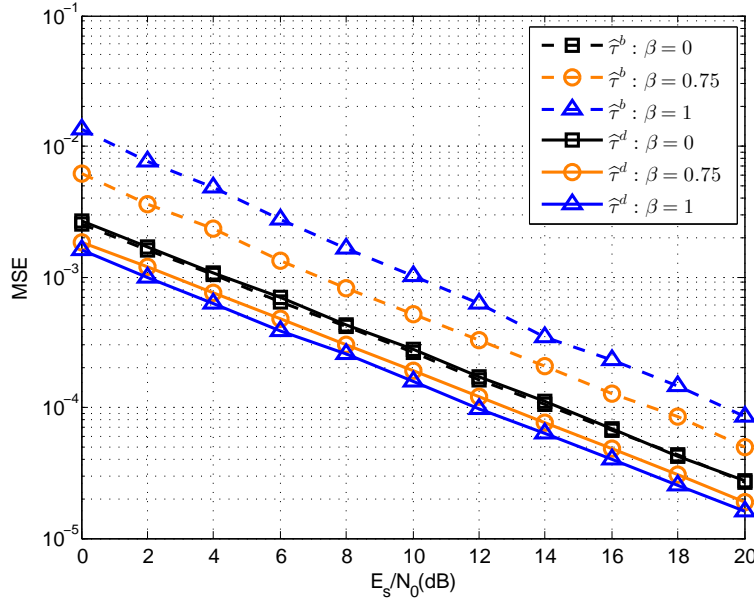

Fig. 5. The MSEs of $\widehat{\tau}^b$ and $\widehat{\tau}^d$ under different β and fixed Q. We set $Q = 31$, $\beta = 0$, 0.75 or 1, $G = 10$, $d = 4$, $T = 1$, $h_A = h_B = 1$, τ is uniformly distributed in $[-0.5, 0.5)$. The dashed curves mark the MSE of $\widehat{\tau}^b$ while the solid curves mark the MSE of $\widehat{\tau}^d$.

1) MSE Performance: First, we set $h_A = h_B = 1$ (i.e., AWGN channel), and evaluate the MSE of $\hat{\tau}^d$ given a fixed ZC-sequence length $Q = 31$ and different roll-off factors $\beta = 0,\ 0.75$ or 1.

The simulation results are shown in Fig.5. When $\beta = 0$ (the case of SINC pulse), the two curves that mark the MSEs of $\hat{\tau}^b$ and $\hat{\tau}^d$ coincide with each other.[3] This outcome is expected since baud-rate is already the Nyquist rate for SINC pulse. However, with the increase of β, the performance of $\hat{\tau}^b$ degrades since the baud-rate is an undersampling rate when $0 < \beta \leq 1$; whereas for $\hat{\tau}^d$, the performance improves with the increase of β. Overall, the performance of $\hat{\tau}^d$ is superior to that of $\hat{\tau}^b$. The performance gain of $\hat{\tau}^d$ over $\hat{\tau}^b$ increases with the increase of β. When $\beta = 1$, the performance gap is about 8 dB.
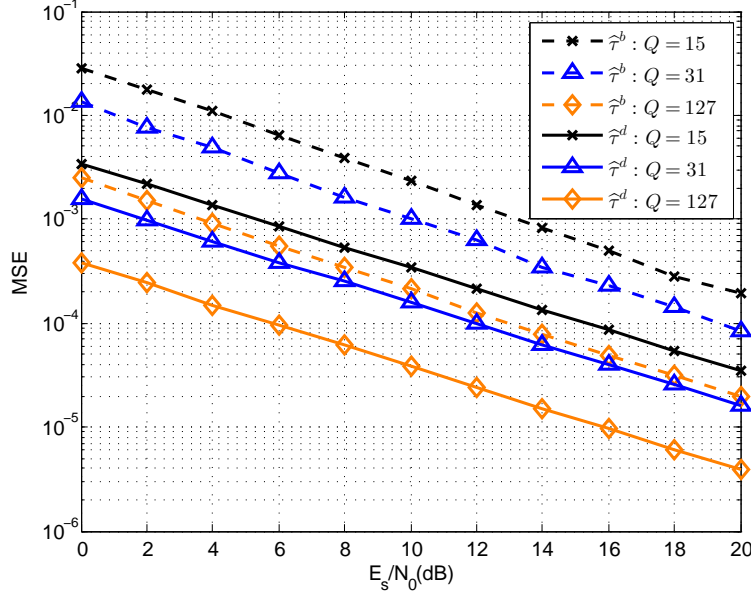

Fig. 6. The MSEs of $\hat{\tau}^b$ and $\hat{\tau}^d$ under different Q and fixed β. We set $\beta = 1$, $Q = 15,\ 31$ or 127, $G = \lfloor Q/3 \rfloor$, $d = 4$, $T = 1$, $h_A = h_B = 1$, τ is uniformly distributed in $[-0.5, 0.5)$. The dashed curves mark the MSE of $\hat{\tau}^b$ while the solid curves mark the MSE of $\hat{\tau}^d$.

Then, given a fixed roll-off factor $\beta = 1$ and different ZC-sequence length $Q = 15,\ 31$ or 127, we evaluate the MSEs of $\hat{\tau}^b$ and $\hat{\tau}^d$ in the AWGN channel. The simulation results are shown in Fig.6. As can be seen, the performance of both estimators improves with the increase of Q, and the double baud-rate estimator consistently outperforms the baud-rate estimator.

[3]In fact, from Fig.5, the MSE of $\hat{\tau}^b$ is slightly better than that of $\hat{\tau}^d$ when $\beta = 0$. The reason is that the whitening process is sensitive to the assumption $Q \gg j$. If we further increase the length of ZC sequence, e.g. we set Q=63, d=4, these two curves would coincide exactly.
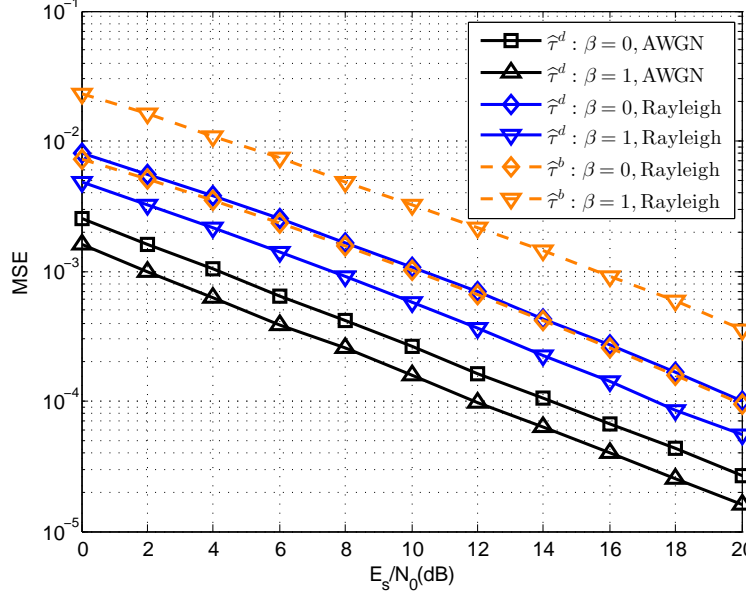
Fig. 7. The MSEs of $\widehat{\tau}^b$ and $\widehat{\tau}^d$ in a flat Rayleigh fading channel. we set $Q = 31$, $\beta = 0$ or 1, $G = 10$, $d = 4$, $T = 1$, the real relative time offset τ is assumed to be uniformly distributed in $[-0.5, 0.5)$; and the channel gains h_A and h_B are Rayleigh distributed.

Fig.7 presents the MSEs of $\widehat{\tau}^b$ and $\widehat{\tau}^d$ in a flat Rayleigh fading channel. After averaging over a large number of (h_A, h_B) pairs, it can be seen that, (i) the conclusions drawn in the AWGN channel (see Fig.5) still holds in the flat Rayleigh fading channel. Specifically, the gain of the double baud-rate estimator over the baud-rate estimator is also about 8 dB when $\beta = 1$ under Rayleigh fading. (ii) compared with AWGN channel, the Rayleigh fading channel leads to a 6 dB degradation of the MSEs for both $\widehat{\tau}^b$ and $\widehat{\tau}^d$. However, it is worth noting that MSE does not fully characterize the estimation performance, since it is only a first-order statistical characteristic of square error. Thus, we further analyze the PDF of the square error in the following.

2) PDF of Square Error: The PDFs of ε^d in the AWGN and flat Rayleigh fading channels are shown in Fig.8(a) and Fig.8(b), respectively, where we fix $Q = 31$, $\beta = 0$,[4] $E_b/N_0 = 10$ dB. Given these two figures, we have the following two key observations:

- In the AWGN channel, $\varepsilon^d \in [0,\ 0.004)$; while in the flat Rayleigh fading channel, $\varepsilon^d \in [0, 1)$. This is intuitive since the estimation may be totally off when Rayleigh channel is under deep fading.

---

[4]Note that $\beta = 0$ is the worst case for our double baud-rate estimator.
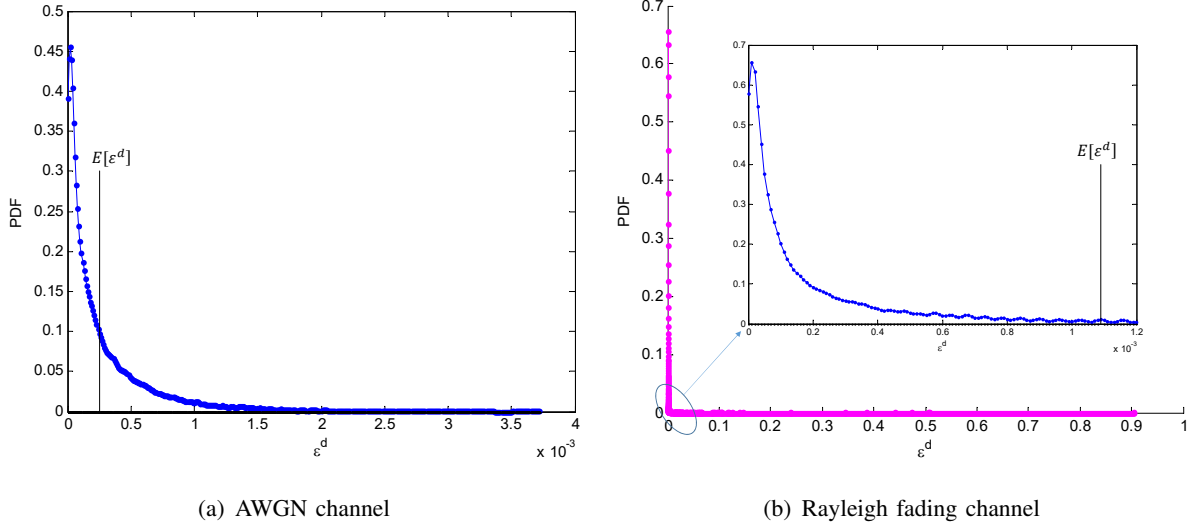
(a) AWGN channel   (b) Rayleigh fading channel

Fig. 8. The PDF of ε^d in the AWGN and flat Rayleigh fading channels. We set $Q = 31$, $\beta = 0$, $E_b/N_0 = 10$, $G = 10$, $d = 4$, $T = 1$, τ is uniformly distributed in $[-0.5, 0.5)$. In Fig.8(a), $h_A = h_B = 1$. In Fig.8(b), h_A and h_B are Rayleigh distributed.

- In both channels, ε^d falls into the range $[0,\ 0.001]$ with high probability, even though the MSEs differ greatly.

Recall that in Fig.7, the MSE performance of our double baud-rate estimator suffers a 6 dB degradation in Rayleigh channel compared with that in an AWGN channel. However, the two observations above imply that, the large square error under deep fading in the Rayleigh fading channel may pull down the average performance (i.e., the MSE performance). The real performance of our double baud-rate estimator in a Rayleigh fading channel is comparable to that in AWGN channel. If we define good/acceptable estimation, say, to be $\varepsilon^d \leq 0.001$, the probability that a good estimation occurs in AWGN channel is about 94.69%, while that in Rayleigh channel is about 90.39%.

IV. Optimal RRC-APNC Decoders

Given the estimated symbol misalignment in Section III, this section devises optimal RRC-APNC decoders assuming baud-rate and double baud-rate sampling on the data part. As stated in the system model, an optimal RRC-APNC decoder aims to find the most likely network-coded (XOR) symbol $s_{A \oplus B}[n]$ in accordance with the metric in (7).

For a conventional point-to-point system with RRC pulse, the optimal decoder is simply a symbol-by-symbol decoder since we could always sample at specific positions to eliminate the ISI [11]. However, for RRC-APNC, when the symbol boundaries of A and B are not aligned,

ISI is inevitable because there are no ISI-free sampling positions for both signals from nodes A and B.

Under ISI, an ideal optimal PNC decoder should decode $s_{A \oplus B}[n]$ based on all the samples within the packet since the information about each $s_{A \oplus B}[n]$ is spread across the whole packet given the long tails of the RRC pulse. This gives rise to a critical issue: the enormous state size of the decoder. Specifically, if the data length in a single packet is N, then each sample in (5) or (6) is related to $2N$ transmitted symbols from both nodes A and B, and each observation node[5] of an optimal decoder will have q^{2N} states (q is the modulation order). Given the state size grows exponentially with N, the ideal optimal PNC decoder is infeasible when complexity is taken into account. In this context, this paper takes the following two measures to enable feasible decoders:

(i) Sample at the symbol boundaries – As indicated in (5) or (6), the samples at the symbol boundaries of s_A or s_B (i.e. the samples at $nT + t_A$ or $nT + t_B$, $n = 1, 2, ..., N$) contain no information relevant to $s_A[n']$, $n' \neq n$ or $s_B[n]$, $n' \neq n$ since RC pulse meets the Nyquist ISI-free criterion. In other words, we could eliminate the ISI from one user if we sample at the symbol boundaries of this user.

(ii) Truncate states – Measure (i) can eliminate the ISI of one user only; the long-tailed ISI of the other user remains. Practically, given a fixed processing capacity of the receiver – e.g., the size of states that the receiver can handle is at most q^L – the only choice is to make an approximation by truncating the states.

Based on the two measures above, this paper constructs a factor graph, for which a sum-product algorithm (SPA) [14], [15] can be used to decode the network-coded symbols. This factor graph has a tree structure, and hence iterative decoding is not necessary inside the PNC decoder. For a channel-coded system, however, we will have an additional factor graph corresponding to the channel code. These two factor graphs are combined into one for an overall system and turbo decoding is also possible [16]. In the following, we focus on the design of the double baud-rate decoder. The design of the baud-rate decoder is elaborated in Appendix C.

---

[5]The terms "observation node", "variable node" and "check node" in this paper follow the conventional definitions in [14], [15]. In particular, an observation node corresponding to a single sampling point.

A. Discrete-time model

We have obtained the double baud-rate samples of $y_R(t)$ in (6). The estimated time offsets of signals from two end nodes have been given in Section III. Thus, we first follow the measure (i) above and derive the samples at the symbol boundaries of s_A and s_B.

Double baud-rate sampling is information-lossless when converting $y_R(t)$ to the discrete-time signal. Thus, the double baud-rate samples y^d can be used to reconstruct the continuous-time signal $y_R(t)$. That is,

$$\widehat{y}_R(t) = \sum_k y^d[k] Sinc(2t - kT), \tag{28}$$

As indicated by the Nyquist-Shannon sampling theorem [11], we have $\widehat{y}_R(t) = y_R(t)$.

Next, given the estimated time offsets $\widehat{t}_A$ and $\widehat{t}_B$, the samples at the symbol boundaries (i.e., $nT + \widehat{t}_A$ and $nT + \widehat{t}_B$) can be obtained by (the following computation can be interpreted as a "virtual resampling process" whereby the samples at the two boundaries are obtained from $\widehat{y}_R(t)$ in (28) through computation rather than actual resampling)

$$
\begin{aligned}
y_1[n] &= \widehat{y}_R(t = nT + \widehat{t}_A) \\
&= h_A \sum_l s_A[n-l] p(lT + \widehat{t}_A - t_A) + h_B \sum_l s_B[n-l] p(lT + \widehat{t}_A - t_B) + w_1[n]; \quad (29) \\
y_2[n] &= \widehat{y}_R(t = nT + \widehat{t}_B) \\
&= h_A \sum_l s_A[n-l] p(lT + \widehat{t}_B - t_A) + h_B \sum_l s_B[n-l] p(lT + \widehat{t}_B - t_B) + w_2[n], \quad (30)
\end{aligned}
$$

where w_1 and w_2 are i.i.d. Gaussian noise, $w_1[n], w_2[n] \sim CN(0, \sigma^2)$, but $[w_1, \ w_2]^T$ is colored because a component in w_1 may not be independent of a component in w_2. Notice that if the estimated boundaries are exact (i.e., $\widehat{t}_A = t_A$ and $\widehat{t}_B = t_B$), the samples $y_1[n]$ and $y_2[n]$ contain information only relevant to the variables $\{s_A[n], \ s_B[n'], \ n' = 1, 2, ..., N\}$ and $\{s_B[n], \ s_A[n'], \ n' = 1, 2, ..., N\}$, respectively.

Further, we can combine (29) and (30) and rewrite them in matrix form:

$$y = \Gamma H x + w, \tag{31}$$

where the vectors y, x, w and the matrix H are defined as

$$y = \Big[y_1[1], y_2[1], \ y_1[2], y_2[2], \ ...,..., \ y_1[N], y_2[N], \Big]^T, \tag{32}$$

$$x = \Big[s_A[1], s_B[1], \ s_A[2], s_B[2], \ ...,..., \ s_A[N], s_B[N], \Big]^T, \tag{33}$$

$$\boldsymbol{w} = \left[w_1[1], w_2[1], \ w_1[2], w_2[2], \ ...,..., \ w_1[N], w_2[N], \right]^T, \tag{34}$$

$$\boldsymbol{H} = \text{diag}(h_A, h_B, h_A, h_B, ..., ..., h_A, h_B). \tag{35}$$

The coefficient matrix $\boldsymbol{\Gamma}$ is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} p(\widehat{t}_A - t_A) & p(\widehat{t}_A - t_B) & p(\widehat{t}_A - t_A - T) & p(\widehat{t}_A - t_B - T) & p(\widehat{t}_A - t_A - 2T) & p(\widehat{t}_A - t_B - 2T) & \cdots \\ p(\widehat{t}_B - t_A) & p(\widehat{t}_B - t_B) & p(\widehat{t}_B - t_A - T) & p(\widehat{t}_B - t_B - T) & p(\widehat{t}_B - t_A - 2T) & p(\widehat{t}_B - t_B - 2T) & \cdots \\ p(\widehat{t}_A - t_A + T) & p(\widehat{t}_A - t_B + T) & p(\widehat{t}_A - t_A) & p(\widehat{t}_A - t_B) & p(\widehat{t}_A - t_A - T) & p(\widehat{t}_A - t_B - T) & \cdots \\ p(\widehat{t}_B - t_A + T) & p(\widehat{t}_B - t_B + T) & p(\widehat{t}_B - t_A) & p(\widehat{t}_B - t_B) & p(\widehat{t}_B - t_A - T) & p(\widehat{t}_B - t_B - T) & \cdots \\ p(\widehat{t}_A - t_A + 2T) & p(\widehat{t}_A - t_B + 2T) & p(\widehat{t}_A - t_A + T) & p(\widehat{t}_A - t_B + T) & p(\widehat{t}_A - t_A) & p(\widehat{t}_A - t_B) & \cdots \\ p(\widehat{t}_B - t_A + 2T) & p(\widehat{t}_B - t_B + 2T) & p(\widehat{t}_B - t_A + T) & p(\widehat{t}_B - t_B + T) & p(\widehat{t}_B - t_A) & p(\widehat{t}_B - t_B) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{36}$$

Moreover, we specify the following observations from (31),

(i) The coefficient matrix $\boldsymbol{\Gamma}$ is related to the real time offsets t_A and t_B as well as the estimated time offsets $\widehat{t}_A$ and $\widehat{t}_B$. Thus, $\boldsymbol{\Gamma}$ is unknown to the receiver since t_A and t_B are unknown.

(ii) The noise $\boldsymbol{w}$ is colored, its covariance matrix is given by

$$\boldsymbol{\Sigma}(\boldsymbol{w}) = \sigma^2 \boldsymbol{\Sigma} = \sigma^2 \begin{bmatrix} 1 & p(\widehat{\tau}) & 0 & p(\widehat{\tau} - T) & \cdots & p(\widehat{\tau} - (N-1)T) \\ p(-\widehat{\tau}) & 1 & p(-\widehat{\tau} - T) & 0 & \cdots & 0 \\ 0 & p(\widehat{\tau} + T) & 1 & p(\widehat{\tau}) & \cdots & p(\widehat{\tau} - (N-2)T) \\ p(-\widehat{\tau} + T) & 0 & p(-\widehat{\tau}) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p(-\widehat{\tau} + (N-1)T) & 0 & p(-\widehat{\tau} + (N-2)T) & 0 & \cdots & 1 \end{bmatrix}, \tag{37}$$

where $\widehat{\tau} = \widehat{t}_A - \widehat{t}_B$. Note that $\boldsymbol{\Sigma}(\boldsymbol{w})$ is only related to the estimated time offsets $\widehat{t}_A$ and $\widehat{t}_B$, and hence is known to the receiver.

(iii) When the estimation is exact, i.e., $\widehat{t}_A = t_A$ and $\widehat{t}_B = t_B$, we have $\boldsymbol{\Gamma} = \boldsymbol{\Sigma}$.

B. Equivalent whitened model

The discrete-time model in (31) is essentially a time-varying ISI channel model, where the coefficients in the matrix $\boldsymbol{\Gamma}$ capture the severities of ISI: the more bib-zero off-diagonal elements the $\boldsymbol{\Gamma}$ has, the more severe is the ISI. On this basis, the conventional optimal decoders designed for ISI channel applies to (31) naturally.

However, we notice that the noise $\boldsymbol{w}$ is colored. For the optimal decoder designed for ISI channel, colored noise generally complicates the optimal metrics and the decoding structure. Let us take the maximum likelihood sequence estimator (MLSE) [17] for instance. MLSE is a well-known optimal decoder for ISI channel subject to simple implementation by a Viterbi

algorithm operating on a trellis. Given an input sequence with white noise, the metric in each transition reduces to a minimum Euclidean distance metric, and the decoding can be operated recursively since successive transitions are independent [17], [18]. Colored noise, on the other hand, would destroy this simple algorithm.

To retain the simplicity, we whiten the noise w before the samples are fed into optimal decoders. Appendix B proves that the matrix Σ in (37) is symmetric and positive definite. This suggests that the noise can be whitened using Cholesky decomposition as that in (22).

First, we factorize $\Sigma : \Sigma = U^T U$, where U is a full-ranked upper triangular matrix. Then, we left multiply (31) by U^{-T}, giving,

$$
\begin{aligned}
U^{-T} y &= U^{-T} \Gamma H x + U^{-T} w \\
&\overset{(a)}{=} U^{-T}(\Sigma + \Delta) H x + U^{-T} w \qquad \overset{\text{def}}{=} \qquad \bar{y} = U H x + \epsilon + \bar{w} \qquad (38) \\
&= U H x + U^{-T} \Delta H x + U^{-T} w,
\end{aligned}
$$

The structure of (38) is explained and detailed below:

(i) The transformed noise $\bar{w}$ is now i.i.d. Gaussian noise with zero mean and covariance matrix $\sigma^2 I$ since $\mathbb{E}[\bar{w}\bar{w}^H] = \mathbb{E}[U^{-T} w w^H U^{-1}] = \sigma^2 U^{-T} \Sigma U^{-1} = \sigma^2 I$. Thus, (38) is the whitened discrete-time model corresponding to (31).

(ii) In step (a), the matrix Γ is broken into the sum of two parts: Σ and Δ. Notice that if the estimation is exact (i.e. $\widehat{t}_A = t_A$ and $\widehat{t}_B = t_B$), we have $\Gamma = \Sigma$. In other words, Δ is an error matrix caused by inaccurate estimation.

(iii) Recall that the matrix Γ captures the severities of ISI. Compare matrix U with Γ, it is interesting to note that the whitening transformation not only whitens the noise but also reduces complexity since the matrix U is an upper triangular matrix.

In (38), we further define an error vector $\epsilon = U^{-T} \Delta H x$. This error vector ϵ, caused by inaccurate estimation, is unknown to the receiver since the error matrix Δ is unknown. Thus, the receiver will ignore ϵ and assume

$$
\bar{y} \approx U H x + \bar{w}. \qquad (39)
$$

This approximation is valid only when the estimation is accurate enough, or else, the decoding performance will suffer. As will be shown later, the estimation accuracy of our double baud-rate estimator enables a small ϵ even with short preamble.

C. Optimal decoder with truncated states

An optimal PNC decoder aims to decode the most likely network-coded symbols in the presence of ISI. In general, two kinds of optimal decoders are possible for RRC-APNC [3]:

(i) The decoder that maximizes the joint APP of the sequence $s_{AB} = [s_A, s_B]^T$, i.e. maximizing $\Pr(s_A, s_B \mid \bar{y})$. Specifically, the decoder first finds the most likely sequence $\widehat{s_{AB}}$ given by (40), and then performs XOR operation to decode the network-coded sequence $\widehat{s}_{A \oplus B}$.

$$\widehat{s_{AB}} = \arg \max_{s_A, s_B} \Pr(x \mid \bar{y}) \tag{40}$$

(ii) The decoder that maximizes the marginal APP of each pair $\{s_A[n], s_B[n]\}$, i.e., maximizing $\Pr\{s_A[n], s_B[n] \mid \bar{y}\}, n = 1, 2, ..., N$. Specifically, the decoder first decodes the maximum APPs: $\Pr\{s_A[n], s_B[n] \mid \bar{y}\}, n = 1, 2, ..., N$, and then find the optimal $\widehat{s}_{A \oplus B}[n]$ given by

$$\widehat{s}_{A \oplus B}[n] = \arg \max_{s_{A \oplus B}[n]} \sum_{\substack{s_A[n], s_B[n]: \\ s_A[n] \oplus s_B[n] = s_{A \oplus B}[n]}} \Pr\left\{s_A[n], s_B[n] \mid \bar{y}\right\} \tag{41}$$

Decoders of type (i) can be realized by MLSE [10], while decoders of type (ii) can be realized using BCJR algorithm [9] or SPA. For channel-coded PNC in this paper, we prefer decoders of type (ii) since it provides soft information that can be fed to the channel decoder. We further remark that BCJR algorithm and SPA are similar [14]; BCJR is essentially a special application of SPA on trellis. Thus, this paper makes no distinctions between these two algorithms and focus on the SPA decoder.

The graphical interpretation of (39) is given in Fig.9, where the error vector ϵ is ignored, and the i-th entries in x and $\bar{y}$ form the variable node X_i and observation node $\overline{Y}_i$, respectively. In particular, we truncate the state size in each observation node to q^L (i.e., the measure (ii) in the beginning of this section to enable feasible decoder), and hence each observation node only connects L variable nodes. In other words, the decoder only considers the $L - 1$ off-diagonals adjacent to the main diagonal beside the main diagonal in matrix U, and the other off-diagonals are truncated. As a result, each variable node X_i has q states and each observation node $\overline{Y}_i$ has q^L states.

The factor graph in Fig.9 is cyclic. If we directly run SPA on this graph, iterative decoding is required and the resulting APP is not exact [14]. However, we could first convert it into an acyclic graph. A possible transformation is to cluster all the variable nodes that connect to the same
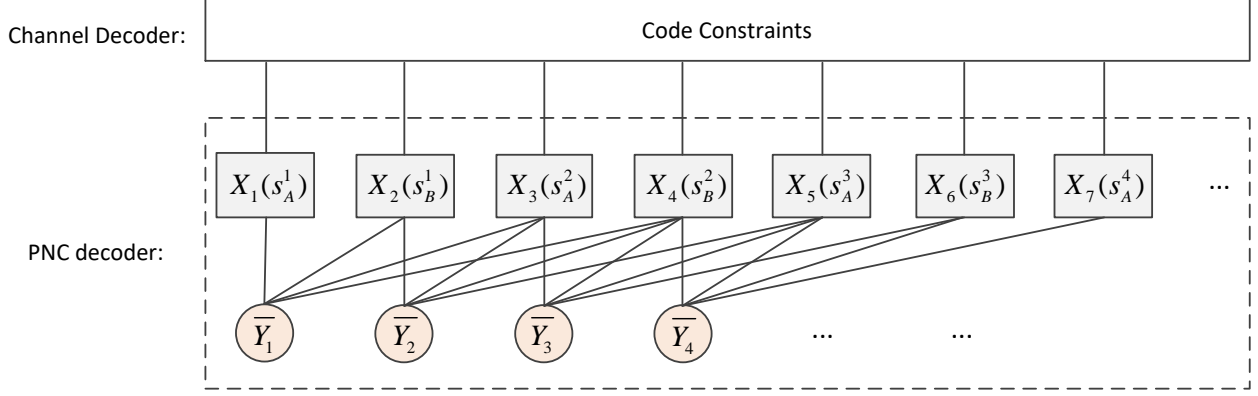
Fig. 9. The graphical interpretation of (39), X_i denotes the i-th variable node, $\overline{Y}_i$ denotes the i-th observation node. Each variable node X_i represents a transmitted symbol $s_A[n]$ or $s_B[n]$ (abbreviate as s_A^n or s_B^n). The state size in $\overline{Y}_i$ is truncated so that each $\overline{Y}_i$ is connected with L X_i ($L = 4$ in this figure).

observation node. Specifically, given Fig.9, we cluster $\{X_i, X_{i+1}, X_{i+2}, X_{i+3}\}$ and construct a new variable node $V_i = \{X_i, X_{i+1}, X_{i+2}, X_{i+3}\}$. The transformed factor graph after clustering is given in Fig.10, and the following variations can be observed:

(i) The link between V_i and $\overline{Y}_i$ is now singly-connected, and each new variable node V_i has q^L states.

(ii) Constraints among V_i emerge, because common symbols are contained in adjacent V_i. These constraints are expressed through the check nodes C_i in Fig.10,[6] whose corresponding check function ψ_i is given by

$$\psi_i = \begin{cases} 1 & \text{if the values of all the common symbols between } V_i \text{ and } V_{i+1} \text{ are equal} \\ 0 & \text{if at least one common symbol between } V_i \text{ and } V_{i+1} \text{ differs in value} \end{cases} \tag{42}$$

For instance, the common symbols between V_1 and V_2 are $\{s_B^1, s_A^2, s_B^2\}$. Thus, $\psi_1 = 1$ for $V_1 = \{b_1, 1, 1, 1\}, V_2 = \{1, 1, 1, b_2\}$, and $\psi_1 = 0$ for $V_1 = \{b_1, 1, 1, 1\}, V_2 = \{-1, 1, 1, b_2\}$, where b_1 and b_2 can be either 1 or -1.

(iii) This graph has a tree structure. This implies that we could start the computation from the leaf nodes (for this graph, the two boundary nodes on the left and right). Each message

---

[6]Note that the check on the adjacent two variable nodes is enough to depict all the constraints. Adding more check node would only result in redundancy and unnecessary cycles, hence degrade the performance. For instance, there is intersection between V_1 and V_3 but we do not need to check these two variable nodes since the constrains are embedded in check nodes C_1 and C_2 already.

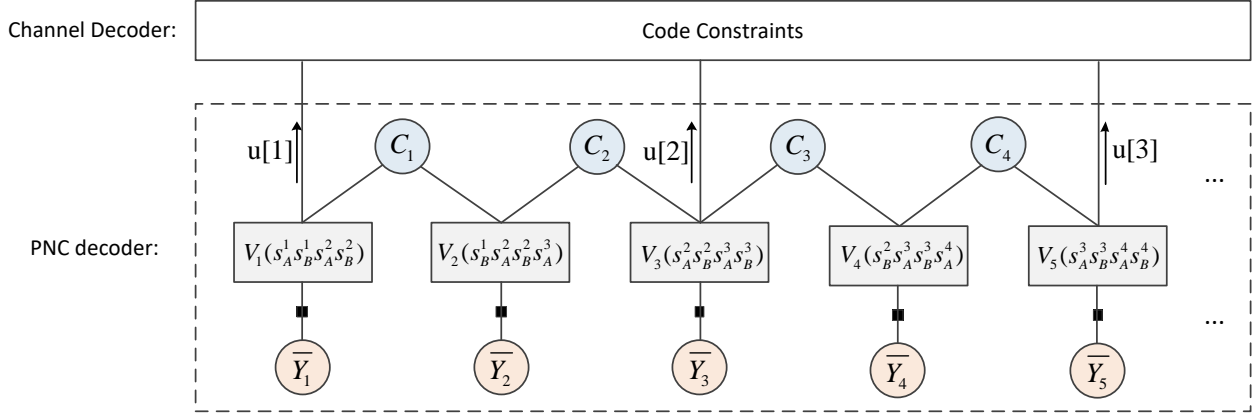needs to be computed once and only once, after which the exact APP can be obtained [3], [14].

Fig. 10. The equivalent acyclic factor graph corresponding to Fig.9, where V_i denotes the i-th new variable node, $V_i = \{X_i, X_{i+1}, X_{i+2}, X_{i+3}\}$ ($L = 4$); C_i is the i-th check node; $\overline{Y_i}$ is the i-th observation node.

Given the transformed acyclic graph in Fig.10, we can then run SPA to maximize the APPs $\Pr\{s_A[n], s_B[n] \mid \bar{\boldsymbol{y}}\}, n = 1, 2, ..., N$. The processing of SPA contains a forward (left-to-right) message passing and a backward (right-to-left) message passing. For the forward message passing, at each variable node, we take the product over all the incoming messages to form the outgoing message; at each check node, we sum up all the messages that link to the same state in the next variable node.

To simplify notations, we denote the L variates in V_i by $\{\omega_i^1, \omega_i^2, ..., \omega_i^L\}$, where each variate can be either 1 or -1. Thus, each V_i has 2^L states, and we define V_i to be in state $(b_1, b_2, ..., b_L)$ if $\omega_i^1 = b_1$, $\omega_i^2 = b_2$, ..., $\omega_i^L = b_L$, where $b_1, b_2, ..., b_L \in \{1, -1\}$. For instance in Fig.10, for V_1, we have $\omega_1^1 = s_A^1$, $\omega_1^2 = s_B^1$, $\omega_1^3 = s_A^2$, $\omega_1^4 = s_B^2$. Thus, V_i is in state (1,1,1,1) when $\omega_1^1 = \omega_1^2 = \omega_1^3 = \omega_1^4 = 1$.

Initialization. First, we calculate the posterior probability of each state in V_i based on the corresponding observation node $\overline{Y_i}$. That is, the probability that V_i is in state $(b_1, b_2, ..., b_L)$ is given by

$$
\begin{aligned}
p_i(b_1, b_2, ..., b_L) &= \Pr(\omega_i^1 = b_1, \omega_i^2 = b_2, ..., \omega_i^L = b_L \mid \bar{\boldsymbol{y}}) \\
&= \kappa_i \exp\left\{ -\frac{1}{2\sigma^2} \left| \bar{y}[i] - \sum_{l=0}^{L-1} \boldsymbol{U}(i, i+l)\boldsymbol{H}(i+l, i+l)b_{l+1} \right|^2 \right\}
\end{aligned} \tag{43}
$$

where κ_i is the normalization factor; $\boldsymbol{U}(i,j)$ or $\boldsymbol{H}(i,j)$ is the (i,j)-th entry of matrix $\boldsymbol{U}$ or $\boldsymbol{H}$ in (39).

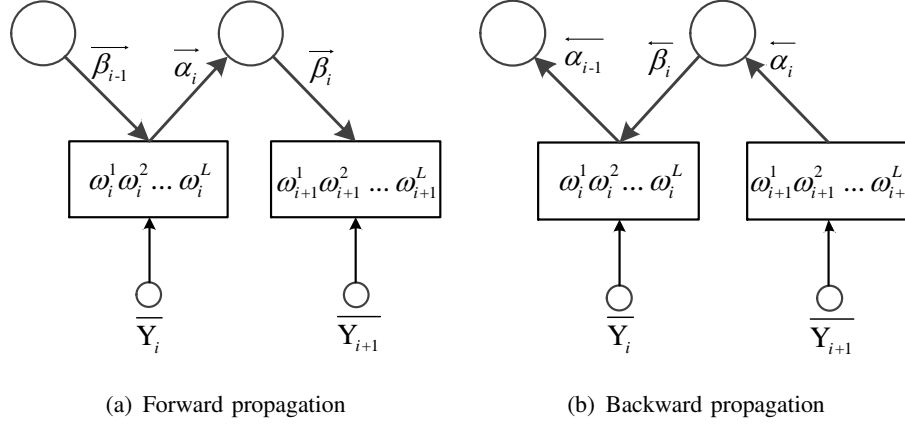(a) Forward propagation   (b) Backward propagation

Fig. 11. The forward and backward propagations in SPA.

Forward propagation. The "sum" and "product" operations in the forward propagation are shown in Fig.11(a), where the messages relevant to state $(b_1, b_2, ..., b_L)$ output from the i-th variable node and check node are defined as $\overrightarrow{\alpha_i}(b_1, b_2, ..., b_L)$ and $\overrightarrow{\beta_i}(b_1, b_2, ..., b_L)$, respectively.

Specifically, we have

$$\overrightarrow{\alpha_i}(b_1, b_2, ..., b_L) = \overrightarrow{\beta_i}(b_1, b_2, ..., b_L)p_i(b_1, b_2, ..., b_L); \tag{44}$$

$$\overrightarrow{\beta_i}(b_1, b_2, ..., b_{L-1}, \pm 1) = \kappa' \sum_{b_1} \overrightarrow{\alpha_i}(b_1, b_2, ..., b_L), \tag{45}$$

where κ' is a normalization factor, and $p_i(b_1, b_2, ..., b_L)$ is given in (43).

Backward propagation. The "sum" and "product" operations in the backward propagation are shown in Fig.11(b), where the messages relevant to state $(b(b_1, b_2, ..., b_L)$ output from the i-th variable node and check node are defined as $\overleftarrow{\alpha_i}(b_1, b_2, ..., b_L)$ and $\overleftarrow{\beta_i}(b_1, b_2, ..., b_L)$, respectively.

Specifically, we have

$$\overleftarrow{\beta_i}(\pm 1, b_2, ..., b_L) = \kappa'' \sum_{b_L} \overleftarrow{\alpha_i}(b_1, b_2, ..., b_L); \tag{46}$$

$$\overleftarrow{\alpha_{i-1}}(b_1, b_2, ..., b_L) = \overleftarrow{\beta_i}(b_1, b_2, ..., b_L)p_i(b_1, b_2, ..., b_L), \tag{47}$$

where κ'' is a normalization factor, and $p_i(b_1, b_2, ..., b_L)$ is given in (43).

Marginalization. After one forward propagation from left to right and one backward propagation from right to left, the APPs of each state in variable nodes now converge, giving

$$\ddot{p}_i(b_1, b_2, ..., b_L) = \overrightarrow{\beta_{i-1}}(b_1, b_2, ..., b_L)\overleftarrow{\beta_i}(b_1, b_2, ..., b_L)p_i(b_1, b_2, ..., b_L), \quad i = 1, 2, ..., 2N. \tag{48}$$

Further, extracting the odd entries of (48), we have

$$\Pr\{s_A[n] = b_1, s_B[n] = b_2 \mid \bar{\boldsymbol{y}}\} = \sum_{b_3, b_4} \ddot{p}_{2n+1}(b_1, b_2, ..., b_L), \quad n = 1, 2, ..., N. \tag{49}$$

Certainly, SPA can also operate in log-domain to avoid the numerical issues (i.e., the log-SPA [19]).

Decoding. For a non-channel-coded system, we directly decode the most likely network-coded symbol $s_{A \oplus B}[n]$ from (7). For a channel-coded system, we could perform XOR-CD decoding or Jt-CNC decoding [3], and turbo decoding is also possible under the framework [16]. However, comparing the performance of different decoders is beyond the scope of this paper. For simplicity, we perform XOR-CD decoding based on (49). Specifically, instead of decoding $s_{A \oplus B}[n]$ directly, we calculate the soft information as

$$\begin{aligned} u[n] &= \frac{\Pr\{s_{A \oplus B}[n] = 0\}}{\Pr\{s_{A \oplus B}[n] = 1\}} \\ &= \frac{\sum_{b_1 \oplus b_2 = 0} \Pr\{s_A[n] = b_1, s_B[n] = b_2 \mid \bar{\boldsymbol{y}}\}}{\sum_{b_1 \oplus b_2 = 1} \Pr\{s_A[n] = b_1, s_B[n] = b_2 \mid \bar{\boldsymbol{y}}\}}. \end{aligned} \tag{50}$$

The soft information is then passed to the channel decoder, and we decode the network-coded symbols after several iterations inside the channel decoder.

D. Numerical results

Given the symbol misalignment estimators presented in Section III and the optimal PNC decoders presented in Section IV, there are four possible estimation-and-decoding solutions for RRC-APNC as shown in Table. I. Specifically, solution I corresponds to a baud-rate sampling

TABLE I

FOUR POSSIBLE ESTIMATION-AND-DECODING SOLUTIONS FOR RRC-APNC.

	Baud-rate estimation	Double baud-rate estimation
Baud-rate decoding	Solution I	Solution II
Double baud-rate decoding	Solution III	Solution IV

system, where estimation and decoding are based on the baud-rate samples. Solution IV corresponds to a double baud-rate sampling system, where estimation and decoding are based on double baud-rate samples. Solution II and III are the simplification of solution IV, where we first sample the received signal at double baud-rate but drop half of the samples corresponding to the preamble part (solution III) or data part (solution II).

This subsection studies these four estimation-and-decoding solutions to evaluates the overall performance of RRC-APNC when incorporating symbol misalignment estimation. We assume the transmitted symbols s_A and s_B are BPSK modulated,[7] and both nodes A and B utilize the same $(1024, 512)$ LDPC code [20], [21]. The relative time offset τ (i.e., symbol misalignment) is uniformly distributed in $[-T/2, T/2)$.

1) Truncation issue: As indicated in Section IV, truncation in state size is inevitable in the design of practical RRC-APNC decoders. For each observation node (in Fig.9 or Fig.15), we truncate the state size from 2^{2N} to 2^L. The choice of L is a tradeoff between performance and complexity. That is, given a fixed roll-off factor of the RRC pulse β, the greater the L, the less the information loss, but the more complicated the decoder. Thus, the first thing we need to figure out is to find the proper L under various β.
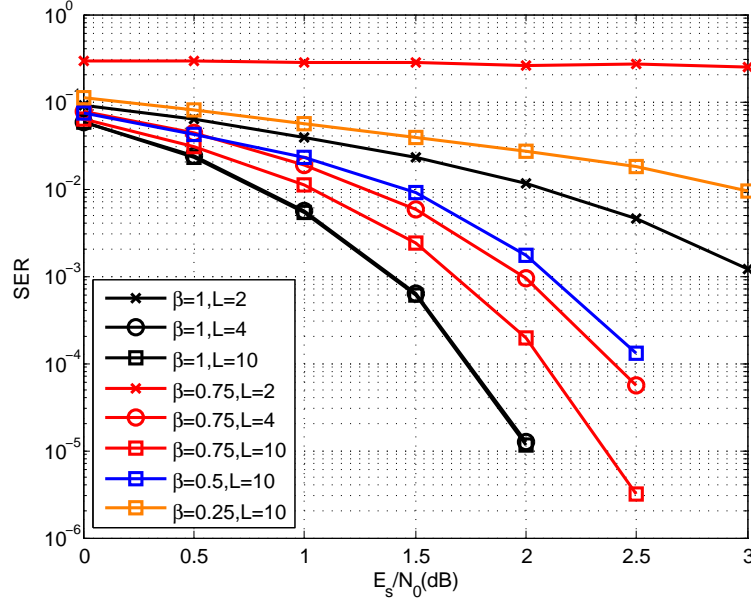

Fig. 12. The SER of RRC-APNC with exact time offset estimation and double baud-rate decoding. We set $h_A = h_B = 1$, $T = 1$, the relative time offset τ is uniformly distributed in $[-0.5, 0.5)$.

Assuming exact symbol-misalignment estimation, we investigate the effect of truncation (i.e., various L) on the symbol error rate (SER) of double baud-rate decoder.[8] The simulation results

---

[7]The simulations focus on the effect of symbol-misalignment estimation on the overall decoding performance, hence BPSK modulation is suffice. Besides, BPSK enables simpler SPA decoder (i.e., the modulation order of BPSK is $q = 2$, and hence the state size in each observation node is 2^L).

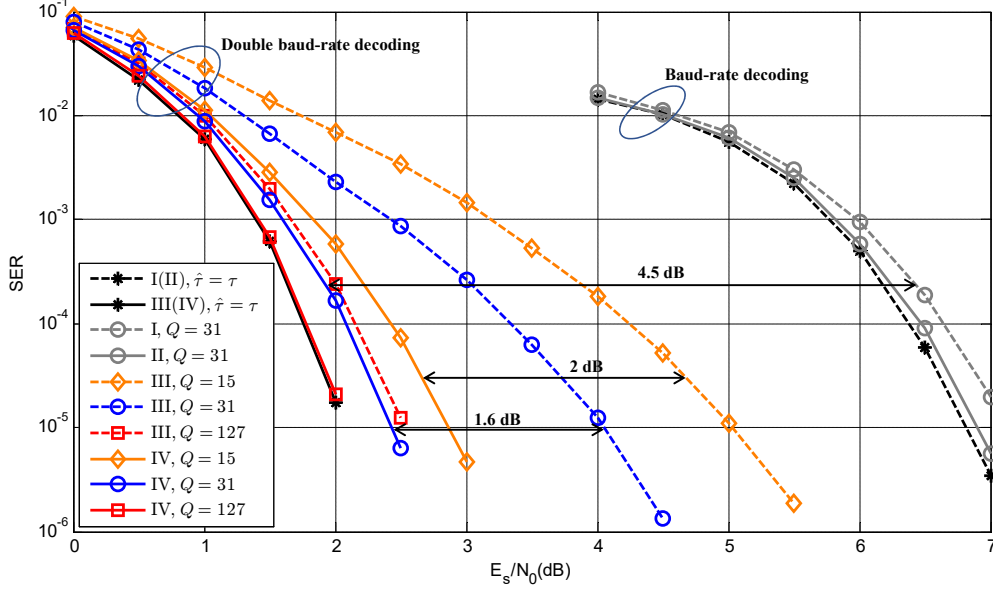[8]We specify that the conclusions drawn in this simulation apply to the baud-rate decoder as well.

Fig. 13. The SERs of the four estimation-and-decoding solutions in Table.I. For estimators, we set $\beta = 1$, $Q = 15, 31$ or 127, $G = \lfloor Q/3 \rfloor$, $d = 4$; for decoders, we set $L = 4$. Besides, $T = 1$, $h_A = h_B = 1$, the relative time offset τ is uniformly distributed in $[-0.5, 0.5)$. The dashed curves mark the SERs of solutions using baud-rate estimator (i.e., solutions I and III) while the solid curves mark the SERs of solutions using double baud-rate estimator (i.e., solutions II and IV).

are shown in Fig.12. Intuitively, a smaller β demands a greater L to guarantee the decoding performance. Thus, we start from $\beta = 1$ and increase L from 2 to 10. As can be seen, the SER improves with the increase of L when $L \leq 4$, whereas when $L > 4$, no distinct improvement can be observed. This implies that $L = 4$ is a good choice when $\beta = 1$. Then, fixing $\beta = 0.75$, we increase L from 2 to 10. The SER improves continuously with the increase of L. Further, if we increase L beyond 10, the SER no longer improves. Thus, we could set $L = 10$ when $\beta = 0.75$.

However, given $L = 10$ and BPSK modulation, the state size in each observation node is 1024. Such an enormous state size has already placed a heavy burden on the receiver, and hence further increasing L may not be wise. Thus, for practical RRC-APNC, a relative large roll-off factor β is preferred.

2) Estimation-and-decoding: Based on the observations in Fig.12, we set $\beta = 1$ and $L = 4$ in this simulation and evaluate the performance of four estimation-and-decoding solutions for RRC-APNC in Table. I. The simulation results are shown in Fig.13.

The first observation from Fig.13 is that, there is a gap between the SERs of solution I(II)
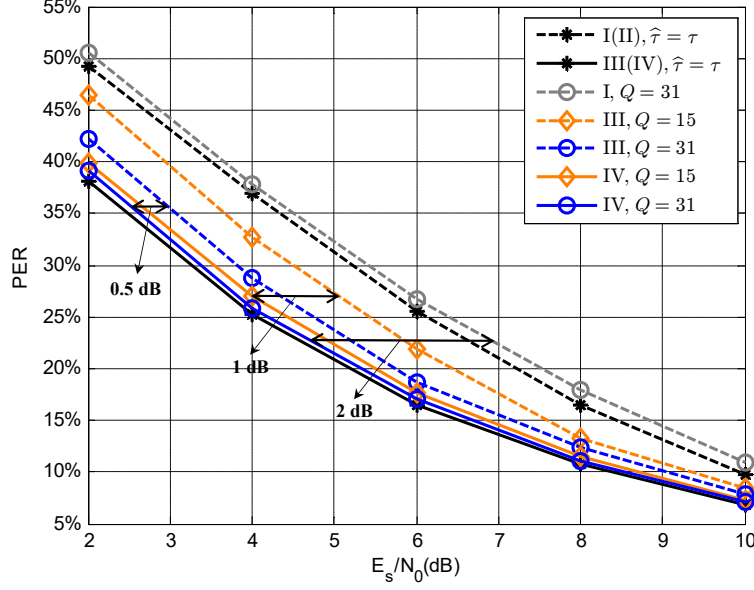
Fig. 14. The PERs of the four estimation-and-decoding solutions in Table.I. For estimators, we set $\beta = 1$, $Q = 15$ or 31, $G = \lfloor Q/3 \rfloor$, $d = 4$; for decoders, we set $L = 4$. Besides, $T = 1$, the relative time offset τ is uniformly distributed in $[-0.5, 0.5)$ and the channel gains h_A and h_B are Rayleigh distributed. The dashed curves mark the SERs of solutions using baud-rate estimator (i.e., solutions I and III) while the solid curves mark the SERs of solutions using double baud-rate estimator (i.e., solutions II and IV).

(with baud-rate decoder) and solution III(IV) (with double baud-rate decoder). When $Q = 31$, a double baud-rate sampling system (i.e., solution IV, $Q = 31$) yields 4.5 dB SER gains over a baud-rate sampling system (i.e., solution I, $Q = 31$). Even with exact symbol-misalignment estimation (i.e., $\hat{\tau} = \tau$), the 4.5 dB gap between solutions I and IV retains (the two dark curves in Fig.13). This implies that, double baud-rate decoding is indispensable for RRC-APNC, especially when a large roll-off factor is used.

Then, we study the impact of different estimators on the decoding performance assuming the double baud-rate decoder is used in both cases (i.e., solutions III and IV). Benchmarked against the exact misalignment-estimation case (solution III(IV), $\hat{\tau} = \tau$), solution III suffers a 2.8 dB SER loss when $Q = 15$, and a 2 dB loss when $Q = 31$; whereas for solution IV, the SER loss is 0.8 dB and 0.4 dB when $Q = 15$ and 31, respectively. Further, when $Q = 127$, solution IV almost achieves the benchmark, while solution III still suffers a 0.5 dB loss.

Overall, solution IV yields considerable SER improvements over solition III under the same preamble length (e.g., 2 dB when $Q = 15$ and 1.6 dB when $Q = 31$). For the same SER performance, solution IV only requires a quarter of the overhead (i.e., preamble length) that

solution III needs.

Fig.14 shows the decoding performances of various estimation-and-decoding solutions in a flat Rayleigh fading channel. As with Fig.8, SER performance does not fully characterize the decoding performance in a Rayleigh fading channel, because the large SER of packets under deep fading may pull down the overall average performance. Thus, we analyze the packet error rate (PER) instead.

As can be seen, a double baud-rate sampling system (i.e., solution IV) yields 2 dB PER gains over a baud-rate sampling system (i.e., solution I) when $Q = 31$. Solution IV yields 1 dB PER gains over solution III when $Q = 15$, and 0.5 dB PER gains over solution III when $Q = 31$. Basically, in terms of PER performance, solution IV achieves the benchmark when $Q = 31$ in a flat Rayleigh fading channel.

V. CONCLUSION

This paper investigated the joint problem of symbol-misalignment estimation and decoding in asynchronous physical-layer network coding systems using the root-raised-cosine pulse (RRC-APNC). For symbol-misalignment estimation, we put forth an optimal double baud-rate estimator that is substantially more accurate than prior schemes under both AWGN and Rayleigh fading channels. For example, the mean-square-error (MSE) gains of the double baud-rate estimator over the baud-rate estimator can be up to 8 dB. For PNC decoding, we devised optimal APNC decoders under inaccurate symbol-misalignment estimates. In particular, we proved and presented a new whitening transformation to whiten the noise of the double baud-rate samples. We investigated the decoding performance of various estimation-and-decoding solutions for RRC-APNC. Our results indicate that, in order to reduce decoder complexity while maintaining good performance, a large roll-off factor of the RRC pulse is preferable; and that with a large roll-off factor, double baud-rate decoding is indispensable. Overall, the scheme that makes use of double baud-rate samples in both symbol-misalignment estimation and PNC decoding has much better performance than other solutions for RRC-APNC.

APPENDIX A

DERIVATIONS ON THE OUTCOME OF CROSS-CORRELATION

A. Cross-correlation results

Given the double baud-rate samples $\bar{y}^d$ and z_A^d in (15) and (16), we derive the cross-correlation results, i.e., c_A^d in (18), in this subsection.

First, we denote by $\bar{y}_e, \bar{y}_o$ and z_e, z_o the even and odd entries of $\bar{y}^d$ and z_A^d, respectively. Thus, we have

$$\bar{y}_e[n] = h_A \sum_l s_A[n-l]p'(lT-t_A) + h_B \sum_l s_B[n-l]p'(lT-t_B) + w_e[n], \tag{51}$$

$$\bar{y}_o[n] = h_A \sum_l s_A[n-l]p'(lT+\frac{1}{2}T-t_A) + h_B \sum_l s_B[n-l]p'(lT+\frac{1}{2}T-t_B) + w_o[n], \tag{52}$$

$$z_e[n] = \sum_l z_A[n-l]p'(lT), \tag{53}$$

$$z_o[n] = \sum_l z_A[n-l]p'(lT+\frac{1}{2}T), \tag{54}$$

where w_e, w_o are the even and odd entries of the i.i.d. Gaussian noise $\bar{w}^d$ in (15).

To obtain c_A^d, we then cross-correlate $\bar{y}^d$ with $z_A^{d\,*}$. Thus, the even and odd entries of c_A^d, i.e., c_e and c_o, are given by

$$c_e[m] = \sum_{n=0}^{Q-1} z_e^*[n]y_e[n+m] + \sum_{n=0}^{Q-1} z_o^*[n]y_o[n+m], \tag{55}$$

$$c_o[m] = \sum_{n=0}^{Q-1} z_e^*[n]y_o[n+m] + \sum_{n=0}^{Q-1} z_o^*[n]y_e[n+m+1]. \tag{56}$$

The first term on the R.H.S. of (55) gives

$$\sum_{n=0}^{Q-1} z_e^*[n]y_e[n+m]$$

$$\overset{(a)}{=} \sum_{n=0}^{Q-1}\sum_l z_A^*[n-l]p'(lT)h_A \sum_{l'} z_A[n+m-l']p'(l'T-t_A) + \sum_{n=0}^{Q-1} z_e^*[n]w_e[n+m]$$

$$\overset{(b)}{=} Qh_A \sum_l p'(lT)p'(lT+mT-t_A) + \widetilde{w}_e^1[m], \tag{57}$$

where (a) follows since we are only interested in the preamble parts embedded in s_A and s_B; (b) follows by setting $l' = l + m$.

In the same way, the second term on the R.H.S. of (55) can be simplified as

$$\sum_{n=0}^{Q-1} z_o^*[n]y_o[n+m] = Qh_A \sum_l p'(lT+\frac{T}{2})p'(lT+\frac{T}{2}+mT-t_A) + \widetilde{w}_e^2[m].$$

Substituting (57) and (58) into (55), we have

$$c_e[m]=Qh_A\sum_l \left[p'(lT)p'(lT+mT-t_A) + p'(lT+\frac{T}{2})p'(lT+\frac{T}{2}+mT-t_A)\right] + \widetilde{w}_e[m], \quad (58)$$

where $\widetilde{w}_e = \widetilde{w}_e^1 + \widetilde{w}_e^2$.

Similarly, (56) can be refined as

$$c_o[m]=Qh_A\sum_l \left[p'(lT)p'(lT+\frac{T}{2}+mT-t_A) + p'(lT+\frac{T}{2})p'(lT+T+mT-t_A)\right]+\widetilde{w}_o[m]. \quad (59)$$

Given (58) and (59), we combine the even and odd entries of c_A^d, yields

$$c_A^d[m]=Qh_A\sum_l \left[p'(lT)p'(lT+\frac{m}{2}T-t_A) + p'(lT+\frac{T}{2})p'(lT+\frac{m+1}{2}T-t_A)\right] + \widetilde{w}_A^d[m]. \quad (60)$$

B. The noise issue

In this subsection, we study the noise term in (60). We will prove that $\widetilde{w}_A^d$ is colored while its even and odd entries, i.e. $\widetilde{w}_e$ and $\widetilde{w}_o$, are white. The auto-covariance function of $\widetilde{w}_A^d$ is also given in this subsection.

First, we focus on the even entries $\widetilde{w}_e$. Based on (57) and (58), we have

$$\begin{aligned}\widetilde{w}_e[m] &= \widetilde{w}_e^2[m] + \widetilde{w}_e^2[m] \\ &= \sum_{n=0}^{Q-1} z_e^*[n]w_e[n+m] + \sum_{n=0}^{Q-1} z_o^*[n]w_o[n+m].\end{aligned} \quad (61)$$

Thus, the auto-covariance function of $\widetilde{w}_e$ is given by

$$\mathbb{E}\left[\widetilde{w}_e[m]\widetilde{w}_e[m+j]\right] \quad (62)$$

$$= \sum_{n=0}^{Q-1} z_e[n]w_e^*[n+m] \sum_{n'=0}^{Q-1} z_e^*[n']w_e[n'+m+j] + \sum_{n=0}^{Q-1} z_o[n]w_o^*[n+m] \sum_{n'=0}^{Q-1} z_o^*[n']w_o[n'+m+j]$$

$$\overset{(a)}{=} 2\sigma^2 \sum_{n=0}^{Q-1} z_e[n]z_e^*[n-j] + 2\sigma^2 \sum_{n=0}^{Q-1} z_o[n]z_o^*[n-j],$$

where in step (a), we set $n' = n - j$ and assume $N \gg j$.

Substituting (53) and (54) into (62), we have

$$\mathbb{E}\left[\widetilde{w}_e[m]\widetilde{w}_e[m+j]\right]$$

$$= 2Q\sigma^2 \sum_l \left[p'(lT)p'(lT - jT) + p'(lT + \frac{T}{2})p'(lT + \frac{T}{2} - jT) \right]$$

$$= 2Q\sigma^2 \sum_k p'(\frac{k}{2}T)p'(\frac{k}{2}T - jT), \tag{63}$$

To prove $\widetilde{\boldsymbol{w}}_e$ is white noise, we need to demonstrate that $\sum_k p'(\frac{k}{2}T)p'(\frac{k}{2}T - jT) = \delta(jT)$. The proof is given below.

Proof. According to Nyquist-Shannon sampling theorem, if the sampling rate is at or above the Nyquist rate, the continuous-time signal can be reconstructed from the discrete sampling points using the SINC function. Thus, if we sample the RRC pulse at double baud-rate, the original continuous RRC pulse could be reconstructed as

$$p'(t) = \sum_k p'(\frac{k}{2}T)\delta(t - \frac{k}{2}T) * p_s(2t)$$

$$= \sum_k p'(\frac{k}{2}T)p_s(2t - kT). \tag{64}$$

Since the convolution of two RRC pulses gives us an RC pulse, we have

$$p(t) = p'(t) * p'(t) = \int_{-\infty}^{\infty} p'(\varphi)p'(t - \varphi)d\varphi$$

$$= \sum_k \sum_{k'} p'(\frac{k}{2}T)p'(\frac{k'}{2}T) \int p_s(2\varphi - kT)p_s(2t - 2\varphi - k'T)d\varphi$$

$$= \sum_k \sum_{k'} p'(\frac{k}{2}T)p'(\frac{k'}{2}T)p_s(2t - kT - k'T). \tag{65}$$

Moreover, the RC pulse and SINC pulse satisfy the Nyquist ISI-free criterion, namely $p(jT) = \delta(jT), p_s(jT) = \delta(jT)$, where j is an integer. Substituting this criterion into (65), we have

$$\delta(jT) = p(jT)$$

$$= \sum_k \sum_{k'} p'(\frac{k}{2}T)p'(\frac{k'}{2}T)p_s(2jT - kT - k'T)$$

$$\overset{(a)}{=} \sum_k p'(\frac{k}{2}T)p'(\frac{k}{2}T - jT), \tag{66}$$

where (a) follows by setting $k' = 2j - k$. Eq.(66) concludes this proof. $\qquad\square$

Given (66), the the auto-covariance function of $\widetilde{\boldsymbol{w}}_e$ in (63) is given by

$$\mathbb{E}\left[\widetilde{w}_e[m]\widetilde{w}_e[m + j]\right] = 2Q\sigma^2\delta(jT). \tag{67}$$

Similarly, we could further obtain

$$\mathbb{E}\big[\widetilde{w}_o[m]\widetilde{w}_o[m+j]\big] \;=\; 2Q\sigma^2\delta(jT), \tag{68}$$

$$\mathbb{E}\big[\widetilde{w}_e[m]\widetilde{w}_o[m+j]\big] \;=\; 2Q\sigma^2 p(jT+\frac{T}{2}). \tag{69}$$

From (67), (68) and (69), we conclude that $\widetilde{\boldsymbol{w}}_e$ and $\widetilde{\boldsymbol{w}}_o$ are white, while $\widetilde{\boldsymbol{w}}_A^d$ is colored and its auto-covariance function is given by

$$\mathbb{E}\big[\widetilde{w}_A^d[m]^*\widetilde{w}_A^d[m+j]\big] = 2Q\sigma^2 p(\frac{j}{2}T). \tag{70}$$

APPENDIX B

PROOF ON POSITIVE DEFINITE MATRIX

In this appendix, we prove that the matrix $\boldsymbol{\Sigma}$ in (37) is positive definite. In particular, the matrix $\boldsymbol{\Sigma_0}$ in (20) is a special case of $\boldsymbol{\Sigma}$ when $\widehat{\tau} = T/2$. Thus, $\boldsymbol{\Sigma_0}$ is also positive definite.

Proof. A matrix $\boldsymbol{A}$ is said to be positive definite if the scalar $\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{b}$ is positive for every non-zero column vector $\boldsymbol{b}$. Thus, we will prove that $\forall\ \boldsymbol{b}=\big[b_0, b_1, ..., b_{2N-1}\big]^T\neq\boldsymbol{0},\ \boldsymbol{b}^T\boldsymbol{\Sigma}\boldsymbol{b} > 0$.

First, we have

$$\begin{aligned}
\boldsymbol{b}^T\boldsymbol{\Sigma}\boldsymbol{b} &= \big[b_0, b_1, ..., b_{2N-1}\big]\boldsymbol{\Sigma}\big[b_0, b_1, ..., b_{2N-1}\big]^T \\
&= \sum_{i=0}^{2N-1} b_i^2 + 2\sum_{i=0}^{N-1} b_{2i-1}\sum_{j=0}^{N-1} b_{2i}\,p(\widehat{\tau}+jT-iT)
\end{aligned} \tag{71}$$

Then, we define a new function $v(t)$ as

$$v(t) = \sum_{i=0}^{N-1} s_A[i]p'(t-jT) + \sum_{i=0}^{N-1} s_B[i]p'(t-jT-\widehat{\tau}). \tag{72}$$

In fact, $v(t)$ is the received noise-free signal at the relay and the energy of $v(t)$ is given by

$$\begin{aligned}
\Psi[v(t)] &= \int_{-\infty}^{\infty} v^2(t)dt \\
&= \int_{-\infty}^{\infty} \Big[\sum_{i=0}^{N-1} s_A[i]p'(t-jT) + \sum_{i=0}^{N-1} s_B[i]p'(t-jT-\widehat{\tau})\Big]^2 dt \\
&= \sum_{i=0}^{N-1} s_A^2[i]\int_{-\infty}^{\infty} p'^2(t-iT)dt + \sum_{i=0}^{N-1} s_B^2[i]\int_{-\infty}^{\infty} p'^2(t-iT-\widehat{\tau})dt \\
&\quad +2\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_A[i]s_B[j]\int_{-\infty}^{\infty} p'(t-iT)p'(t-jT-\widehat{\tau})dt
\end{aligned}$$

$$\overset{(a)}{=} \sum_{i=0}^{N-1} s_A^2[i] + \sum_{i=0}^{N-1} s_B^2[i] + 2 \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s_A[i]s_B[j]p(jT-iT+\widehat{\tau}), \tag{73}$$

where (a) follows since $\int_{-\infty}^{\infty} p'^2(t - iT)dt = 1$.

A key observation is that, if we take s_A and s_B as the even and odd entries of b, (73) is equivalent to (71). This implies that, given $b \neq 0$, $b^T\Sigma b$ equals to the energy of $v(t)$, i.e., $\Psi[v(t)] > 0$. Thus, Σ is a positive definite matrix. $\qquad\square$

APPENDIX C
OPTIMAL RRC-APNC DECODER UNDER BAUD-RATE SAMPLING

This appendix presents the optimal RRC-APNC decoder under baud-rate sampling. In general, the design follows the steps in Section IV: (1) Construct the discrete-time model using the baud-rate samples at the symbol boundaries. Note that baud-rate sampling enables samples with white noise, hence no noise whitening is needed. (2) Construct the factor graph based on the discrete-time model and transform it into an acyclic graph. (3) Run SPA to maximize the APPs and decode.

First, we resample $y_R(t)$ to obtain the samples at the symbol boundaries of s_A.[9] Giving,

$$\begin{aligned} y_0^b[n] &= y_R(t = nT + \widehat{t}_A) \\ &= h_A \sum_l s_A[n - l]p(lT + \widehat{t}_A - t_A) + h_B \sum_l s_B[n - l]p(lT + \widehat{t}_A - t_B) + w_0^b[n] \\ &= h_A s_A[n] + h_B \sum_l s_B[n - l]p(lT + \widehat{\tau}) + \epsilon^b[n] + w_0^b[n], \end{aligned} \tag{74}$$

where w_0^b is i.i.d. Gaussian noise, $w_0^b[n] \sim CN(0, \sigma^2)$; ϵ^b is an error vector as the ϵ in (38).

The error vector ϵ^b, caused by inaccurate symbol-misalignment estimation, is a function of t_A and t_B, and hence is unknown to the receiver. Thus, the receiver will ignore ϵ^b and assume

$$y_0^b[n] = h_A s_A[n] + h_B \sum_l s_B[n - l]p(lT + \widehat{\tau}) + w_0^b[n]. \tag{75}$$

We emphasize that this approximation is valid only when the symbol-misalignment estimation is accurate enough, or else, the decoding performance will suffer.

Then, we interpret (75) by a cyclic factor graph in Fig.15. In Fig.15, $y_0^b[n]$ forms the n-th observation node Y_n^b, $s_A[n]$ or $s_B[n]$ forms the odd-numbered observation node $X_{2n-1}^b, n =$

---

[9]We can also obtain the samples at the symbol boundaries of s_B. The decoder can be designed in the same way.

$1, 2, ..., N$ or even-numbered observation node $X_{2n}^b, n = 1, 2, ..., N$. The state size in each observation node is truncated so that each Y_n^b only connects 1 odd-numbered variable node and $L - 1$ even-numbered variable nodes. For instance in Fig.15, $L = 4$, and hence Y_n^b connects $\{X_{2n-2}^b, X_{2n-1}^b, X_{2n}^b, X_{2n+2}^b\}$.
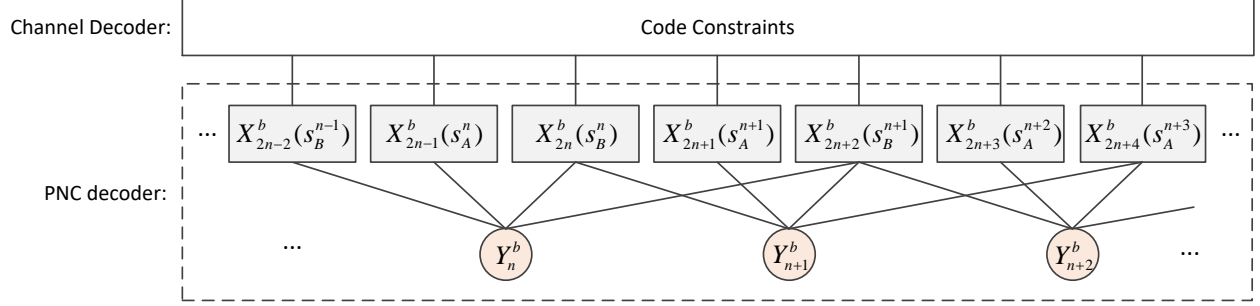

Fig. 15. The graphical interpretation of (75), X_i^b denotes the i-th variable node, $i = 1, 2, ..., 2N$; Y_n^b denotes the n-th observation node, $n = 1, 2, ..., N$. Each variable node X_i^b represents a transmitted symbol $s_A[n]$ or $s_B[n]$ (abbreviate as s_A^n or s_B^n). The state size in Y_n^b is truncated so that each Y_n^b is connected with L X_i^b ($L = 4$ in this figure).

Following the clustering in Fig.10, we group all the variable nodes that connect to the same observation node. Specifically, in Fig.15, we cluster $\{X_{2n-2}^b, X_{2n-1}^b, X_{2n}^b, X_{2n+2}^b\}$ and construct a new variable node $V_n^b = \{X_{2n-2}^b, X_{2n-1}^b, X_{2n}^b, X_{2n+2}^b\}$. The transformed acyclic factor graph is shown in Fig.16.
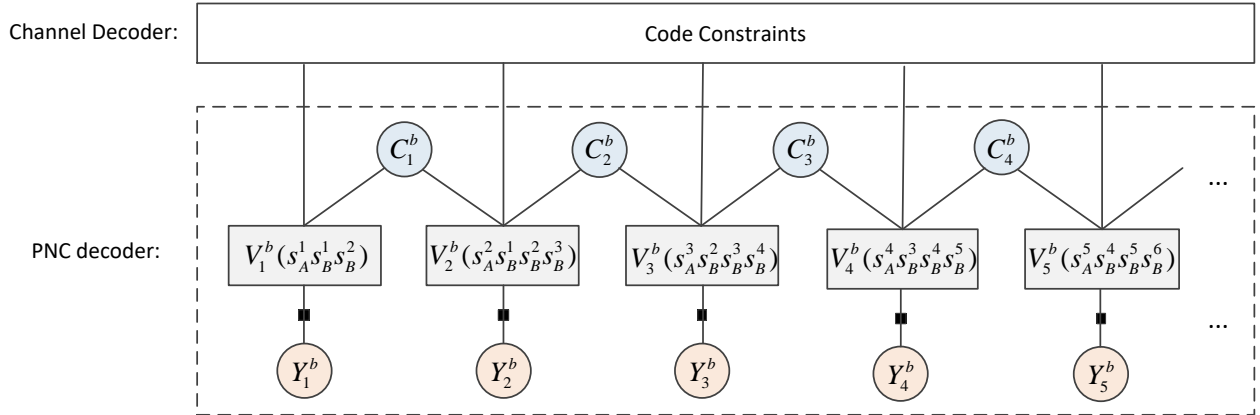

Fig. 16. The equivalent acyclic factor graph corresponding to Fig.15, where V_n^b denotes the n-th new variable node, $V_n^b = \{X_{2n-2}^b, X_{2n-1}^b, X_{2n}^b, X_{2n+2}^b\}$ ($L = 4$); C_n^b is the n-th check node; Y_n^b is the n-th observation node.

Given the transformed acyclic graph in Fig.16, we can then run SPA to maximize the APPs $\Pr\{s_A[n], s_B[n] \mid \bar{\boldsymbol{y}}\}, n = 1, 2, ..., N$. The operations afterwards are similar to that in Section IV.

REFERENCES

[1] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proceedings of the 12th annual international conference on Mobile computing and networking*.   ACM, 2006, pp. 358–365.

[2] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *IEEE ICC*, vol. 9, 2006, pp. 3885–3890.

[3] S. C. Liew, S. Zhang, and L. Lu, "Physical-layer network coding: Tutorial, survey, and beyond," *Physical Communication*, vol. 6, pp. 4–42, 2013.

[4] L. Lu and S. C. Liew, "Asynchronous physical-layer network coding," *IEEE Transactions on Wireless Communications*, vol. 11, no. 2, pp. 819–831, 2012.

[5] U. Bhat and T. M. Duman, "Decoding strategies at the relay with physical-layer network coding," *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4503–4513, 2012.

[6] Q. Yang, "Wireless communication by exploiting multi-user interference," Ph.D. dissertation, THE CHINESE UNIVERSITY OF HONG KONG (HONG KONG), 2015.

[7] X. Dang, Q. Li, and X. Yu, "Symbol timing estimation for physical-layer network coding," *IEEE Communications Letters*, vol. 19, no. 5, pp. 755–758, 2015.

[8] Q. Yang, S. C. Liew, L. Lu, and Y. Shao, "Symbol misalignment estimation in asynchronous physical-layer network coding," *IEEE Transactions on Vehicular Technology*, 2016.

[9] X. Wu, C. Zhao, and X. You, "Joint ldpc and physical-layer network coding for asynchronous bi-directional relaying," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 8, pp. 1446–1454, 2013.

[10] S. L. Miller, "Physical layer network coding with spectrally efficient pulse shapes," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 665–668, 2015.

[11] J. G. Proakis, "Digital communications, fifth edition," *McGraw-Hill, New York*, 1998.

[12] D. Tse and P. Viswanath, *Fundamentals of wireless communication*.   Cambridge university press, 2005.

[13] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, p. 15, 2008.

[14] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[16] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *IEEE Journal on selected areas in communications*, vol. 16, no. 2, pp. 140–152, 1998.

[17] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[18] ——, "An intuitive justification and a simplified implementation of the map decoder for convolutional codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 260–264, 1998.

[19] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, vol. 8, no. 2, pp. 119–125, 1997.

[20] B. S. Nugroho, "Digital communication systems simulation," https://sites.google.com/site/bsnugroho/, 2013.

[21] J. Chen, A. Dholakia, E. Eleftheriou, M. P. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of ldpc codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.